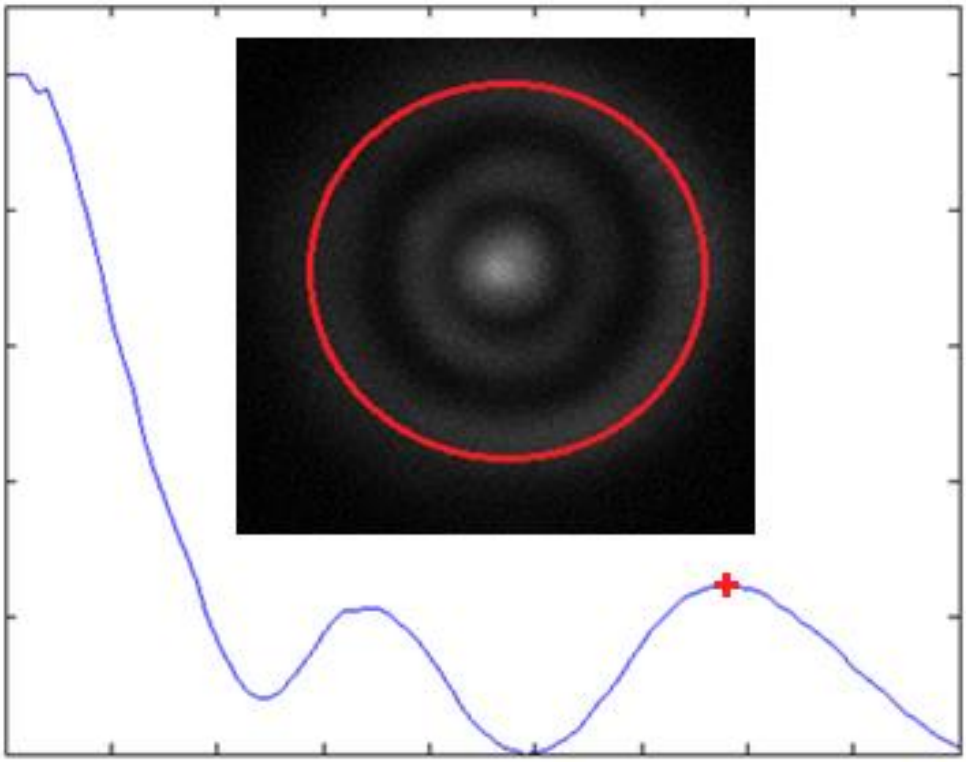
UNIVERSITY OF TROMSØ UIT

FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF PHYSICS AND TECHNOLOGY

Particle Tracking in 3D



Petter Brox

FYS-3921 Master's Thesis in Electrical Engineering

June 2013

Abstract

The evanescent field of an optical waveguide can be used to trap and propel microscopic particles. A 1µm particle is trapped on a strip waveguide and propelled forward until it reaches a 10µm gap at the centre of a loop, created by splitting the waveguide into two branches. According to simulations, the particle will be elevated 3µm in the gap and be stably trapped in the middle of the gap. The goal of this thesis is to implement an algorithm that can track a spherical particle in three dimensions and verify the simulated results. Information about all three coordinates is found by using off-focus images, in which diffraction rings will appear around the particle. The *x* and *y* coordinates are determined by calculating the centre of the particle in each image. A linear relationship between the radius of the diffraction ring and the displacement along the *z*-axis (vertical) is exploited to determine the *z* coordinate. Prior to tracking, it is necessary to make a calibration series where the radius is found for known displacements along the *z*-axis.

First, the algorithm was applied on a video of a trapped particle made with darkfield microscopy. That did not work, because of too much noise from the waveguides near the gap. Instead, fluorescence microscopy was used. Unfortunately, trapping within the gap was not achieved with the latter technique, but a video of a particle moving along the waveguide was analysed. The precision of ±0.2µm with fluorescence will be sufficient to verify simulations where the particle is elevated 3µm.

Acknowledgements

First of all, I would like to thank my supervisor, Professor Olav Gaute Hellesø. You have been of much help with regards to everything from laboratory work to discussions of methods. Thanks to Balpreet Singh Ahluwalia, who has come to the rescue when the things did not work in the optics lab. I also am grateful for the discussions I have had with Professor Frank Melandsø.

Petter Brox
Tromsø, June 2013

Contents

1 Introduction

Single particle tracking deals with spatiotemporal characterization of a single particle through analysis of videos. By determining the particle position for each instance of time, properties of the particle such as motility, diffusivity, velocity and morphology can be found [1]. Different techniques have been used for particle tracking in fields such as cell biology, fluid mechanics and colloidal sciences. In cell biology, examples of applications are determination of diffusion of proteins in cell membranes [2], and finding the 3D position of swimming E. coli cells [3].

Particle tracking in two dimensions, the x-y plane, is a fairly simple task. It mainly amounts to finding the centre of the particle for each image frame in the video. Information about the *z* coordinate is more challenging to acquire. Various approaches have been proposed for this purpose. In four dimensional microscopy (3D + time), information about the *z* position is obtained by acquiring a stack of images at various *z* positions [4]. This approach suffers from low temporal resolution and is less convenient than a simple 2D recording with regards to measurement setup. Another method determines the *z* position with respect to the intensity of the imaged particle. This can be done when using total internal reflection fluorescence microscopy (TIRFM) because the illumination field decays exponentially as the distance to the coverslip surface increases. Compared to 4D microscopy, this method is faster, as images are needed for one *z* position only. The weakness, however, is that the *z* position can only be assessed within a range of 250nm above the coverslip [5]. *Speidel et al.* suggest using off-focus images to determine the z position [6]. When the particle is imaged off-focus, diffraction rings become visible. In their work, they find a linear relationship between the radius of the outermost ring and the displacement along the *z*-axis. Their method is applied in this project.

This thesis has been written in relation to an experiment conducted by Professor Olav Gaute Hellesø and researcher Balpreet Singh Ahluwalia at the Department of Physics and Technology at the University of Tromsø. In their experiment, a 1µm particle is trapped by and propelled along a waveguide and reaches a gap between two opposing waveguides. There, according to their simulations, the particle will be trapped and elevated slightly above the substrate. The purpose of this thesis is to develop a 3D tracking algorithm and investigate this assumption from a recorded video. The particle is expected to be elevated 3µm along the *z*-axis after it has entered the trap.

Chapter 2 gives a brief introduction some experimental methods used in the thesis. This includes waveguides and how they can be used for particle trapping, diffraction, darkfield and fluorescence microscopy, and hysteresis in a piezoelectric actuator. The chapter also shows how images were preprocessed in free software, prior to analysis. The concepts of the tracking algorithm are introduced in the third chapter. This includes both the final method and approaches that were tried at an earlier stage. The setup of the microscopes for making calibration curves is explained in chapter 4. Chapter 5 discusses the different components and implementations of the tracking algorithm. Evaluation of the microscope setups and the piezoelectric actuators is done in the seventh chapter. The tracking algorithm applied on calibration images and videos of trapped particles are shown in chapter 7. Finally, chapter 0 concludes the thesis and gives suggestions for further work.

2 Brief introduction to some experimental methods

This chapter gives a brief explanation on how optical waveguides work and how they can be used to trap particles. Then, we look at the two different microscopy techniques for that have been used to image the particles. Section 2.5 says a few words about hysteresis in a piezoelectric actuator. Finally, the use of the image processing software Fiji in this project is described.

2.1 Optical trapping with waveguides

Optical waveguides are constructions that have the ability to transfer light. Commonly used waveguides include optical fibres and channel waveguides. In this project, a channel waveguide is used to trap a particle, and this section describes the design of the waveguides and the mechanisms of trapping.

Figure 2.1 shows slab waveguide that consists of three different dielectric layers which extend to infinity. The core layer has the highest refractive index. Monochromatic light is coupled into the waveguide. If the angle of incidence of the light is above the critical angle θ_c, given according to Snell's law, the light will travel through the waveguide by total internal reflection. Since the tangential components of both the electric and the magnetic fields must be continuous at the boundary between two layers, an evanescent wave, decaying exponentially, is formed outside the core layer. This evanescent field can affect particles nearby. It exerts a force that can be divided into two components, the gradient and the radiation force. The gradient force pulls particles down towards the waveguide, while the radiation force propels these particles along the waveguide surface. If a strip waveguide is being used and the particle is travels along the strip, it is also kept at the centre of it. The horizontal component of the force brings the particle back to the centre if it drifts slightly to the side.

Figure 2.2 illustrates the strip (channel) waveguide used in this project, seen from the front and including the dimensions. The waveguide is made of tantalum pentoxide (Ta_2O_5) on oxidised silicon substrate, with diffractive indices of 2.15 and 1.45, respectively. It has a thickness of 200nm and a width of 1.5µm (not shown).

Propulsion of particles long a waveguide is illustrated in **Figure 2.3**. In order to keep the trapped particle firmly in one place, a waveguide with a loop and a gap at the centre of it can be used [7], as shown in **Figure 2.4**. In this case, the particle is pulled to the waveguide and propelled towards the gap where it is held stably by the rapidly diverging fields from the waveguide ends. The forces on the particle depend on its position in the gap, which contains both stable and unstable trapping locations. In a stable point, a restoring force brings the particle back in place if it is moved slightly away. On the other hand, if the particle is moved slightly away from an unstable point, the forces will push it further away. Ideally, the particle is held in place as long as the laser remains on. If there is no phase difference between the light from the two waveguide ends, the particle will be trapped in the middle between them. Otherwise, it will be closer to one of them.

The evanescent field reaches about 150nm from above the waveguide surface [7]. Unpublished simulations by O.G. Hellesø and B.S. Ahluwalia of a 1µm trapped within a 10µm suggest that once it reaches the gap, it will be elevated 3µm.

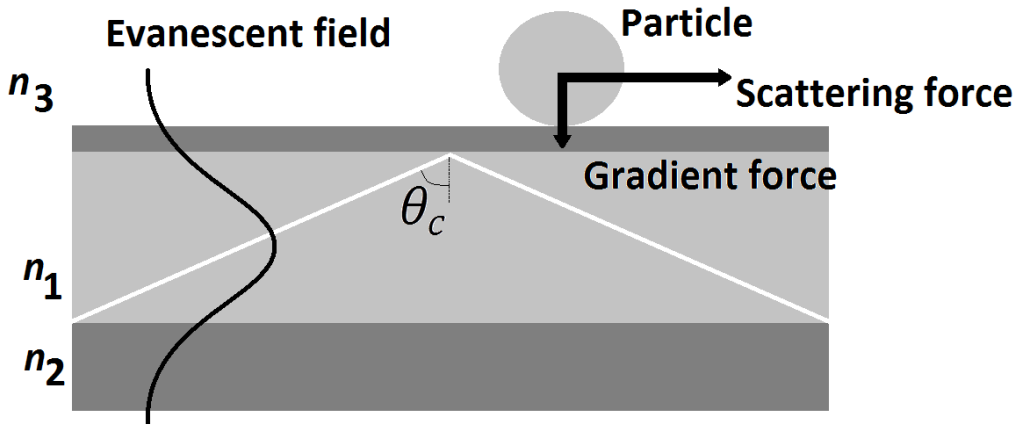


Figure 2.1: Slab waveguide with different layers. Propagating and evanescent field. Water on top with particle.

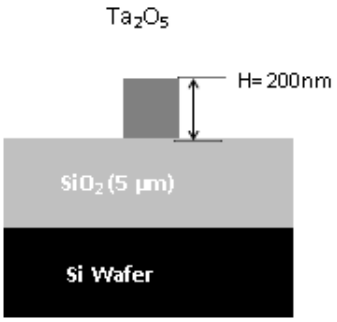


Figure 2.2: Strip waveguide seen from the front. Thickness 200nm. (From [8]**)**

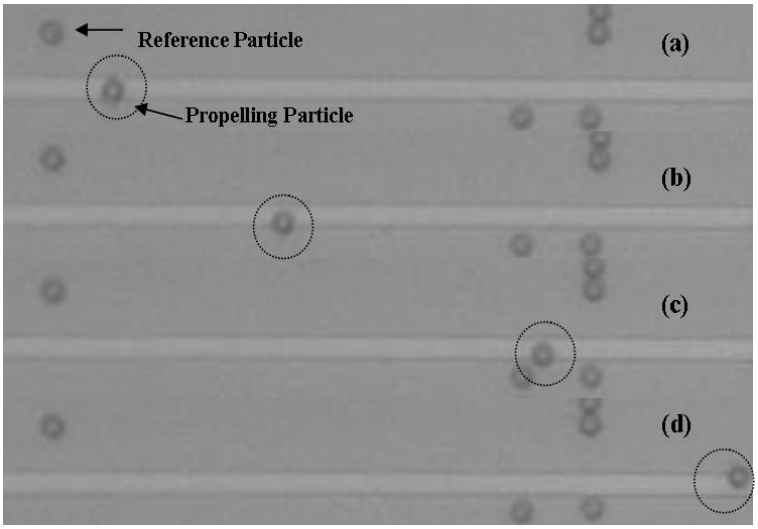


Figure 2.3: Optical propelling of 10 µm polystyrene particles (refractive index 1.59) in water on Ta_2O_5 waveguide. (From [8]**) (The dimensions on the figures are different from the ones in this project)**

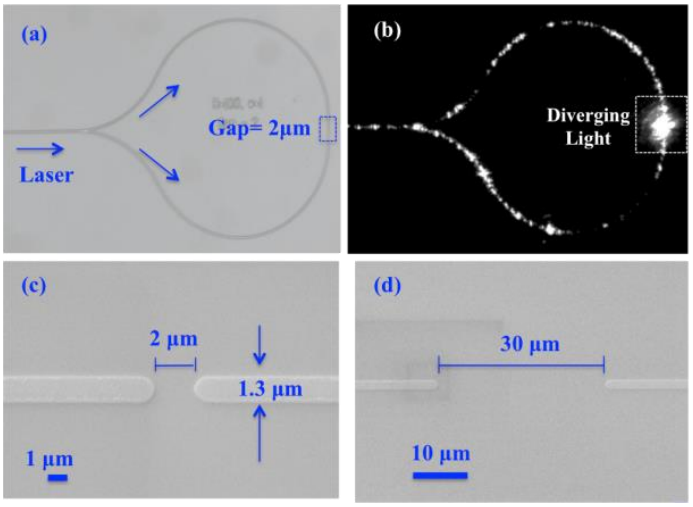


Figure 2.4: a) Bright field optical image of the waveguide loop with a gap. b) Scattered light from the waveguide and from the gap. c) and d) Scanning electron micrographs of gaps of width 2 µm and 30 µm, respectively. (From [7]**) (The dimensions on the figures are different from the ones in this project)**

2.2 Diffraction from a circular aperture

Diffraction is the term used to describe the changes that happen to a wave as it passes an obstacle. Depending on the shape of the obstacle, the diffraction pattern will take on different shapes. An example of diffraction is when light from a sample enters a microscope lens. The lens is confined by a circular region, outside of which no light is transmitted further in the microscope. The diffraction pattern that arises as the light passes through the lens can be modelled according to that of a circular aperture, as shown in **Figure 2.5**.

When a plane optical wave passes through a uniformly illuminated circular aperture, a symmetric diffraction pattern known as an Airy pattern will appear at a distance d if d is large enough so that the Fresnel number $N_F' = D^2/\lambda d \ll 1$ [9]. Here, D is the radius of the aperture and λ is the wavelength of the illumination light. The intensity of this pattern at a point (x, y) at the distance d is given by the equation [9]:

$$I(x,y) = I_0\left[\frac{2J_1(\pi D\rho/\lambda d)}{(\pi D\rho/\lambda d)}\right]^2, \quad \rho = \sqrt{x^2+y^2} \tag{1}$$

where J_1 is the Bessel function of the first kind and of order 1. The first kind of Bessel functions of order ν is given by [10]:

$$J_\nu(z) = \left(\frac{1}{2}z\right)^\nu \sum_{k=0}^{\infty} \frac{\left(-\frac{1}{4}z^2\right)^k}{k!\,\Gamma(\nu+k+1)} \tag{2}$$

An example of an Airy pattern is shown in **Figure 2.6**. The bright region at the centre, before the first minimum, is called the Airy disc. After the disc, there is an alternating series of dark and bright diffraction rings. As they become further away from the central disc, their intensity decreases. The alternating peaks and bottoms are caused by constructive and destructive interference, respectively. For an imaging system such as a microscope, the Airy pattern is the best focused image that a perfect, aberration free lens with a spherical aperture can make of a point source of light. If the image is defocused, the symmetry of the pattern is preserved, as equation (1) only depends on ρ, not on *x* versus *y*.

The appearance of the Airy pattern, when digitally recorded with a microscope, depends on the sensitivity of the camera, as well as the intensity of the light source. Therefore, the entire Airy pattern may not be apparent. When the specimen, say a micro particle, is imaged in focus, the ring pattern will often not be visible. As the image is defocused, the ring pattern will become clearer. For a perfect optical system with no aberrations, the image would be identical on either side of the focus. In reality, that is not the case. If off-focus imaging is desirable, one has to examine both sides to find which gives the best image.

In a real imaging system, various kinds of aberrations can occur to the image. One of them, spherical aberration, is caused by an increased refraction of light rays as they strike the lens at the edge [11]. This makes the outermost visible ring become brighter. An example of that is shown in **Figure 2.7**. The distance from the centre to the outermost ring becomes larger as the image is more defocused.

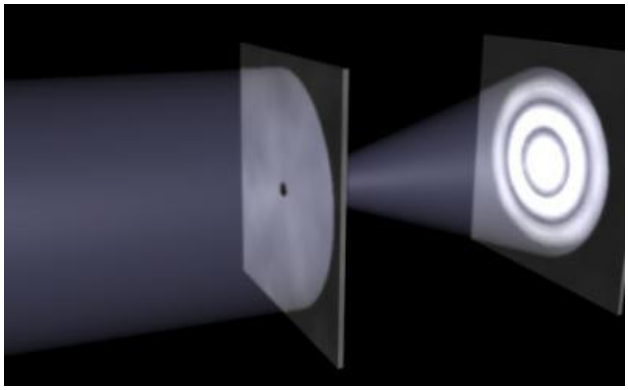


Figure 2.5 Diffraction of plane wave when passing through a circular aperture. (From [12]**)**

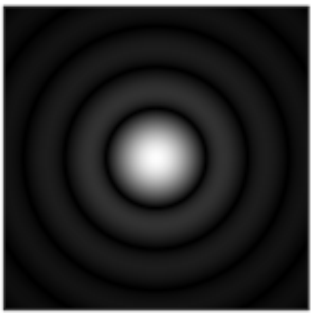


Figure 2.6: Example of Airy disk and pattern. (From [11]**)**

Figure 2.7: Airy pattern with spherical aberration. Outermost ring is very bright. (From [13]**)**

2.3 Darkfield microscopy

The information about darkfield microscopy, provided in this section, has been taken from the web pages Olympus Microscopy Resource Center, [14]. The starting point of the most basic darkfield microscope is a brightfield microscope. There, lenses refract the illumination light such that an inverted cone of light, with the apex in the specimen plane is formed, as shown on the left in **Figure 2.8**. A darkfield setup is created by inserting a circular disc which blocks central rays along the optical axis so that the cone of light becomes hollow. If there is no specimen present, none of the oblique rays enter the microscope objective and the field of view appears dark, as shown in **Figure 2.9**.

The presence of a specimen will cause the oblique rays to be either reflected, refracted or diffracted. In terms of Fourier optics, the zeroth order unscattered light from the diffraction pattern is removed in darkfield illumination. Since only higher order diffraction intensities scattered by the specimen enter the objective to form the image, the image has a higher degree of glare than the brightfield counterpart, and might therefore be slightly distorted. Darkfield microscopy is also inefficient, as most of the light is wasted in the zeroth order. It is sensitive to features above and below the focus plane, which easily distort the signal. Unlike fluorescence microscopy, darkfield is not selective, and will show every object present in the field of view.

There are several benefits of darkfield microscopy. No staining of the specimen is required. If the refractive index of the specimen is close to that of the surrounding medium, the specimen appears with more details than when imaged with brightfield microscopy. Because the image is based only on scattered light by the specimen, it is good for revealing edges, boundaries and refractive index gradients.

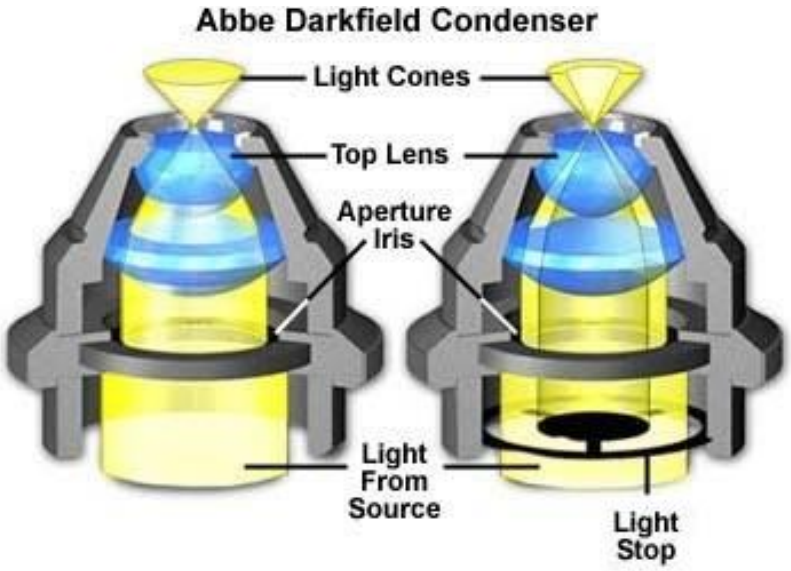


Figure 2.8: Brightfield (left) and darkfield (right) illumination. The difference in the images is a circular disk which blocks the central rays. (From [15]**)**

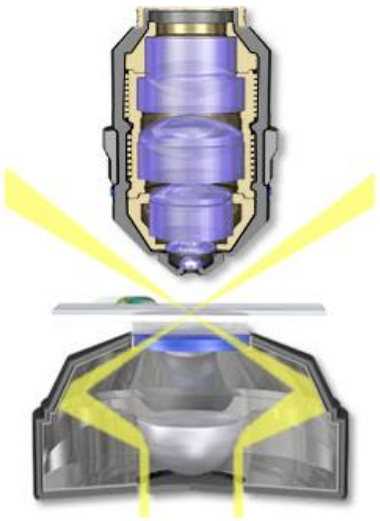


Figure 2.9: Darkfield setup where the specimen is not illuminated. The field of view will appear dark. (From [15]**)**

2.4 Fluorescence microscopy

The following text on fluorescence is taken from the web pages Olympus Microscopy Resource Center, [16]. Photoluminescence is the phenomenon of emission of light from a matter after the absorption of light of a shorter wavelength. When the emission happens almost spontaneously and lasts for a very short time, it is called fluorescence. If the radiation persists, the term phosphorescence is used. Fluorophores are materials that have this ability to fluoresce.

The main principle of fluorescence microscopy is to illuminate a sample with light of a certain wavelength, and transmit to the camera only the light of a longer wavelength that is emitted upon the excitation of the sample. There are several benefits of fluorescence microscopy. A very dark background (almost black) can be quite easily achieved, and only the parts of the sample that have the ability to fluoresce will produce a light signal. In this project, it is applied to view a particle trapped between two opposing waveguides. When using fluorescent particles, it is possible to get an image where the particle appears alone, thereby avoiding the noise in the image that would be due to waveguides nearby in a darkfield image. In this section we first look at some of the basics of fluorescence, and then the optical components that are necessary in a fluorescence microscope.

The atoms of fluorophores have the ability to absorb photons of certain wavelengths. Normally, at room temperature, atoms reside in what is called the ground state, S(0). When an atom absorbs an incoming photon, it may get excited to a higher energy state, S(1), S(2) or above. The atom can be excited to higher levels the shorter the wavelength of the absorbed photon is, as $E = h\nu = hc/\lambda$. Within each state, there are different vibrational and rotational levels. The Jabonski energy diagram in **Figure 2.10** shows the transition from the ground state to the first singlet state, and the resulting emission of a photon as the electron returns to the ground state. Each of the black, parallel lines represents a different vibrational level within the state. The transition from S(0) to S(1), marked with a red arrow, happens almost instantaneously, at a timescale of about 10^{-15} seconds. The relaxation within the first state, marked with a yellow, wavy arrow, takes anything from 10^{-12} to 10^{-10} seconds. The final relaxation to the ground state, marked with a blue, wavy arrow, represents the process of fluorescence where a photon is emitted as the electron loses vibrational energy. This step lasts for about 10^{-9} to 10^{-7} seconds. If the electron is excited to a higher state than S(1), it can relax directly back to the ground state or via lower states, including S(1). It is only when the electron relaxes from S(1) to S(0) that a photon can be emitted, although the excess energy can be dissipated non-radiatively as heat without any emission of light.

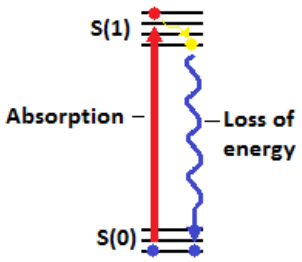


Figure 2.10: Jabonski diagram showing transition from ground state to higher state and back, and the associated emission of a photon. (Modification of figure in [16]**)**

Each flurophore absorbs a unique band of wavelengths. Only the wavelengths that can produce fluorescence constitute the excitation band. Often, the excited atom ends up in the lowest excited state, S(1), in which case the absorption and excitation spectra are the same. On many occasions, the emission spectrum is a mirror image of the absorption spectrum. As an example, **Figure 2.11** shows the absorption and emission spectra of the fluorescent dye Alexa Fluor 555. The second axis shows the relative intensity for each wavelength, and is related to the number of photons absorbed or emitted. The emission spectrum has been obtained by illuminating the flurophore with the wavelength of the absorption curve with the highest relative intensity. It should be noted that the wavelength of the emitted light is always longer than that of the absorbed light. The wavelength distance between the two maxima is known as the Stokes shift. In the figure, there is a wide region of overlap between the curves. In that region, the emitted photon has the same wavelength as the one absorbed. When using fluorescence microscopy, it is necessary to apply filters of the light at certain positions in the microscope. The final filter before the device that captures the image should be designed to pass and block as much as possible of the emitted and the illumination light, respectively. If, for instance, the fluorophore in **Figure 2.11** is used, a good choice of filter is one that passes 600nm and above.

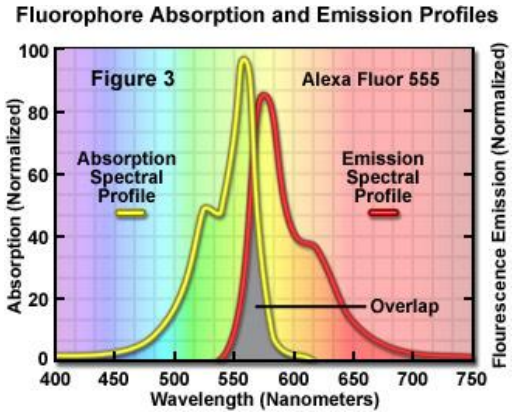


Figure 2.11: Excitation (here: absorption) and emission spectra for the fluorescent dye Alexa Fluor 555. (From [14]**)**

2.5 Hysteresis in a piezoelectric actuator

Piezoelectricity is the phenomenon of displacement of charge in a crystalline material in response to applied mechanical stress. Conversely, applying an electric field to the material leads to a mechanical strain, and is known as reversed piezoelectric effect. The latter principle is used in piezoelectric actuators, which allow for control with submicron accuracy. Upon application of the electric field, the material produces acoustic or ultrasonic motions used for linear or rotatory motion. There is, however, not a quite linear relationship between the drive voltage and the displacement produced.

Hysteresis is a process where the output or response of a system not only depends on the current input, but also on the past. In the case of piezoelectricity, hysteresis stems from molecular friction and crystalline polarization. Both the gain of a piezoelectric actuator, related to the remanent polarization, and the applied electric field determine the absolute displacement. Whether the actuator previously was operating at a high or a low voltage therefore affects the deflection of the piezo gain and the remanent polarization [17]. In the example in **Figure 2.12**, three envelopes with arrows show the increase/decrease from minimum to maximum drive voltage and the displacement produced by the actuator.

Piezoelectric actuators show hysteresis in their large range of operation. When uncompensated, this is known as open loop operation. In some applications this might not be a problem. If, on the other hand, a high degree of repeatability and linearity between the response and drive voltage is required, the system needs to be equipped with a measurement system and closed loop electronics. The motion of the piezo system can then be controlled by feedback from the position sensor.

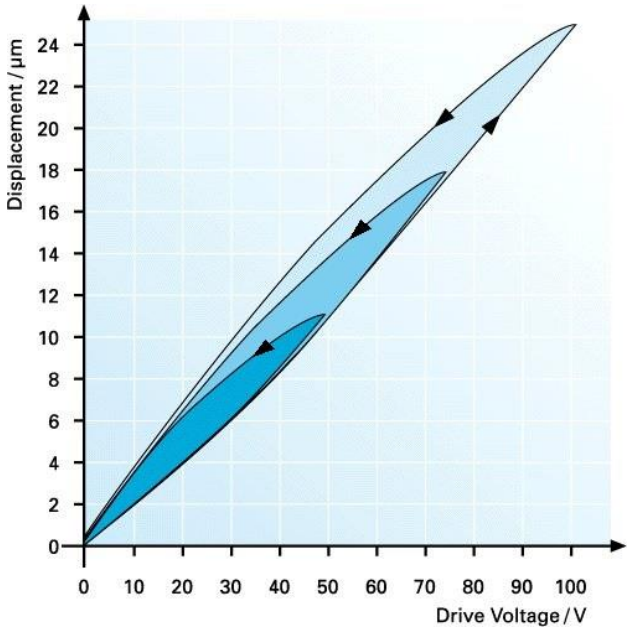


Figure 2.12: Hysteresis curves. Displacement as the function of voltage. (From [17]**, slightly modified)**

2.5 Preprocessing of images in Fiji

Fiji is an open source imaging processing software which contains numerous tools and plugins for image analysis, including particle tracking in 2D, 3D+t [18], [19]. In this project, it is used extensively for preprocessing of images prior to applying the tracking algorithm described in chapter 5. For videos of moving particles, the only processing used is cropping the image stack to isolate the region of interest, as shown in **Figure 7.4** and **Figure 7.5**.

For calibration series, it is desirable that the algorithm finds the centre of the particle with as high accuracy as possible. It is thus desirable to remove as much noise as possible first. **Figure 2.13** shows an image from a darkfield calibration series. The whole sequence of images, taken at various *z* positions (section 3.3), is imported and processed as a stack. **Figure 2.14** A) shows the particle in **Figure 2.13** marked with a yellow arrow after the image has been cropped. A copy is made of the image in **Figure 2.14** A). The region covered by the particle is deleted and becomes black. The image is then thresholded so that all intensities above 0 become 255 (white). This gives the mask shown in **Figure 2.14** B). This image is subtracted from the entire stack of images, thereby removing large areas of noise. **Figure 2.14** C) and D) show the result for A) and an image later in the calibration series, respectively. The images are then saved as a sequence to be imported in MATLAB.

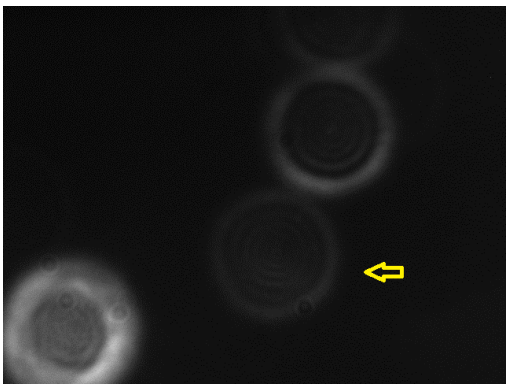


Figure 2.13: The yellow arrow shows the most out-of-focus image in a calibration series made with darkfield microscopy.

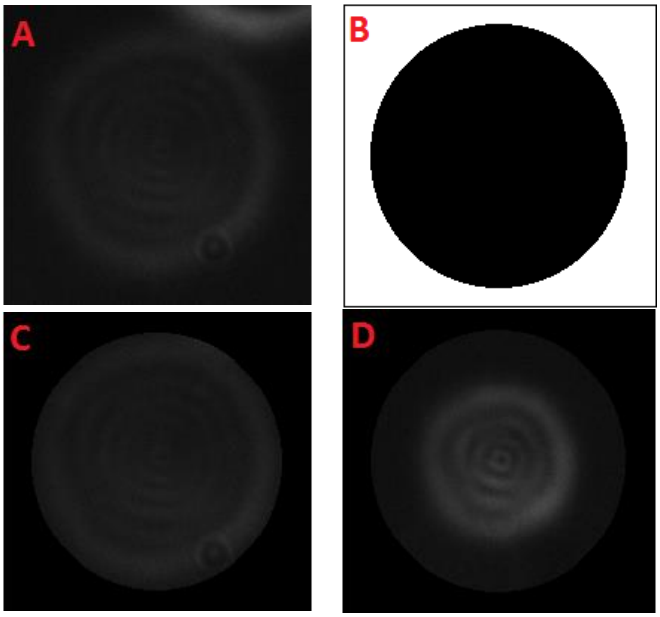


Figure 2.14: A) shows the particle in Figure 2.13 cropped. B) shows the mask. When it is subtracted from the entire stack, the particle in A) looks like C). D) shows the result after subtraction for an image later in the stack.

3 Introduction to particle tracking

A method for particle tracking in 3D based on off-focus images was presented in [6]. This chapter discusses the necessary steps of an algorithm which uses that approach. The first section looks at two methods for finding the *x* and *y* centre coordinates of a particle in an image. Once the centre is known, a symmetric particle can be represented in one dimension instead of as a two dimensional image. Two approaches will be presented for this purpose in section 2. The third section discusses the use of the one dimensional representations of the particle in off-focus images for finding the *z* coordinate.

3.1 Finding the *x* and *y* coordinates of the particle centre

Two methods for finding the *x* and *y* coordinates of the centre of a particle will be presented. The task is to determine the centre of a discrete set of points $\{I_{ij}(x_{ij}, y_{ij})\}$, where I denotes the intensity and the subscripts label rows and columns, respectively. Both methods are analytical, meaning that they estimate the centre of the particle directly through a formula. Iterative methods for this purpose also exist. A frequently used iterative method is fitting a two dimensional Gaussian to the intensity distribution of the particle. For small particles, the algorithm yields accurate (low bias) and precise (low variance) results [20]. However, being an iterative method, it is more computationally demanding than a direct procedure. It also presupposes a shape of the intensity of the particle. When a particle moves out of focus and an Airy pattern appears, the Gaussian fitting is no longer a suitable approach.

The first method to be presented to determine the centre of the particle is called centre of mass or centroid. An improved method using radial symmetry will be explained afterwards.

3.1.1 Centre detection with the centre of mass (centroid) method

The centre of mass method estimates the centre (x_c, y_c) of the particle by the following equations [20]:

$$x_c = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} x_i I_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}} \quad \text{and} \quad y_c = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} y_j I_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}} \tag{3}$$

The equations represent weighted averages of the image coordinates, where each pixel is weighted by its intensity value. Two implementations of the algorithm will be used in this project. MATLAB provides a built-in function named *regionprops* which can return various properties of an image, including the centroid and area of each object in the image. The function requires a binarily thresholded image. In order to be able to apply the centroid method to non-thresholded images, it was implemented as the function *centerofmass*. The results for both functions are shown in section 5.2.

The centroid method is intuitive and easy to implement. However, it is quite sensitive to noise. Prior to applying it to an image, it is therefore common to threshold the image, either binarily, or by subtracting a certain intensity level from all pixels in the image in order to remove as much as much of the background as possible.

3.1.2 Centre detection by exploiting the radial symmetry of the particle

An algorithm using radial symmetry to find the centre of a particle is introduced by R. Parthasarathy in [21]. It provides an analytic calculation of the particle centre with subpixel precision, and is applicable to any symmetric distribution. Its performance has been shown to be comparable to Gaussian fitting when a Gaussian distribution is suitable [21].

The principle behind the algorithm is that for a radially symmetric intensity distribution, a line parallel to the gradient in each point will intersect the centre. At the centre, the distance to each line is zero. This is illustrated in **Figure 3.1**.

Theoretically, the centre of the image is the point where the distance to all the lines that are parallel to the gradient is zero. In a digital image, where noise obscures symmetry, the centre can be approximated as the point that minimizes the sum of distances to the gradient-oriented lines. In the algorithm, the gradient in each pixel position is calculated at the centre of four neighbour pixels, and with the coordinate system rotated 45° counter clockwise, as shown in **Figure 3.2**.

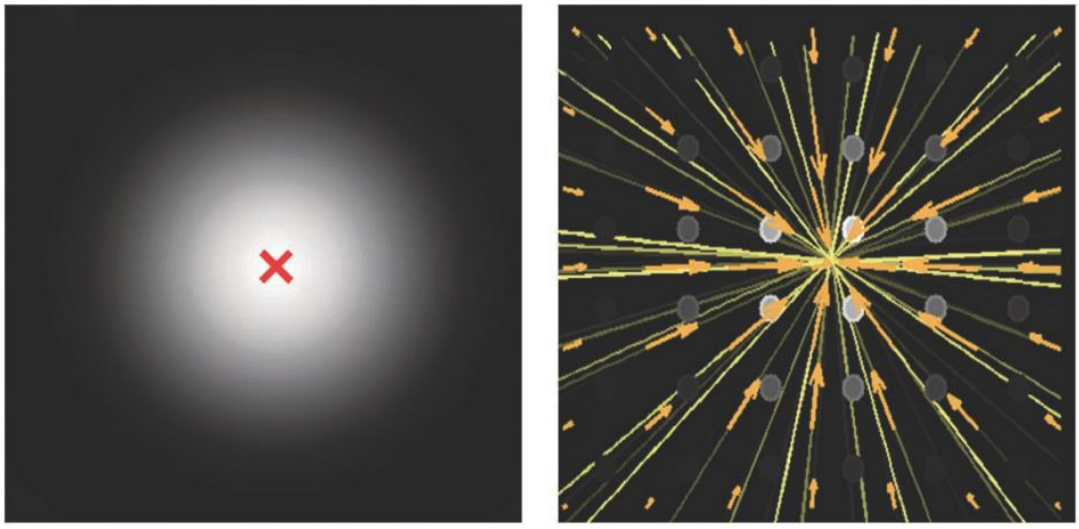


Figure 3.1: The red cross on the image on the left shows the centre of the particle. On the right, the orange arrows mark the gradient in each point, all starting at the centre of four neighbouring pixels. Note that the yellow lines, parallel to the neighbouring gradient, all pass through the centre. (Figure from [21]**)**

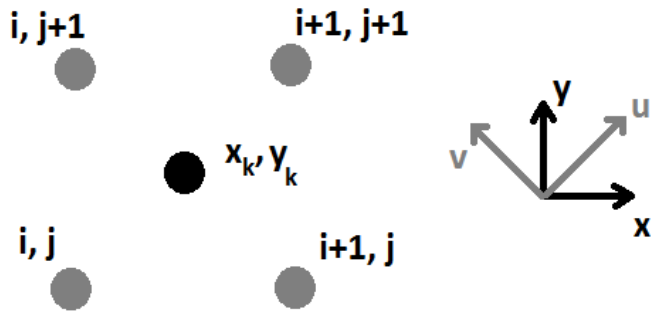


Figure 3.2: For each pixel the intensity gradient is calculated between the four neighbouring pixels.

A summary of the derivation of the formulae for the centre, shown below, is taken from the supplementary notes of [21].

The gradient at the midpoint (x_k, y_k) is calculated by the equation:

$$\vec{\nabla} I_k = (I_{i+1,j+1} - I_{i,j})\vec{u} + (I_{i,j+1} - I_{i+1,j})\vec{v}$$

The slope of the gradient at that point is given by:

$$m_k = \frac{(I_{i+1,j+1} - I_{i,j}) + (I_{i,j+1} - I_{i+1,j})}{(I_{i+1,j+1} - I_{i,j}) - (I_{i,j+1} - I_{i+1,j})}$$

A line with slope m_k (parallel to the gradient) passing through (x_k, y_k) is given by the equation $y = y_k + m_k(x - x_k)$. The distance d_k from an arbitrary point (x_c, y_c) to the line is given by:

$$d_k^2 = \frac{[(y_k - y_c) - m_k(x_k - x_c)]^2}{m_k^2 + 1}$$

The centre point (x_c, y_c) can be found by minimizing the weighted sum of all the various distances d_k, i.e. by minimizing $\mathrm{X}^2 \stackrel{\text{def}}{\equiv} \sum_k w_k d_k^2$. Differentiating the expression with respect to x_c and y_c, respectively, equating each equation to zero and then combining them, yields the following solutions for the centre coordinates:

$$x_c = \Delta^{-1}\left[\left(\sum_k \frac{m_k w_k (y_k - m_k x_k)}{m_k^2 + 1}\right)\left(\sum_k \frac{w_k}{m_k^2 + 1}\right) - \left(\sum_k \frac{m_k w_k}{m_k^2 + 1}\right)\left(\sum_k \frac{w_k (y_k - m_k x_k)}{m_k^2 + 1}\right)\right] \quad (4)$$

$$y_c = \Delta^{-1}\left[\left(\sum_k \frac{m_k w_k (y_k - m_k x_k)}{m_k^2 + 1}\right)\left(\sum_k \frac{m_k w_k}{m_k^2 + 1}\right) - \left(\sum_k \frac{m_k^2 w_k}{m_k^2 + 1}\right)\left(\sum_k \frac{w_k (y_k - m_k x_k)}{m_k^2 + 1}\right)\right] \quad (5)$$

where

$$\Delta = \left(\sum_k \frac{m_k w_k}{m_k^2 + 1}\right)^2 - \left(\sum_k \frac{m_k^2 w_k}{m_k^2 + 1}\right)\left(\sum_k \frac{w_k}{m_k^2 + 1}\right) \quad (6)$$

For two reasons, weighting the distances d_k differently is reasonable. Firstly, the intensity gradient is determined more accurately near the centre, as the signal to noise ratio is higher there. Secondly, the distances d_k are determined more accurately near the actual centre because errors in m_k lead to errors in d_k, which grow in a noisy image as the distance to the centre increases. The following weight takes care of both concerns:

$$w_k = \frac{\left|\vec{\nabla} I_k\right|^2}{d_{k,c}} \quad (7)$$

The distance $d_{k,c}$ is the distance between the centre and each pair of (x_k, y_k). It can be approximated by the distance d_c between (x_k, y_k) and the centroid of $\left|\vec{\nabla} I_k\right|$. A smoothing of the m_ks prior to calculating the distances d_k yields a significant improvement in accuracy and execution time. It can be achieved by applying a 3x3 smoothing filter to the m_ks.

The MATLAB implementation of this algorithm, named *radialcenter*, is provided by R. Parthasarathy in [22].

3.2 Representing a rotationally symmetric particle in one dimension

The intensity distribution of a particle with rotational symmetry in two dimensions can be represented in one dimension, as illustrated in **Figure 3.3**. In this project, this was done in two different ways. In the text, they are referred to by the names of their MATLAB implementations. The method of discrete circles is referred to as *meanint*, while the method using weighted distances is named *radial_intensity* in MATLAB. Both methods find the average intensity along concentric circles around the particle centre. The term used for the 1D intensities as a function of radius is the radius vector. Sometimes it is referred to as the intensity plot.

In this project, 8 bits grayscale images are used, and directly after calculation the radius vectors will also contain values from 0 to 255. For the purpose of detection of the peak, each radius vector has its lowest value subtracted and is then scaled so that it contained values between 0 and 1.

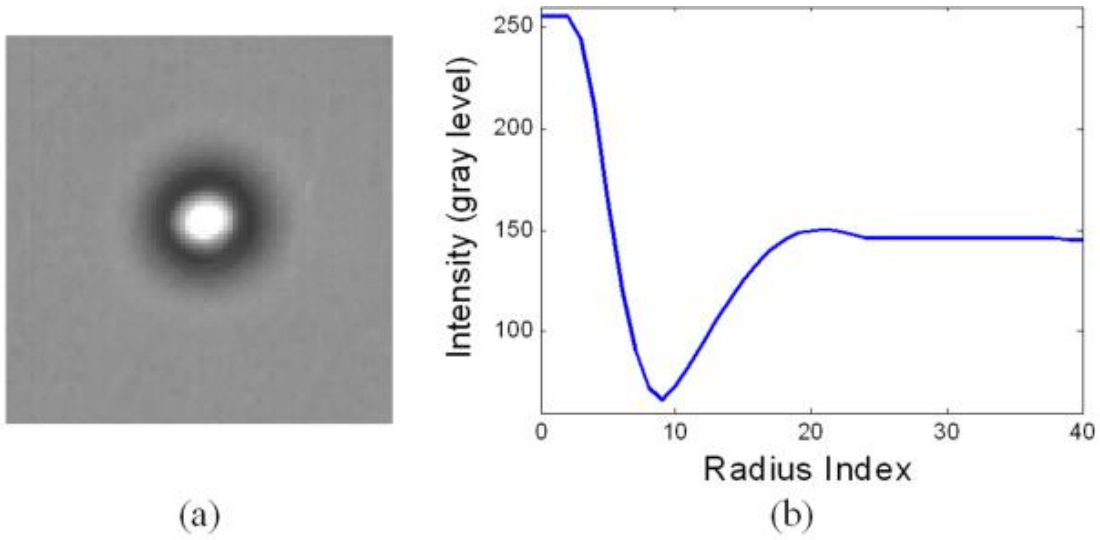


Figure 3.3: The 2D intensities of the particle in (a) can be represented in one dimension (b). (From [23]**)**

3.2.1 The method of discrete circles

Given the centre coordinates (x_c, y_c) of a particle, points along a continuous circle with radius *r* and centre in (x_c, y_c) are given by the equations:

$$x = r\cos\theta + x_c \quad \text{and} \quad y = r\sin\theta + y_c \tag{8}$$

where $\theta \in [0, 2\pi)$.

In **Figure 3.4**, the centre of a particle is plotted as a yellow cross. The green circles are the points given by equation (8), and with radii 1 and 2. When the values of the points that make up the continuous circles are rounded off, they are reduced to the alternating blue and yellow points, which are the discrete representations of the circles. In the figure, these integer points are plotted at the centre of each pixel. The yellow and blue points represent the points of the discrete circles of radius 1 and 2, respectively. Some of the blue points have been plotted on top of the yellow ones. This shows that one pixel may be used by more than one discrete circle. The radius vector is calculated by averaging the intensities of the pixels that make up each discrete circle, for all radii that are necessary to cover the entire particle.

In the MATLAB implementation of this procedure, named *meanint*, the coordinates calculated with equation (8) are rounded to the nearest integer value. The radius is set to vary in steps of 1, from 0 to just beyond the border of the diffraction pattern of the particle. The largest radius has to be chosen by inspecting the image.

One weakness of this method is that the pixels used to calculate the radius vector are weighted equally for different radii, independent of their distance to the continuous circle. Even if the

continuous circle just barely touches a pixel, the intensity contribution is treated no differently than if the circle passes through the centre of it.

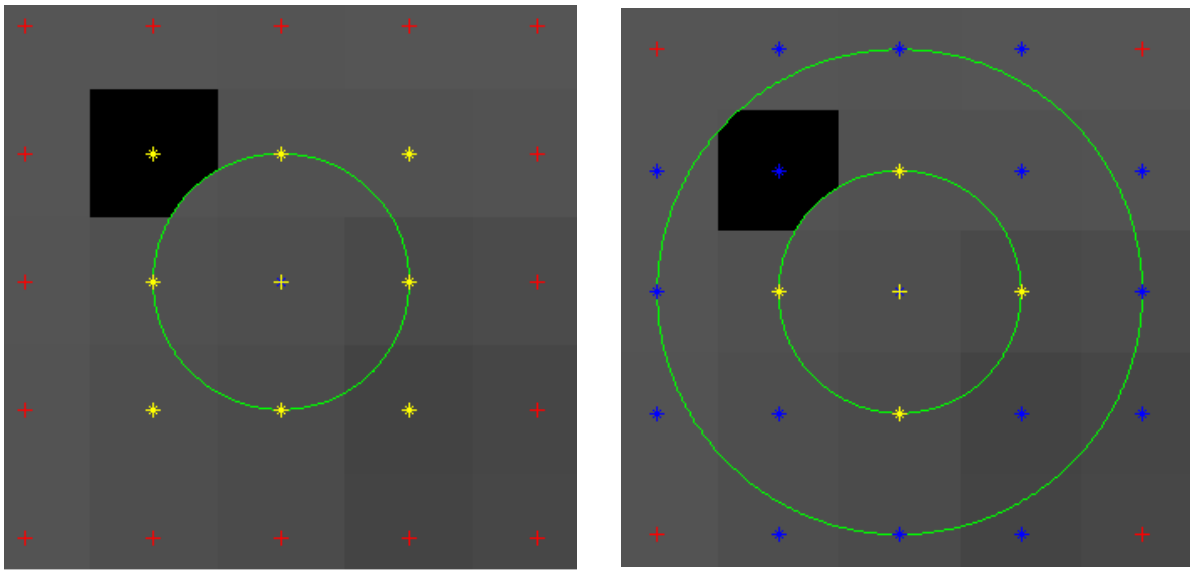


Figure 3.4: The particle centre is plotted with a yellow cross. Pixels that have not been used for drawing a discrete circle are marked with a red cross. The green circles have radii 1 and 2. On the left, the yellow points show the discrete circle for a radius of 1. On the right, blue points show the discrete circle of radius 2. The black pixel shows one pixel that is used for both radii.

3.2.2 The method of weighted distances

A more sophisticated algorithm for projecting the 2D image onto a single axis is given in [23]. We define the radii of the radius vector as discrete values $r_i = i\Delta$, $i = 0,1, \dots, N$, where Δ is the spacing between two adjacent radii in the radius vector and i is the index of the radius vector. The ith point of the radius vector, m_i, is given by the following equation:

$$m_i = \left(\sum_{k=1}^{M_i} c_{i,k} \right)^{-1} \sum_{k=1}^{M_i} c_{i,k} I_{i,k} \tag{9}$$

where

$$c_{i,k} = \frac{d - r_{i-1}}{\Delta}, \quad \text{if } r_{i-1} \leq d < r_i$$
$$\text{or} \tag{10}$$
$$c_{i,k} = \frac{r_{i+1} - d}{\Delta}, \quad \text{if } r_i \leq d < r_{i+1}$$

M_i is the total number of pixels within the region $[0, r_1)$ for $i = 0$, within $[r_{i-1}, r_{i+1})$ for $0 < i < N$,and within $[r_{N-1}, r_N]$ for $i = N$. $I_{i,k}$ is the intensity of the kth pixel associated with the ith point in the radius vector, and its weight $c_{i,k}$ is evaluated according to the distance d of a pixel to the centre of the particle. An advantage of this method, compared to *meanint*, is that each pixel is weighted differently for two for two adjacent radii in the radius vector.

In **Figure 3.5**, the radius vector is being calculated for the ith radius, r_i, which corresponds to the green circle between the grey and blue region. The pixels within the grey and blue annuli satisfy equation (10), and are the ones that are used for this radius index.

The algorithm is implemented in MATLAB as the function *radial_intensity*. In addition to a maximum radius used to represent the particle, a choice of Δ also must be entered. It should not be much smaller than the pixel width [23]. Throughout the entire project a value of 1 is used. Section 6.3 briefly discusses the choice of the parameter when applying the method it to calibration images.

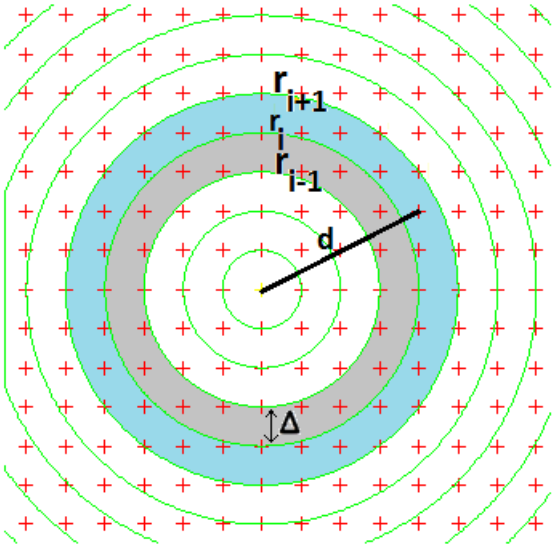


Figure 3.5: The radius vector corresponding to the circle of radius r_i, between the grey and the blue annuli, is being calculated. Red crosses mark the centre of each pixel. Only pixels within the two annuli are considered when calculating the radius vector. The pixels within the two regions are weighted according to equation (10)**.**

3.3 Finding the *z* coordinate of the particle

When imaging a particle with a microscope, it appears as a small, bright spot at and near the focus. As the image is increasingly defocused, the size of the particle seems to increase and it appears as an Airy pattern (section 2.2). In [6], a linear relationship is found between the displacement along the *z*-axis and the radius at the peak of the outermost ring, as seen **Figure 3.6**. Such a plot can be made by immobilizing a particle, imaging it at various known *z* displacements, and then finding the radius at the outermost peak in the radius vector. A fitted line through these data points can then be used for describing the displacements along the *z* axis in a video. It must be emphasized that this approach

only shows relative displacement of the particle in the *z* direction, not the exact position. In the remainder of this thesis, the term *rz* curve is used for the linearly interpolated plot of the calibration data. A measure of linearity is the standard deviations of the observations from the fitted line (section 6.4).

Care must be taken to image the particle at a large enough distance from the focus. If it is too close, the ring pattern may not appear. Section 4.3 describes in more detail the experimental procedure for obtaining a calibration curve. The implementation of the image processing procedure necessary to use this approach is described in chapter 5.

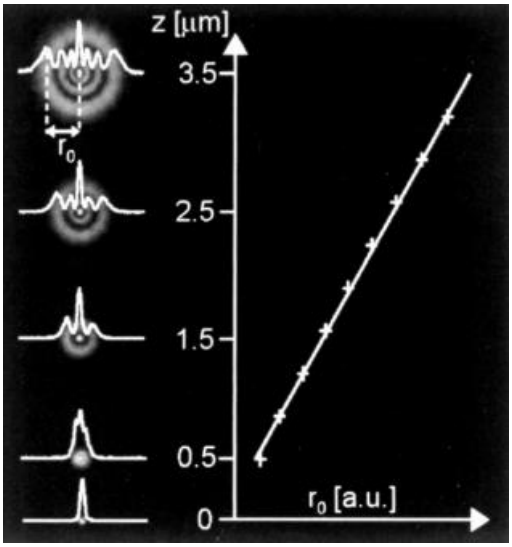


Figure 3.6: Example of a linear relationship between the radius of outermost ring and the particle displacement along the *z*-axis. (From [6]**)**

4 Setup

4.1 Setup for darkfield microscopy

The Olympus brightfield/darkfield microscope that was used is sketched in **Figure 4.1**. **Figure 4.2** outlines the pathway of the illumination light, which is slightly different from the one in section 2.3.

The equipment was mounted on an optical table. Instead the light coming from below, the source of the illumination light was behind the microscope and was reflected down to the specimen. Darkfield was created by pulling a switch which introduced a light stop in the pathway to block off the central rays.

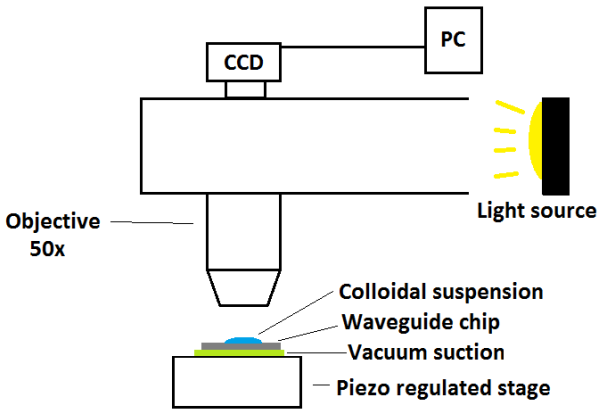


Figure 4.1: Outline of the darkfield microscope.

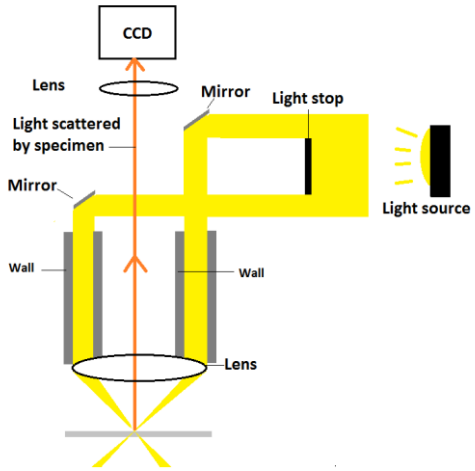


Figure 4.2: Details of light passage in darkfield microscope. Central rays are blocked before the light is reflected to the lens which focuses the light.

4.2 Setup for fluorescence microscopy

The fluorescence microscope was built around the Olympus brightfield microscope by adding some components above the objective. **Figure 4.3** shows the setup. The passage of the illumination light is sketched in **Figure 4.4**. A green laser of 532nm, used to excite fluorescence, is collimated by a lens of focal length 5cm. A dichroic mirror reflects the green light down. The light is spread evenly across the sample by a 50x Olympus objective with a long working distance and numerical aperture 0.50.

Both reflected illumination light from the waveguide chip and fluorescence light enter the microscope. Some of the green light is reflected by the dichroic mirror and thus prevented from reaching the CCD, while a small portion is transmitted. Most of the remaining fluorescence light is blocked by the 625±52nm band-pass filter. The emitted light is transmitted through both the dichroic mirror and the filter and reaches the Infinity 2 CCD-camera.

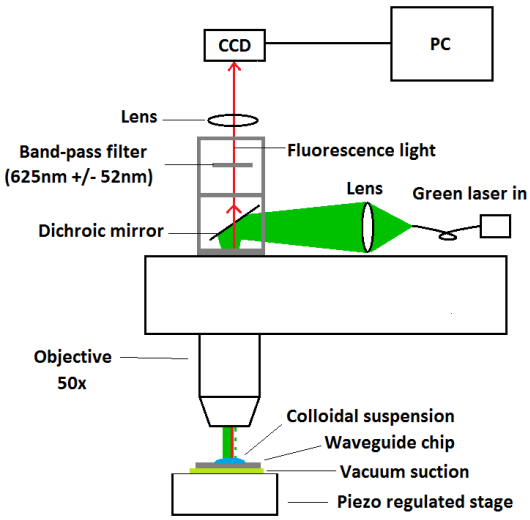


Figure 4.3: Outline of the fluorescence microscope.

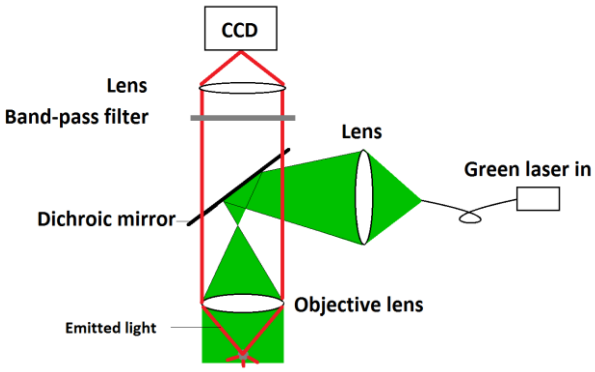


Figure 4.4: The passage of the illumination light in the fluorescence microscope.

4.3 Making a calibration curve for tracking in the *z* direction

Our method for particle tracking in 3D is based on measuring the radius r of the outermost ring in an off-focus diffraction pattern. For this method, it is necessary to have a calibration curve that shows the relationship between the radius and the change of displacement in the *z* direction. This section describes the experimental procedure for obtaining such an *rz* plot.

A waveguide chip, typically around 20x20 mm^2, was thoroughly cleaned. A drop of a colloidal suspension of particles and water was placed inside a PDMS micro chamber on the chip. The plate was left in ambient conditions for about 30 minutes for the water to evaporate. Particles in the suspension had then adhered to the glass plate and were immobilized. A few drops of water were added within the micro chamber and a cover strip was put on top of it.

The chip was then placed on a piezoelectric actuator. For both imaging with darkfield and fluorescence microscopy, the particle was first localized with brightfield microscopy to verify that the particle was of the desired size. Before imaging 1µm particles, we first tried with 3.87µm particles to see if the microscope worked well. There still might be larger particles stuck to the chip after cleaning it. Also, in the case of darkfield microscopy, dust or dirt could easily be mistaken for a particle. Verification with darkfield was thus necessary.

The particle to be imaged must be sufficiently far away from others so the diffraction rings did not interfere when defocusing the image. Before starting to image the particle, it was necessary to check which side of the focus, above or below, that gave the clearest diffraction rings. In both cases, the particle was imaged for decreasing distances to the microscope objective. Steps of either 1µm (or Volts) or 0.5µm were carefully made with the piezoelectric actuator. An image was acquired for each position. Deviations from this was noted and considered later when making the *rz* plot.

After an image series had been made, some simple processing of the stack was performed in Fiji, as described in section 2.5.

5 Finding the radius of diffraction rings for a particle at known *z*-position

This chapter compares the results of different approaches used in the steps of the algorithm that finds the 3D coordinates of a particle using the radius of the diffraction pattern in off-focus images. The first step is to find the centre of the particle, which is done both manually and automatically. Further, two different approaches for representing the particle in one dimension will be compared. The chapter proceeds with looking at how to detect the peak in the radius vector.

Darkfield images from two different calibration series, as well as a couple of images taken with fluorescence microscopy, have been used to illustrate the methods. All images that are shown have been normalized in order to see them clearly, as the original intensities often are quite low. The normalization is done by to the following equation:

$$I_n = 255\frac{I - \max(I)}{\max(I) - \min(I)} \tag{11}$$

I is the entire original image array, and I_n is the normalized. The lowest intensity is subtracted from all the pixels. Dividing by the difference between the maximum and minimum intensity, the intensities are scaled to $[0, 255]$. The linear transformation of normalizing the images does not affect the detection of the radius found from the radius vector notably (not shown). Before the particle centre could be found, images had to be cropped so that the particle of interest appeared in the centre and as the only object in the image, as can be seen in all images of a particle in this chapter.

The series of darkfield images used in section 5.1 and 5.2 were made using the open loop piezoelectric actuator. From section 5.3 and through the remainder of the chapter, the images were made with the closed loop piezoelectric actuator and darkfield microscopy, except for two plots in section 5.4.1 with fluorescence microscopy. The use of darkfield or fluorescence microscopy is not of importance for the development of the tracking algorithm, and is not commented on further in this chapter.

5.1 Manual centre detection

The image processing software Fiji [19] has a plugin called Manual Tracking, which allows the user to go through a stack of images and for each image simply click on the centre of the particle. Fiji automatically records the coordinates and shows the next image. The information is then saved in an Excel document and can be transferred to MATLAB either by copy-pasting or by importing the file.

This method was used extensively in the beginning in of the project, as the first automatic procedures that were tried sometimes gave obviously wrong estimates of the centre. A reasonably well detected centre was necessary to continue developing the next steps of the tracking algorithm. **Figure 5.3** and **Figure 5.4** compare the results of this “method of the eye” with the automatic procedure that became the standard.

5.2 Automated centre detection

The first automated method that is tried for finding the centre of the particle is centre of mass, described in section 3.1.1. The algorithm is included in the built-in MATLAB function *regionprops*, which determines the centroid from a thresholded binary image. In order to choose a threshold, it might be necessary to guess an appropriate value and inspect the series of images with the particle centre plotted on top, then adjust the threshold and repeat. The choice of threshold can greatly affect the estimated centre.

Figure 5.1 shows two examples where the centroid algorithm in *regionprops* sometimes misses the centre. In both images, the red cross shows the centre determined from an image thresholded at a value of 50, while for the yellow cross, a value of 70 was used. (The intensity range was from 0 to 255.)

The method of centre of mass can just as well be used on non-thresholded images. MATLAB's *regionprops* requires a thresholded image, while the implemented method named *centerofmass* does not. Its performance varies a lot too, as shown in blue on the left in **Figure 5.2**. The intensities of the dark background of the images in this series were generally around 50. The blue cross shows the centre found with *centerofmass* where the value 50 first had been subtracted from the original image. For that particular image, binary thresholding at 70 gave a quite good result, while a threshold at 50 missed the centre completely. On the right in **Figure 5.2**, where the particle is in focus, all three, binary threshold at 50 and 70 and centre of mass with 50 subtracted from the image, find practically the same estimate for the centre. That seemed more or less the case upon inspection of all images where the particle was near focus.

Figure 5.1: Centres estimated with centre of mass on binary images. Threshold at 50 (red) and 70 (yellow).

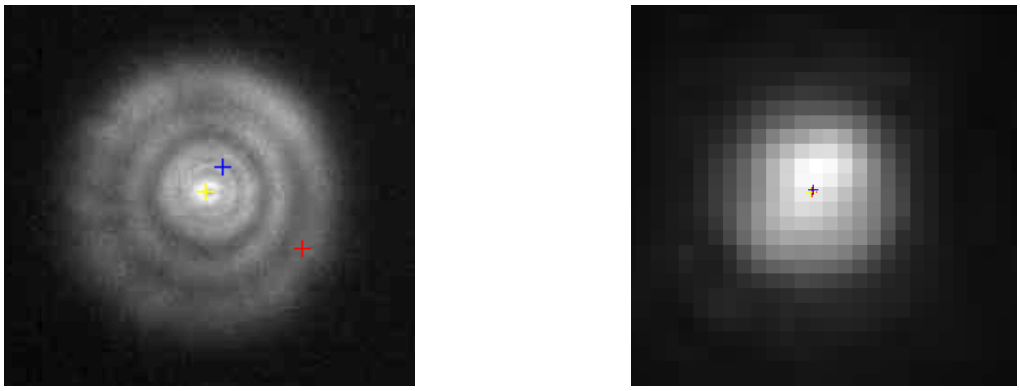


Figure 5.2: Left: Red and yellow show centre with regionprops for thresholds 50 and 70, respectively. Blue shows centre calculated with centerofmass, having subtracted 50 from the image. Right: Almost the same result for red, yellow and blue in this in-focus image.

A second method for finding the centre exploits the radial symmetry of the particle. It is named *radialcenter* in MATLAB and was explained in section 3.1.2. **Figure 5.3** shows the centres found with *radialcenter* (green) and two separate series of manually selected centres (red and yellow). The first and the last are the most in- and off- focus images of the entire series. The new automatic method seems to find the centre with convincing accuracy. **Figure 5.4** shows the distances between the centres found with radial symmetry and each of the two separate series with Manual Tracking. The distances generally stay below 2.5 pixels. For the last two or three images, the difference is above 3 pixels. That is simply because the same coordinates were hastily labelled as the centre, as it was hard to guess which pixel was the centre by visual judgement. This shows that using Manual Tracking was a fairly good solution when lacking a precise automatic procedure. It also shows that the algorithm using radial symmetry is likely to produce results that look reasonable. *radialcenter* thus became the standard method for centre detection. Still, when processing an image and detecting the centre, it was necessary to always perform a visual check and not simply assume that the algorithm will find the centre correctly.

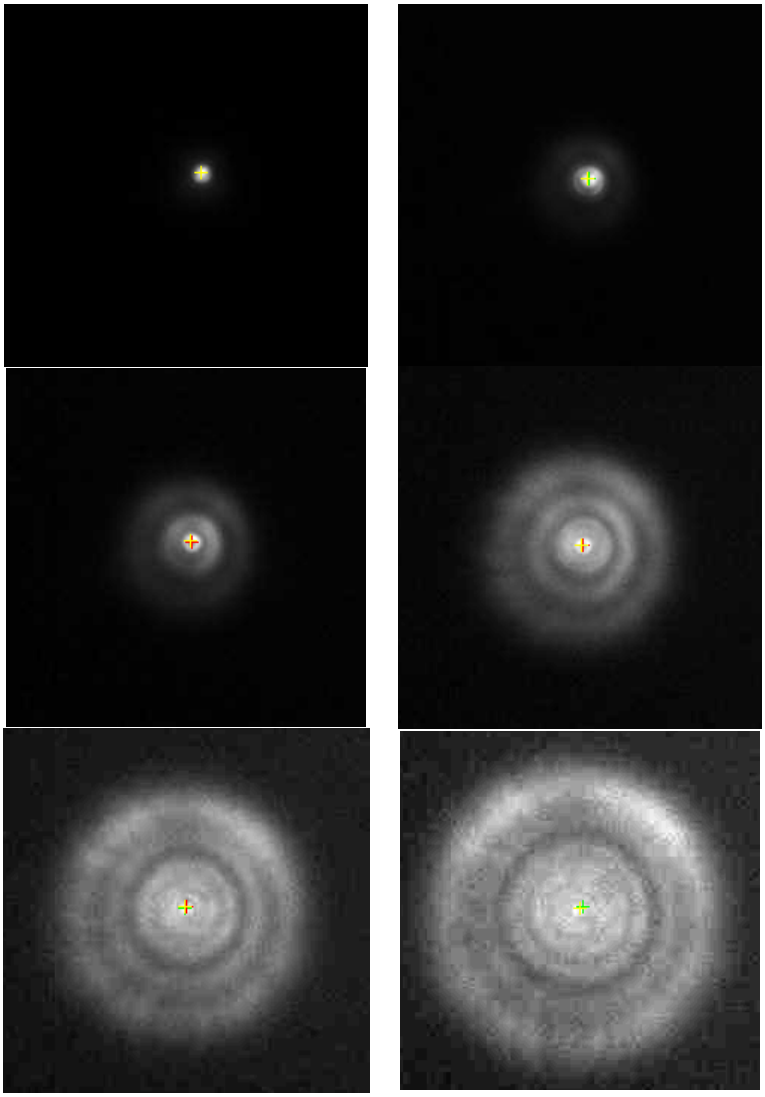


Figure 5.3: Particle increasingly out of focus. The green cross shows centre determined with radial symmetry. Red and yellow crosses show the centre determined with two separate series using Manual Tracking.

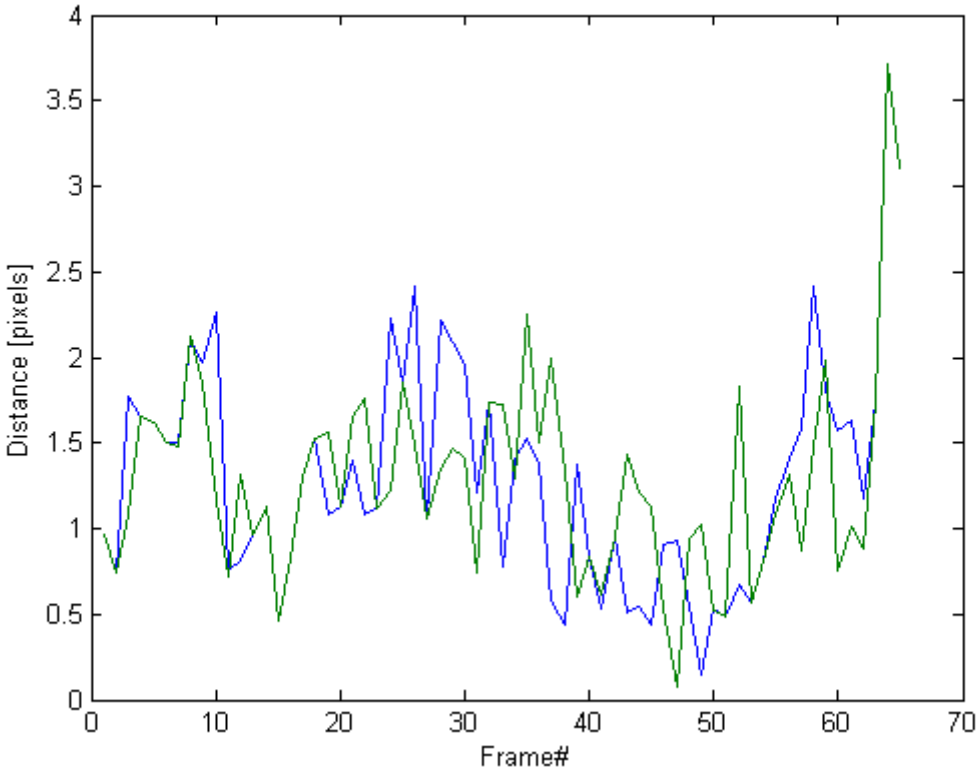


Figure 5.4: Two series of distances between centres found with radial symmetry and with Manual Tracking.

Smoothing the images prior to calculating the centres was tested to reduce the contribution of noise. **Figure 5.5** shows four images of the particle where the centre has been calculated with the original image and with a smoothed version of it (red and green cross, respectively). The images in the figure are not smoothed. Smoothing was done with a simple 3x3 mask of values 1/9. The distance between the centres for smoothed and unprocessed images for the entire series is shown in **Figure 5.6**. The image on the bottom right in **Figure 5.5** is the last one in the series, for which the distance between the calculated centres is between 8 and 9 pixels. By visual inspection, it is clear from **Figure 5.5** that the centre was most accurately detected using unprocessed images. No other methods of pre-processing were attempted to improve the results of the centre coordinates.

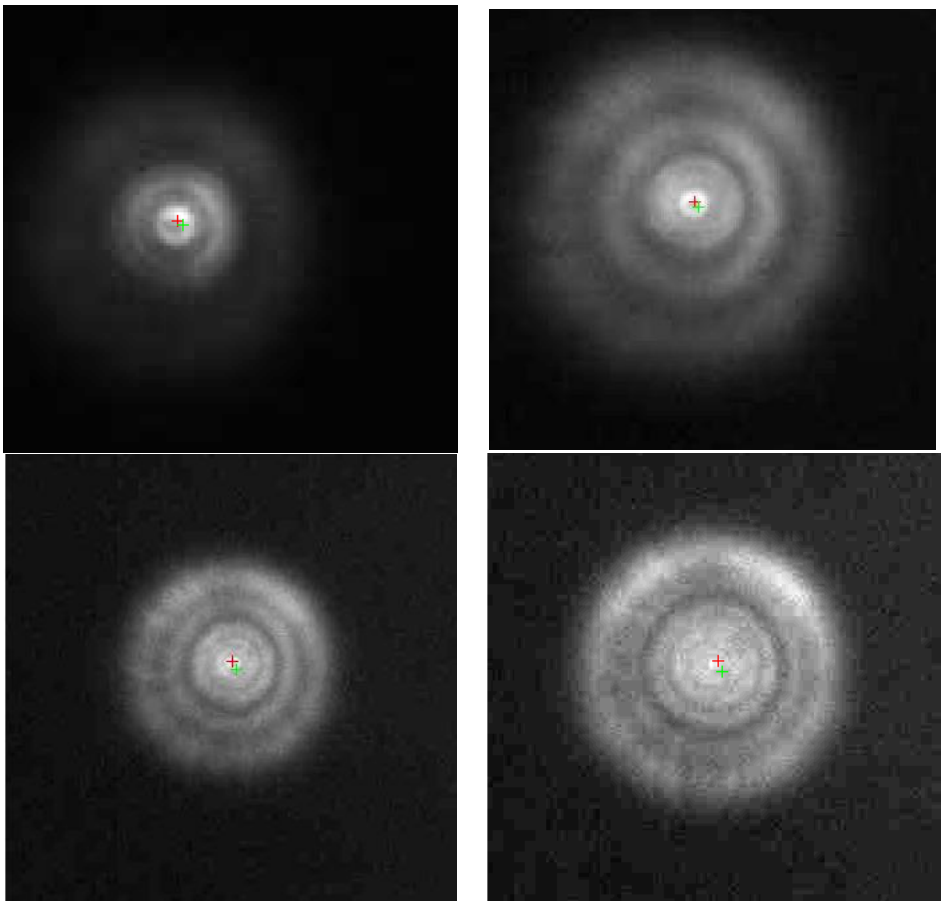


Figure 5.5: Centres detected with *radialcenter*. Red and green cross for unsmoothed and smoothed images, respectively. Unprocessed images seem to give more accurate results.

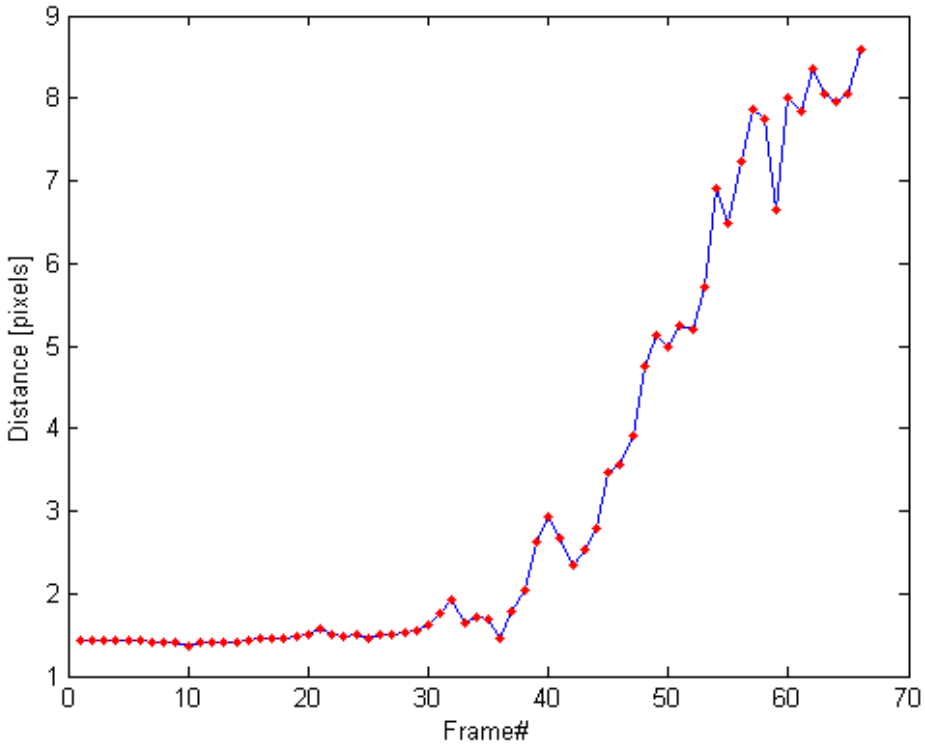


Figure 5.6: Distance between centres calculated with *radialcenter* using original and smoothed image. Particle radius increases with each frame.

5.3 Calculation of radius vector

Once the centre of the particle is found, the next step in the tracking procedure is to represent the imaged particle in one dimension. For that purpose, two different approaches are presented to calculate the so called radius vector. The method using discrete circles (named *meanint*) is explained in section 3.2.1. The method of weighted distance (named *radial_intensity*), is explained in section 3.2.2.

An example of a smooth and a noisy radius vector are shown on the left and right in **Figure 5.7**, respectively. Smooth is taken to mean "no abrupt changes among the data points with respect to the general trend visible to the eye". The outermost peak seems to be at a radius of 35. It is reasonable to assume that the noisier and rougher a plot is, the more inaccurately the detected peak will be.

To compare the two methods, a roughness measure is implemented and applied on radius vectors found with both procedures. Given a continues function $f(r)$, the roughness related to curvature can be measured by considering the second order derivative. Here, the roughness R is defined as:

$$R(f) \equiv \int (f''(r))^2 dr \qquad (12)$$

over the region where $f(r)$ is defined. This is a slight change of the definition used in [24].

If $I(r)$ is a vector of intensities where the radii are spaced by unit steps, we define the roughness as:

$$R(I) \equiv \sum \left(I''(r)\right)^2 \tag{13}$$

where the discrete second derivative is calculated by convolving the points of $I(r)$with the vector $[1 \quad -2 \quad 1]$.

A comparison of the two methods is made on both a computer generated image of an Airy disc and a darkfield image from a calibration series. For both images, the roughness of the radius vectors produced by the two procedures is calculated without noise, and with Gaussian noise added to the image. The noise has expectation 0 and standard deviation up to 0.5. The intensity values in the radius vectors are between 0 and 1. The same centre coordinates are used within each of the two series, based on the points detected with no noise added. **Figure 5.8** and **Figure 5.9** show the original image of the Airy disc and the particle, respectively, as well as two examples of images with noise added.

The roughness plots for the two methods, used on the Airy disc and the real particle, respectively, are show in **Figure 5.10**. In both plots, the general trend of the roughness is that it increases with greater noise. *radial_intensity* seems to produce marginally smoother radius vectors for both images. **Figure 5.11** shows the roughness calculated for the entire calibration series of the particle in **Figure 5.9**. Again, *radial_intensity* seems to be the best choice with regards to smoothness.

radial_intensity generally produces smoother plots than *meanint* because it finds a weighted average of intensities from two adjacent annuli. *meanint* finds the mean intensities along discrete circles. Some pixels might occur in two adjacent discrete circles, and their intensities count the same for both radii. Since its development, *radial_intensity* became the standard algorithm for finding the radius vector.

However, upon the final development of a procedure to locate the radius of the outermost ring, (explained in section 5.4.2) a comparison was made of two *rz* plots based on *meanint* and *radial_intensity*, respectively. For this particular series of calibration images, the largest difference between the two *rz* plots was below 1 pixel (not shown). This confirms that the two algorithms work equally well.

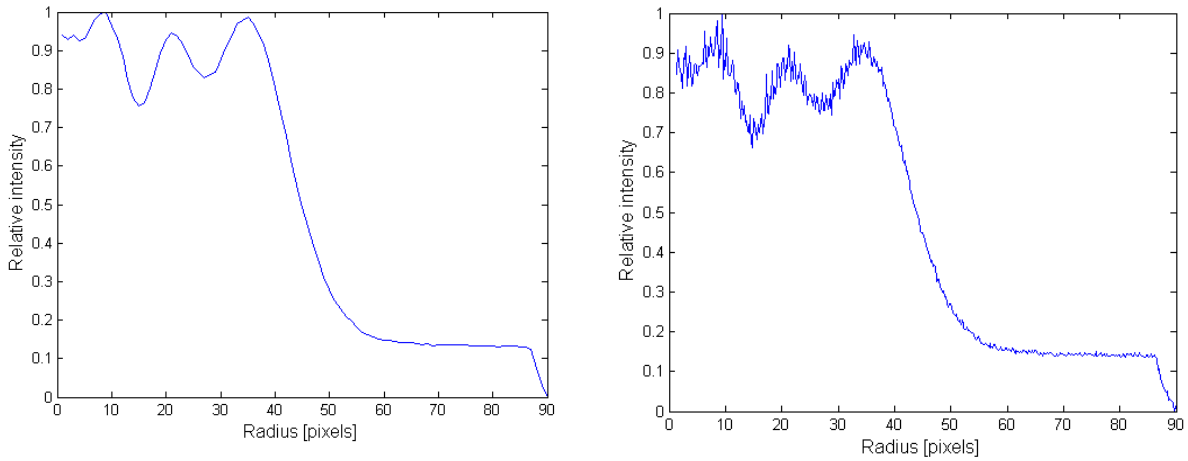


Figure 5.7: On the left: A smooth radius vector. On the right: A noisy radius vector.

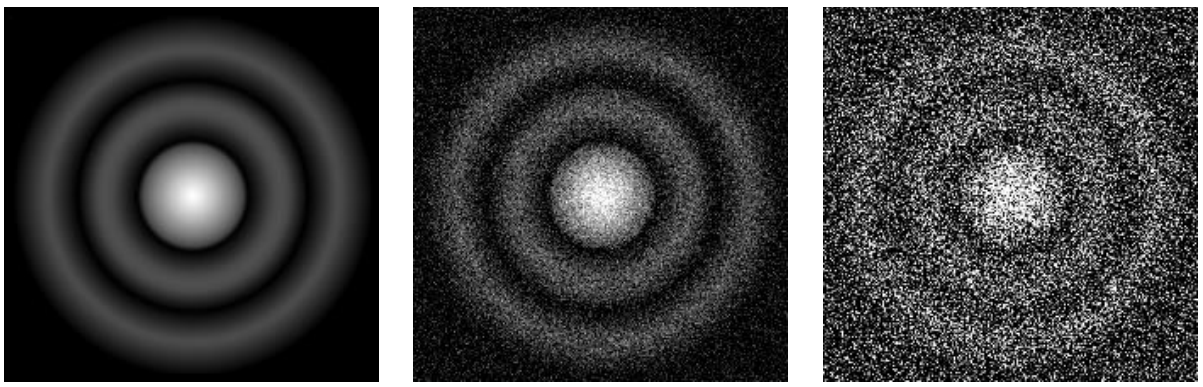


Figure 5.8: Computer generated Airy disc with no noise, and Gaussian noise with standard deviation 0.13 and .5, respectively.

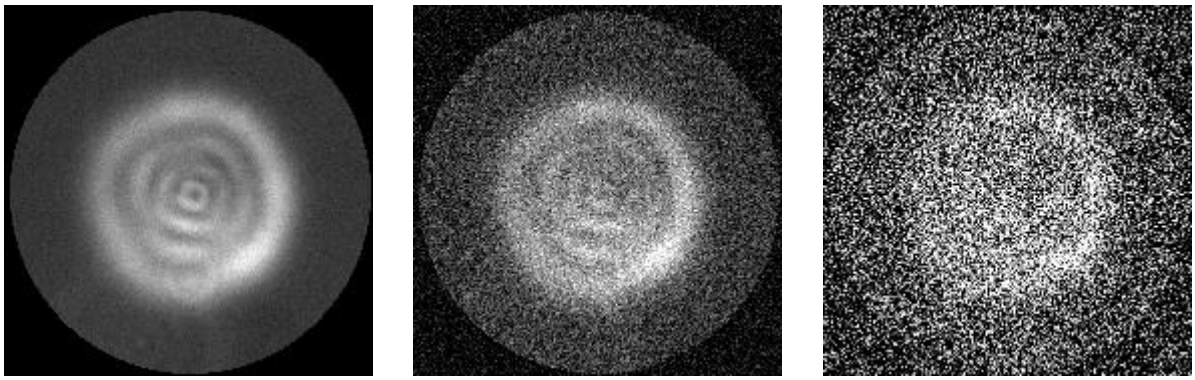


Figure 5.9: Calibration image with no extra noise, and Gaussian noise with standard deviation 0.13 and .5, respectively.

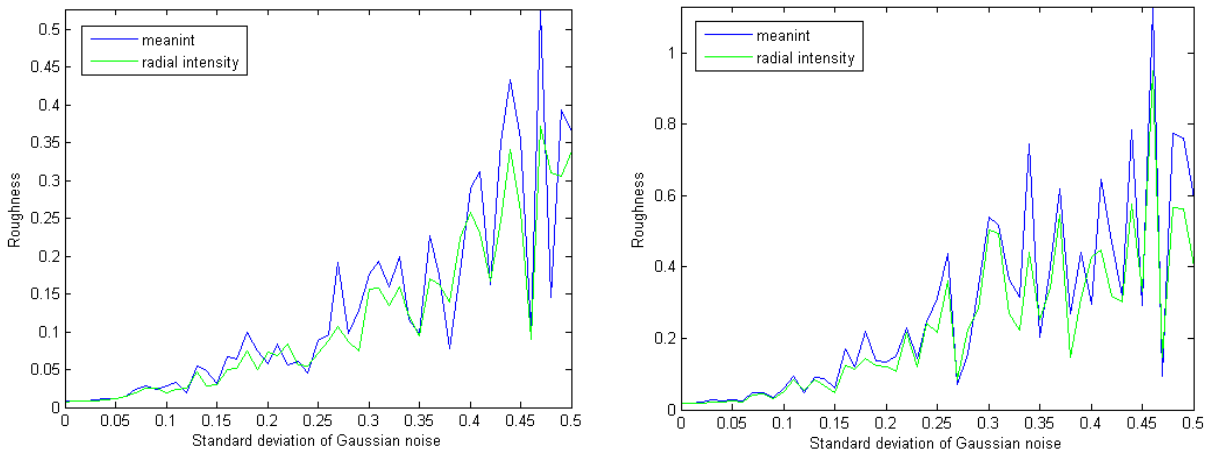


Figure 5.10: The roughness of the radius vectors for two different images are calculated with *meanint* and *radial_intensity*. Gaussian noise is added to the images. On the left: The Airy disc in Figure 5.8. On the right: The particle in Figure 5.9.

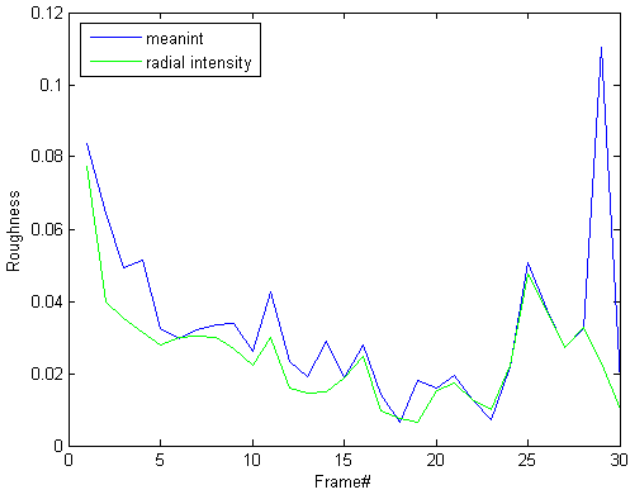


Figure 5.11: Measure of roughness for the two different approaches to obtain the radius vector. The data is the entire calibration series of images of the particle in Figure 5.9. The radius decreases with each frame.

As mentioned in section 3.2.2, the method of weighted distances for computing the radius vector requires a choice of distance Δ between each annulus. For the entire work a value of 1 is used. Increasing the value tends to smooth out details. The smallest value that was tested was 0.1. Values much below 1 gave noisy radius vectors, as fewer pixels were considered in the calculation of the intensity for each radius. Actually, a value of 0.8 or 0.9 seemed to give the smallest roughness for the radius vectors in the darkfield calibration series, but only marginally smaller than for 1. After the development of the algorithm for finding the radius of the peak, the difference of the detected radii for $\Delta = 1$ and low values such as $\Delta = 0.4$ was compared for the entire calibration series. If the noisy radius vector for $\Delta = 0.4$ was smoothed with a running average, the maximum difference of radii became less than 1 pixel.

5.4 Finding the radius of the outermost peak in the radius vector

This chapter presents the development of methods for finding the radius of the outermost peak of the radius vector. The first subsection shows two methods for finding the peak within the data set. In the second section, a more precise value of the radius is found by fitting a fourth degree polynomial around the estimate.

The procedures were tested on calibration images. The goal was to find a linear relationship between the known displacements along the *z*-axis and the calculated radius of the diffraction pattern. The performance of the methods was thus evaluated by fitting a straight line $r = a + bz$ to the *rz* plots and calculating the standard deviation σ_r. That number indicates the linearity of the *rz* plot, and should be as small as possible. The variance is given by the equation:

$$\sigma_r^2 = \frac{1}{N-2}\sum\left(r_i - (a + bz_i)\right)^2 \quad (14)$$

5.4.1 Estimating the peak from the derivative

A simple approach for determining the location of a maximum or minimum point on a curve is by using the first derivative. In a discrete set of data where the observations are one unit apart along the first axis, the equation of the first derivative is $I'(r) = I(r+1) - I(r)$. Here, I represents the intensity as a function of radius in the radius vector. In a one dimensional intensity plot of a particle imaged with a dark background, the intensities will decrease after the outermost peak. In other words, the first derivative will be negative after the outermost peak. This is generally true, except that noise can easily cause the first derivative to change sign. This is shown **Figure 5.12**. The algorithm searches for the peak from the largest radius and inwards. It stops when the value of the first derivative is larger than zero. The algorithm gets stuck at a small bump at the tail and fails to find the true peak at around radius 30. This is mostly a problem in flat regions of the curve. By also requiring the peak to be above a certain intensity level, the algorithm is more likely to find the radius correctly. **Figure 5.13** shows the peak successfully detected by requiring it to be above 0.65. The choice of value for the threshold has to be made by visual inspection of the series of radius vectors.

Another approach is to require a certain distance *Δ* between the maxima and minima, in addition to the first order derivative crossing zero. This method is less sensitive to noise than the one described above. E. Billauer provides a MATLAB implementation of the method (named *peakdet*) in [25].

The two methods have been compared in **Figure 5.14**. The red and green cross show the peak found with *peakdet* and first order derivative with threshold, respectively. In this calibration series, taken with fluorescence microscopy, the relative intensity level of the outermost ring varies all the way from almost 0.1 to 1. In order to find the peak on the left plot, the threshold has to be lower than 0.2, and is set to 0.15. On the right, the peak would have been detected correctly with a threshold of 0.6, but because it was set to 0.15, the algorithm got stuck at a tiny bump. Instead, when requiring a difference of 0.02 between the intensity at a maximum and minimum point, all visually clear peaks in this series are correctly found (not shown). Just like the lower intensity limit, this difference has to be chosen upon visual inspection and some trial and error. A large value is preferred, as it lowers the sensitivity to noise. If it is too large, though, the peak might be overlooked.

The plot on the left in **Figure 5.15** shows an *rz* curve for a calibration series made with darkfield microscopy. Here, the first derivative and a threshold of 0.65 found all the peaks down to some small radius where no outer ring could be distinguished. For the 21 images with the largest radius, the relationship to the *z* displacement seems quite linear. (The standard deviation σ_r for the 21 first observations is 0.50.) It must be emphasized that the detected peaks are data points explicitly found in the radius vector, and the radii are thus integer values. The plot on the right in **Figure 5.15** indicates that the true peak might not necessarily be the one detected by either of the two algorithms above. This may explain some of the deviations from a straight line in the *rz* plot. The estimates for the radius can be improved by performing some form of regression in a neighbourhood around it, as will be explained in section 5.4.2.

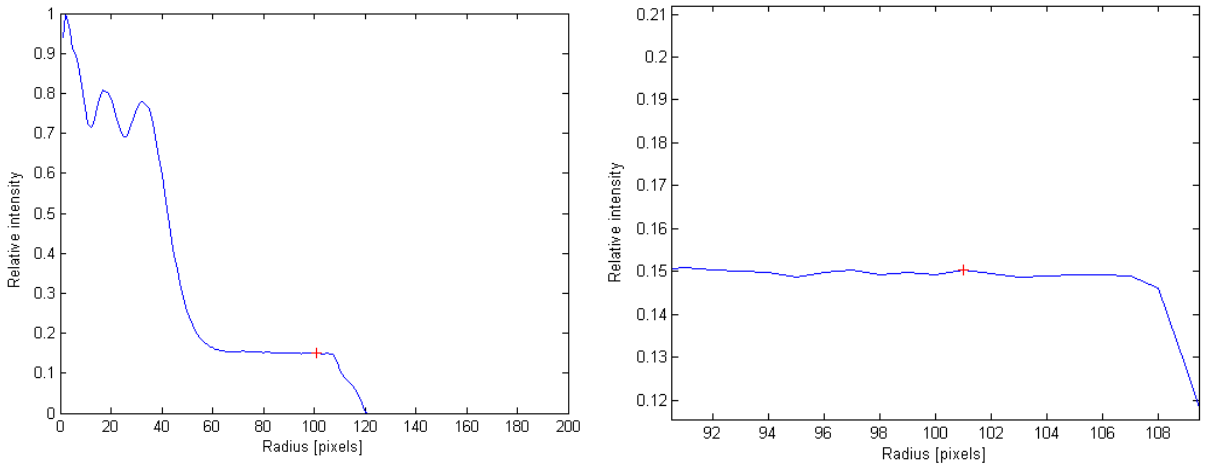


Figure 5.12: On the left: Noise on the plot causes the algorithms with only first derivative to fail. On the right: Zoom-in on the area where the algorithm got stuck.

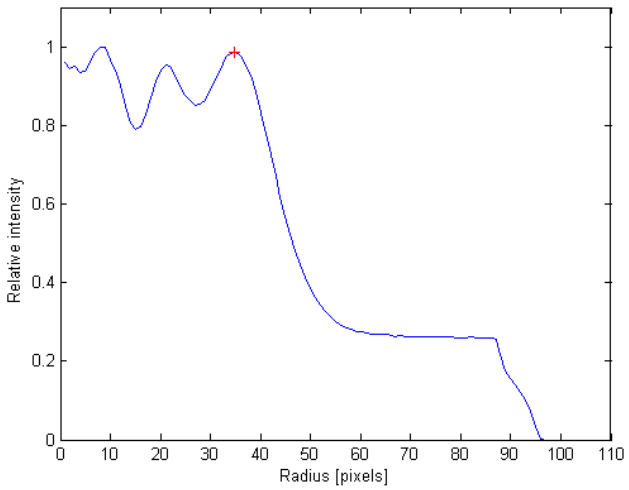


Figure 5.13: Peak correctly detected with the first derivative and a threshold of 0.6.

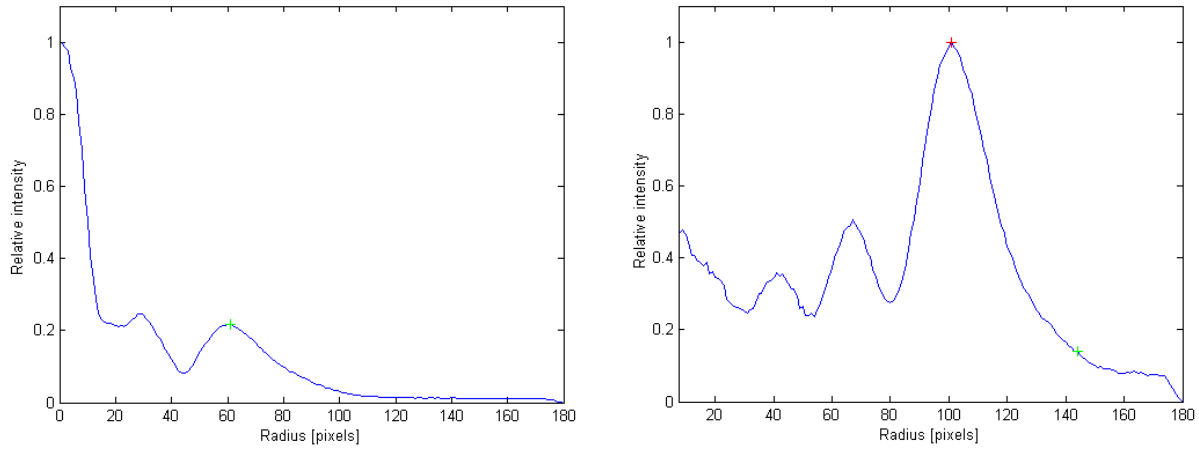


Figure 5.14: On the left: Both *peakdet* (red) and first order derivative with threshold (green) found the peak. (The green has overlapped with the red.) On the right: *peakdet* has found the peak, while first order derivative with threshold got stuck.

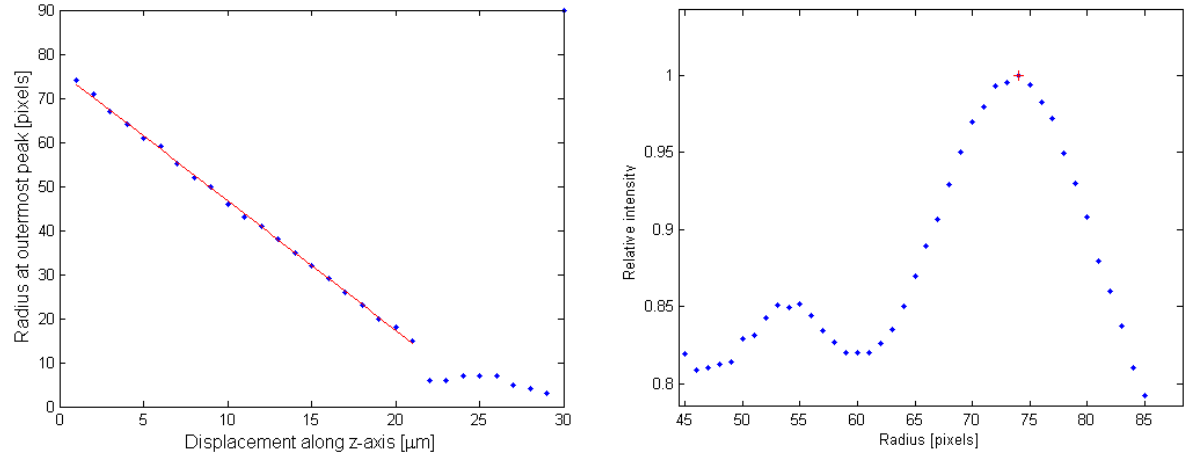


Figure 5.15: On the left: rz plot with algorithm using first order derivative and threshold of 0.65. On the right: Peak detected with peakdet with Δ = 0.01. The true peak seems to be more to the left.

5.4.2 Polynomial regression around the estimate

The previous section described two similar ways for finding the radius of the outermost peak in the radius vector, considering only points within the data set. This can be used as a starting point to find the radius more accurately. If the first estimate of the radius is r_p, an improved estimate can be found by fitting a higher order polynomial in a neighbourhood around r_p, and then finding the radius as the peak of the polynomial. Since the curve around the peak often is asymmetric, a fourth order polynomial was chosen in order to leave more room for variations of the shape of the peak. The choice of region for fitting was set as follows: If the radial distance between the first estimate of the

peak and the closest minimum to the left is Δr, the fourth degree polynomial is fitted in the region $r_p \pm \Delta r$ using MATLAB's *polyfit*. With built-in functions, MATLAB then solves the differentiated polynomial equated to zero in order to find the radius. The resulting *rz* plot, when the method is applied to the same darkfield series as in section 5.4.1, is plotted in **Figure 5.16**. For this method also, there is a clear linear relationship between the radius and the *z* displacement for the 21 first images. The standard deviation has improved from 0.50 to 0.33.

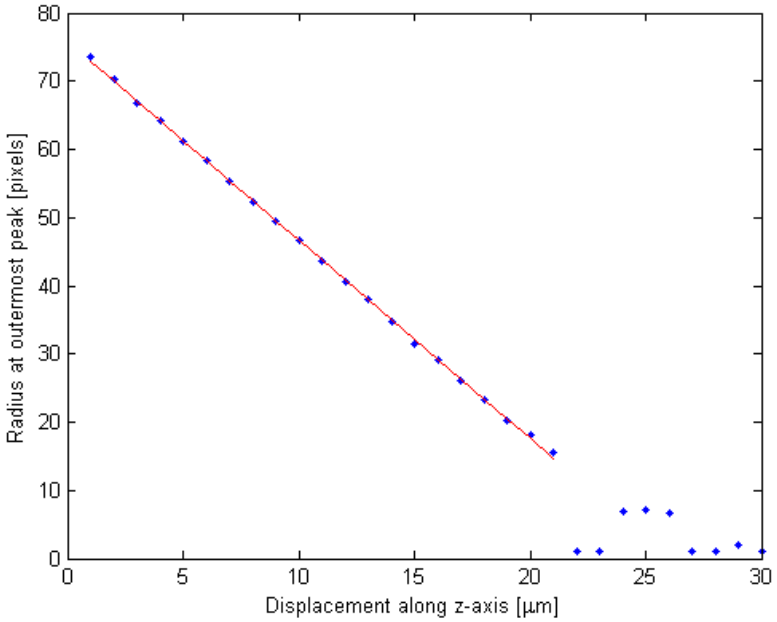


Figure 5.16: *rz* plot from algorithm that fits fourth degree polynomial around the peak. (Same as Figure 7.3)

5.5 The dependence of the radius on exact centre coordinates

In this section we look at how much the position of the centre influences the radius of the outermost ring. This is done by considering one particular image and varying the *x* and *y* coordinates in turn and calculating the radius. The image, from a darkfield calibration series, that has been used is shown in **Figure 5.17**. **Figure 5.18** shows the calculated radius when *x* and *y* vary, respectively. The centre, the radius vector and the peak have been found with *radialcenter*, *radial_symmetry* and a fourth degree polynomial, respectively. With no offset on either coordinates, the radius is 43.5 pixels. For this image, the radius is most sensitive to changes along the *x*-axis, but even an offset of ±2 pixels produces an error of maximum 0.7 pixels for the radius. Within ±2 pixels, the error is less than 0.2 pixels along the *y*-axis.

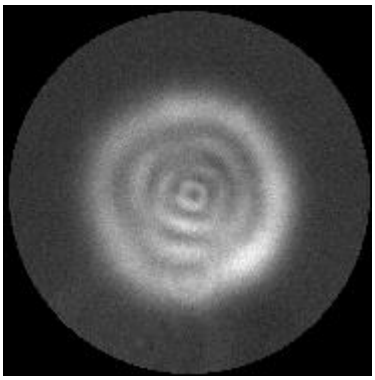


Figure 5.17: Image used to test how radius of outer ring depends on centre offset.

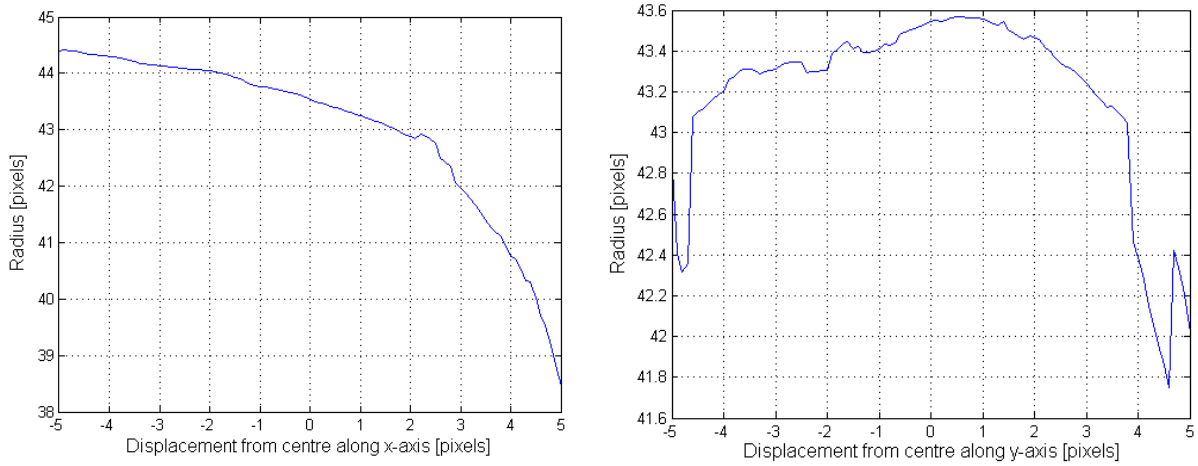


Figure 5.18: On the left and right: Radius depending on displacement from centre along the *x*- and *y*-axis, respectively.

5.6 Summary of methods

In this chapter, we have looked at the various steps in the 3D tracking algorithm for symmetric particles. The method requires off-focus images where diffraction rings are clearly visible. With this algorithm, information about all three coordinates can be obtained from a single image for each time step. The position in the plane, the *x* and *y* coordinates, are found by calculating the centre of the particle. Upon isolation of the particle in the image, i.e. avoiding disturbing objects, the centre coordinates are found with the procedure *radialcenter*, which uses radial symmetry. The *z* coordinate is found by considering the size of the diffraction pattern. The size is defined by the radius of the outermost peak in the ring pattern that forms when the particle is imaged off-focus. The size varies linearly with the displacement along the *z*-axis. The radius of the outermost peak is found by first transforming the 2D image to a one dimensional radius vector. A coarse estimate of the position of the peak is found with the function *peakdet*, which uses the first order derivative in addition to requiring a certain distance between the maximum point and the closest minimum. The radius is then more precisely estimated by finding the maximum point of a fourth order polynomial fitted around the estimated peak. Hereafter, the use of *radialcenter*, *radial_intensity* and the fourth degree polynomial is termed the *final tracking algorithm*. It was shown in section 6.5 that the radius showed little sensitivity to a ±2 pixels change of centre.

In order to use this tracking procedure for a particular microscope setup and particle size, a calibration series of the radius as function of the *z* displacement needs to be made. This is done by imaging an immobilized particle as function of increasing/decreasing *z* position. Using the algorithm described above for finding the radius of the outermost peak, a linear plot relating the radius to relative *z* movements can be made and used as a reference curve for tracking a moving particle in 3D.

6 Evaluation of darkfield setup and piezoelectric actuators

6.1 Resolution

In order to determine the optical resolution of the darkfield setup, we looked at the displacement of a point when we moved the microscope a known distance. We put a waveguide chip under the microscope and aligned it so that the waveguides pointed as vertically on the screen as possible. A point of reference near one of the waveguides was chosen. Then, the microscope was moved a known distance along the *x*-axis using a stepper motor and a LabVIEW program so that the reference point travelled across almost the entire field of view. This amounted to 17 steps of 10µm each. The two images are show in **Figure 6.1** and **Figure 6.2**, and the dark spot within the red circle shows the point of reference. Using Fiji, we put the images in a stack and drew a line from the particle in the first frame (on the left) to the particle in the second frame (on the right). Fiji showed the length of the line to be 1150 pixels, amounting to a resolution of 148nm/pixel.

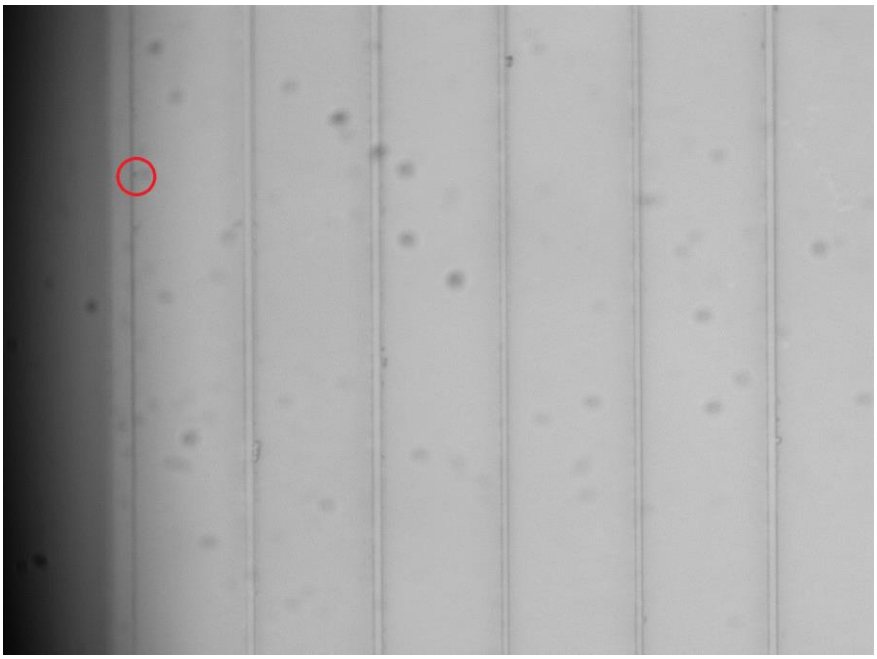


Figure 6.1: The position of the particle at the first frame. The point of reference is inside the red circle.

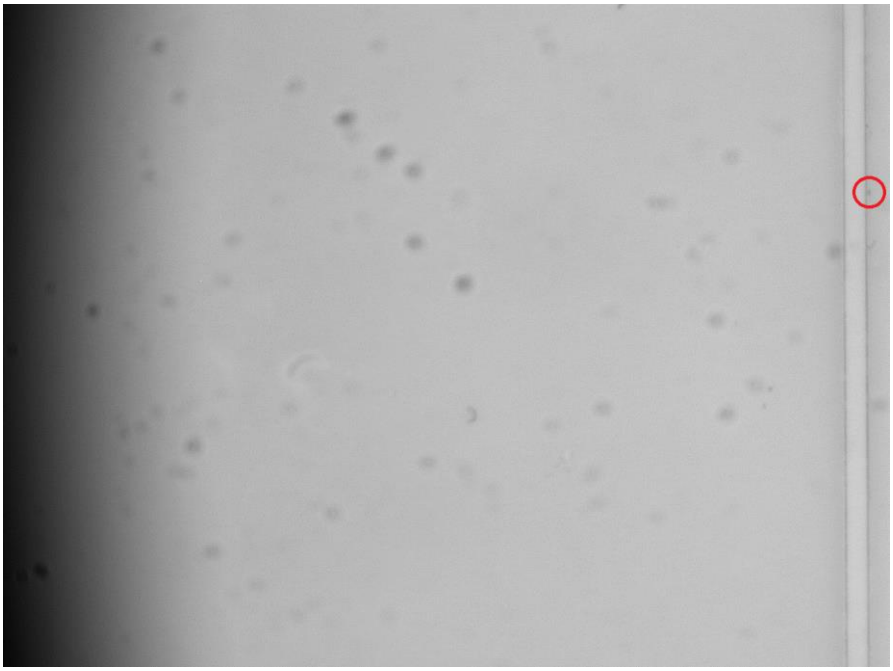


Figure 6.2: The position of the particle at the second frame. The point of reference is inside the red circle.

6.2 The darkfield microscope setup and the Melles Griot piezoelectric actuator

In this section, we look at the stability of the microscope setup that was used at the start of the project, as well as the linearity of the Melles Griot 17MPZ001 piezoelectric actuator. Here, stability means the degree to which the centre of the particle remains the same, without drift in any direction. A piezoelectric actuator from Jena is tested in section 6.3.

6.2.1 Stability

In order to determine how stable the darkfield microscope actually was, two series of 20 similar images were taken of the particle. Each series was made by capturing and saving an image manually, and then repeating these two steps. This was done without making any changes to the setup. The two series featured the particle being out of focus to a small and to a large degree, i.e. having a small and a large outer ring, as shown in **Figure 6.3**. **Figure 6.4** and **Figure 6.5** show the radius of the outermost ring in the two series, where the particle has a small and large outer ring, respectively. The final tracking algorithm was used. The *x* and *y* coordinates for the centre, when the ring is large, are shown in **Figure 6.6**.

In the case of the small ring, the radius varies with more than 0.8 pixels, while for the large one, the difference between the smallest and the largest radius is as much as 2 pixels. For the large ring, the position of the centre varies with more than 2 pixels. It should be emphasised that it was not the particle that had drifted, but the entire field of view. A planar drift would not itself cause the radius to change. The varying radius shows mechanical instability of the microscope.

Figure 6.3: Particle with small and large ring. Both imaged 20 times. (Normalized images shown)

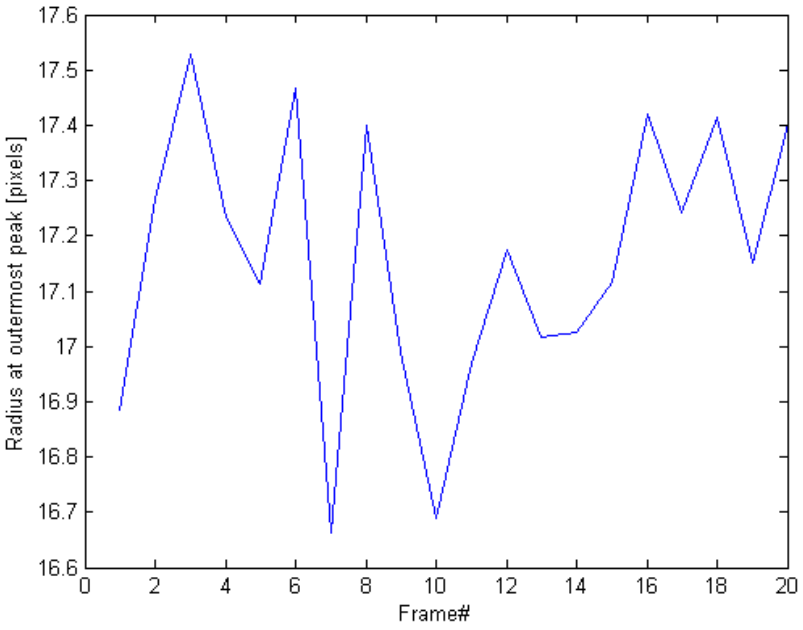


Figure 6.4: The radius of outermost peak for particle on the left in Figure 6.3. Radius varies more than 0.8 pixels.

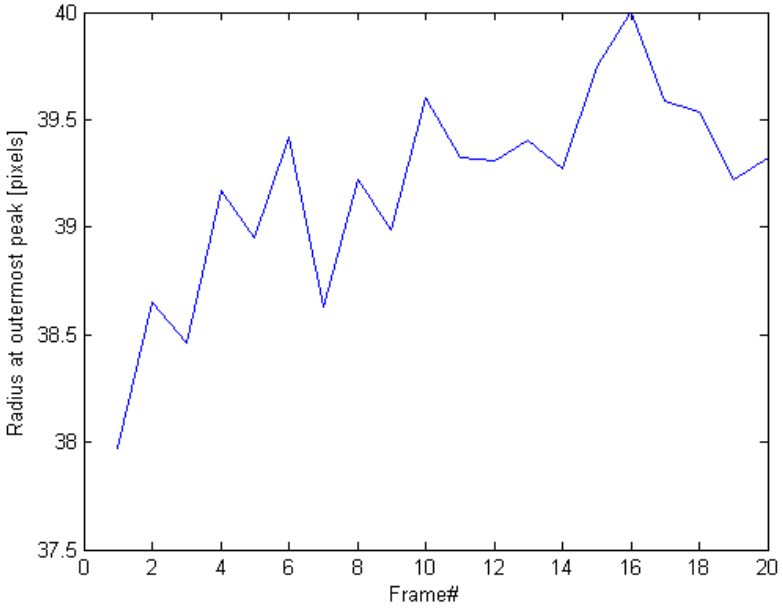


Figure 6.5: The radius of outermost peak for particle on the right in Figure 6.3. Radius varies more than 2 pixels.

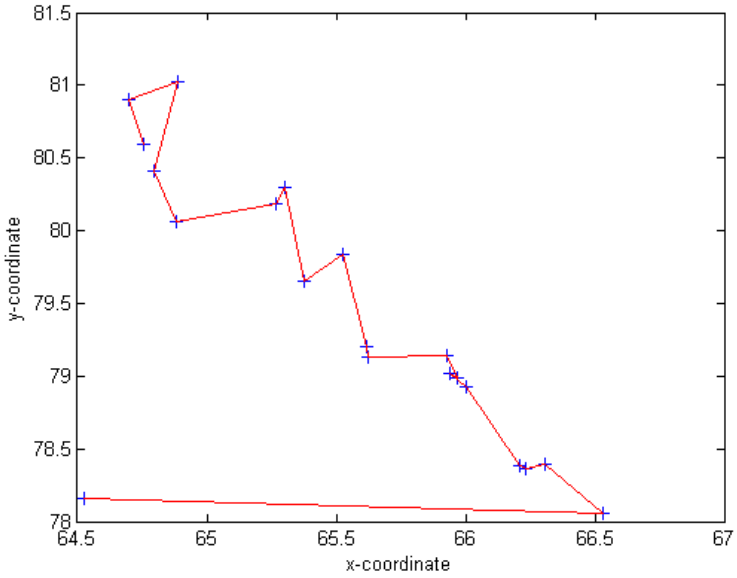


Figure 6.6: The position of the centre of the large ringed particle on the right in Figure 6.3.

6.2.2 Test of hysteresis

At the start of the project, the setup of the stage for the sample was such that it could be moved in all three directions with a Melles Griot piezoelectric actuator. Even when the final tracking algorithm was developed, the *rz* plot for a calibration series deviated largely from a straight line, as shown in **Figure 6.7**. The particle had been imaged by varying the drive voltage in steps of 1V, from 0 to 75V. The actuator did not have a button on it to select closed loop operation, which indicated that it had no built-in correction for hysteresis.

Though it was the *z* direction that was used for moving the particle towards the objective, and thus the most important one, we started off by testing the x and y direction for linearity. That was done by imaging the particle in steps of 1V, from 0 to 75V, varying one direction and keeping the two others fixed, and calculating the position of the centre of the particle as $p = \sqrt{x^2 + y^2}$. The results for the *x* and *y* direction are given in **Figure 6.8**. Both the blue curve for the *x* coordinate and the red one for *y* have a shape characteristic of hysteresis. This did not seem promising, so we rearranged the setup so that the *z* part of the piezo, which used to move the stage vertically, now moved the stage in the horizontal plane. Three series of images from 0V to 75V were taken. They revealed the disappointing result shown in **Figure 6.9**, clearly showing hysteresis. The greatest gap between the curves is at 75V between series number 1 and 3, where the difference is 21 pixels. With a resolution of 148nm/pixel, the difference of 21 pixels then corresponds to 3.1µm. This was far too much and the Melles Griot piezoelectric actuator could not be used. Another piezoelectric actuator from Jena was available in the lab, and it is tested in section 6.3.

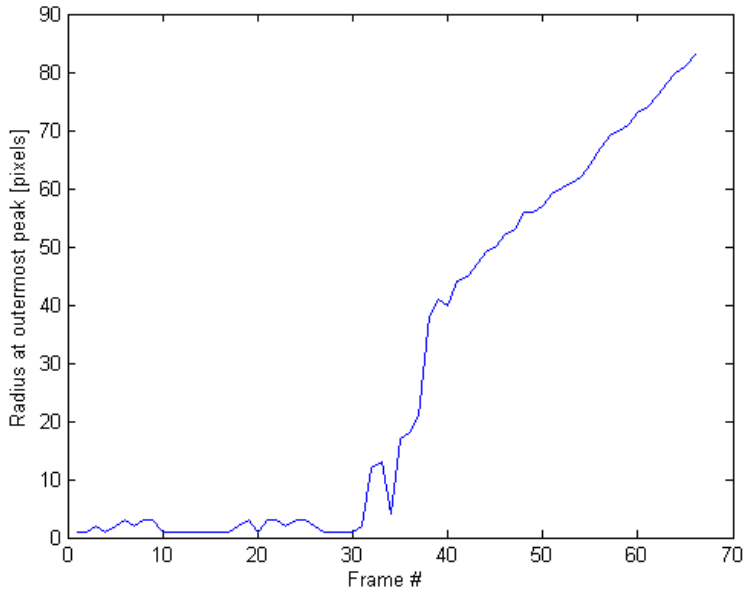


Figure 6.7: *rz* plot obtained with *radialcenter*, *radial_intensity* and a fourth degree polynomial in an image series where the open loop piezoelectric actuator was used.

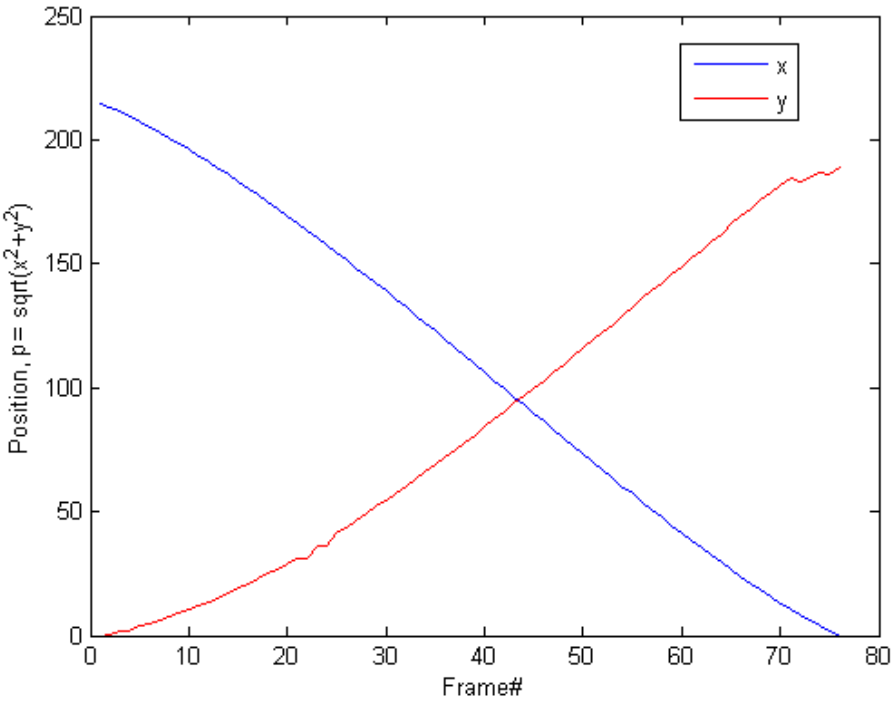


Figure 6.8: Position of the particle when varying only one direction and keeping the two others fixed.

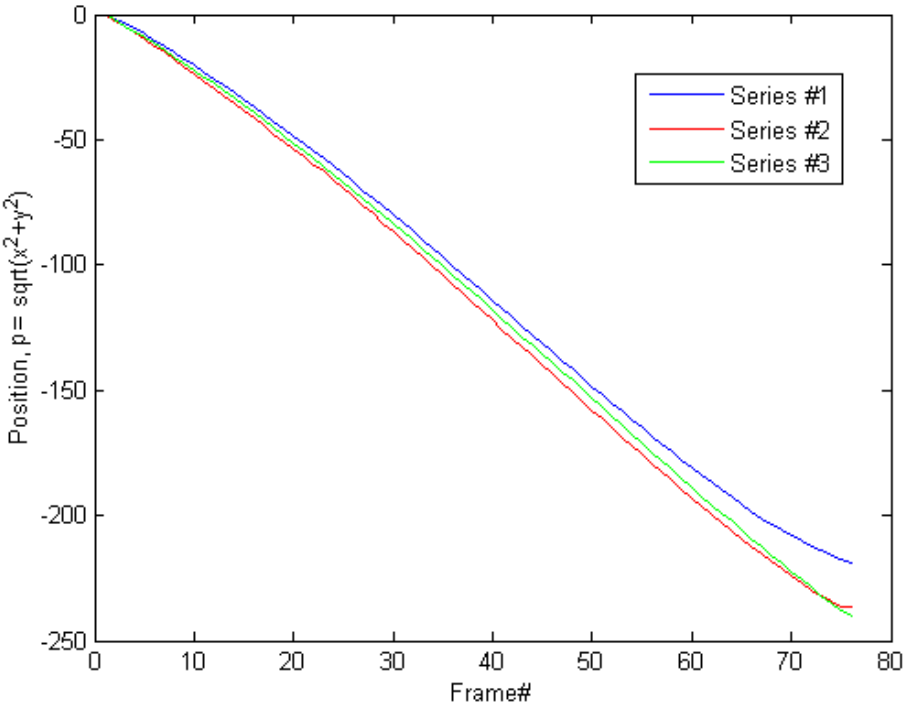


Figure 6.9: Position of the particle as the voltage of the *z* piezo increases.

6.3 Test of open versus closed loop

Since the Melles Griot piezoelectric actuator did not produce linear output, it had to be set aside. There was another piezoelectric actuator available in the lab, a Jena NV 40/1 CLE, which we tested for linearity. The actuator has a button to select "closed loop" operation, which was always activated during the experiments. Moving the particle horizontally in steps of 1V from 0V to 10V, and in steps of 5V from 15V to 100V, a series of 28 images were taken of the particle and the position ($p = \sqrt{x^2 + y^2}$) of the centre was calculated. The result is shown in **Figure 6.10**. It looks pleasingly linear. The last value at 100V is left out, as it was clearly an outlier caused by some mistake in the image taking. This actuator was used for all further experiments.

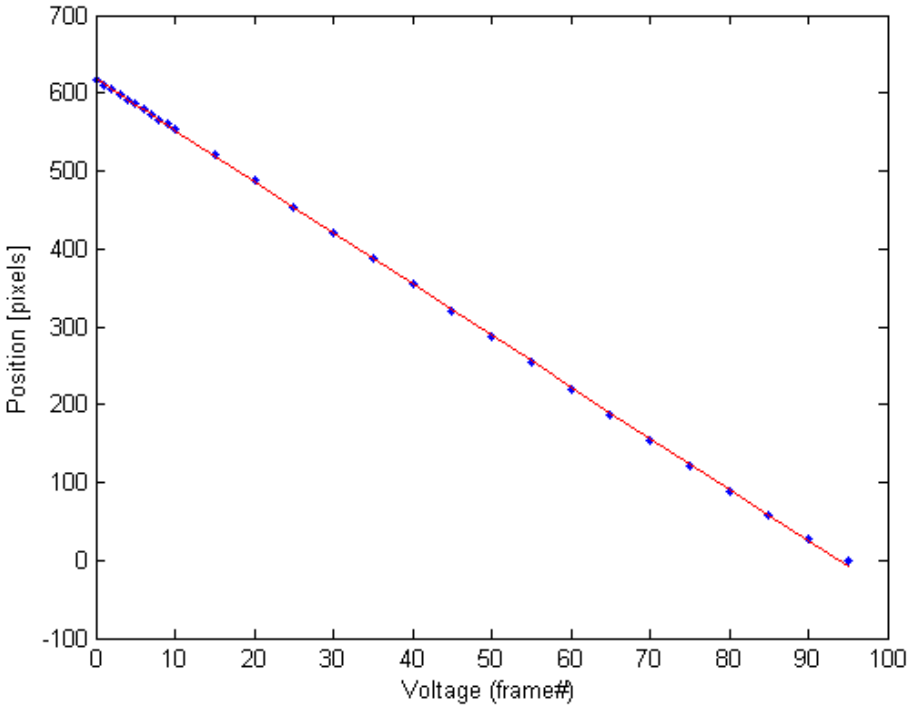


Figure 6.10: Position of particle for the Jena piezoelectric actuator. Data points are plotted blue.

7 Application of tracking algorithm to create calibration plots and track particle in video

7.1 Calibration curve with darkfield images

After the tracking algorithm had been developed and the Melles Griot piezoelectric actuator replaced with the one from Jena, a new calibration series of darkfield images was made. These are the darkfield images that have been used for presenting the tracking algorithm in section 5.3 – 5.5. The particle was imaged below focus, starting at a large radius and then moved closer towards the focus in steps of 1µm. **Figure 7.1** shows a brighfield image of the 1µm particle. The first, 10th and 22nd frame in the calibration series are shown in **Figure 7.2**, along with the respective radius vectors. The detected peaks are not shown. **Figure 7.3** shows the resulting *rz* plot found with the final tracking algorithm. At frame number 22 the algorithm fails to detect an outer ring, as the particle is too close to focus. Using the entire linear region from frame number 1 to 21, the least means squares line is equal to $r = -2.91z + 75.8$ with an error $\sigma_r = 0.33$pixels. Ignoring the negative sign, a slope of 2.91pixels/µm means that 1 pixel change in radius corresponds to a displacement of 0.3µm along the *z*-axis. We assume that when the Jena actuator moves 1µm, the precision is 0.1µm. Thus, the calibration is good enough for tracking an elevation of 3µm.

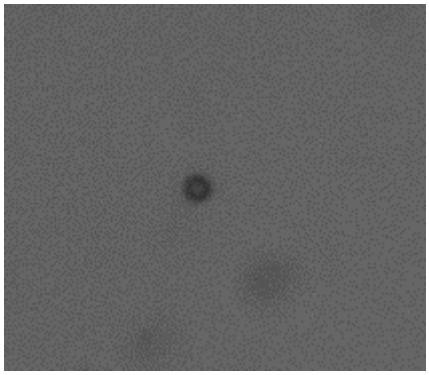


Figure 7.1: Brightfield image of the 1µm particle in this series.

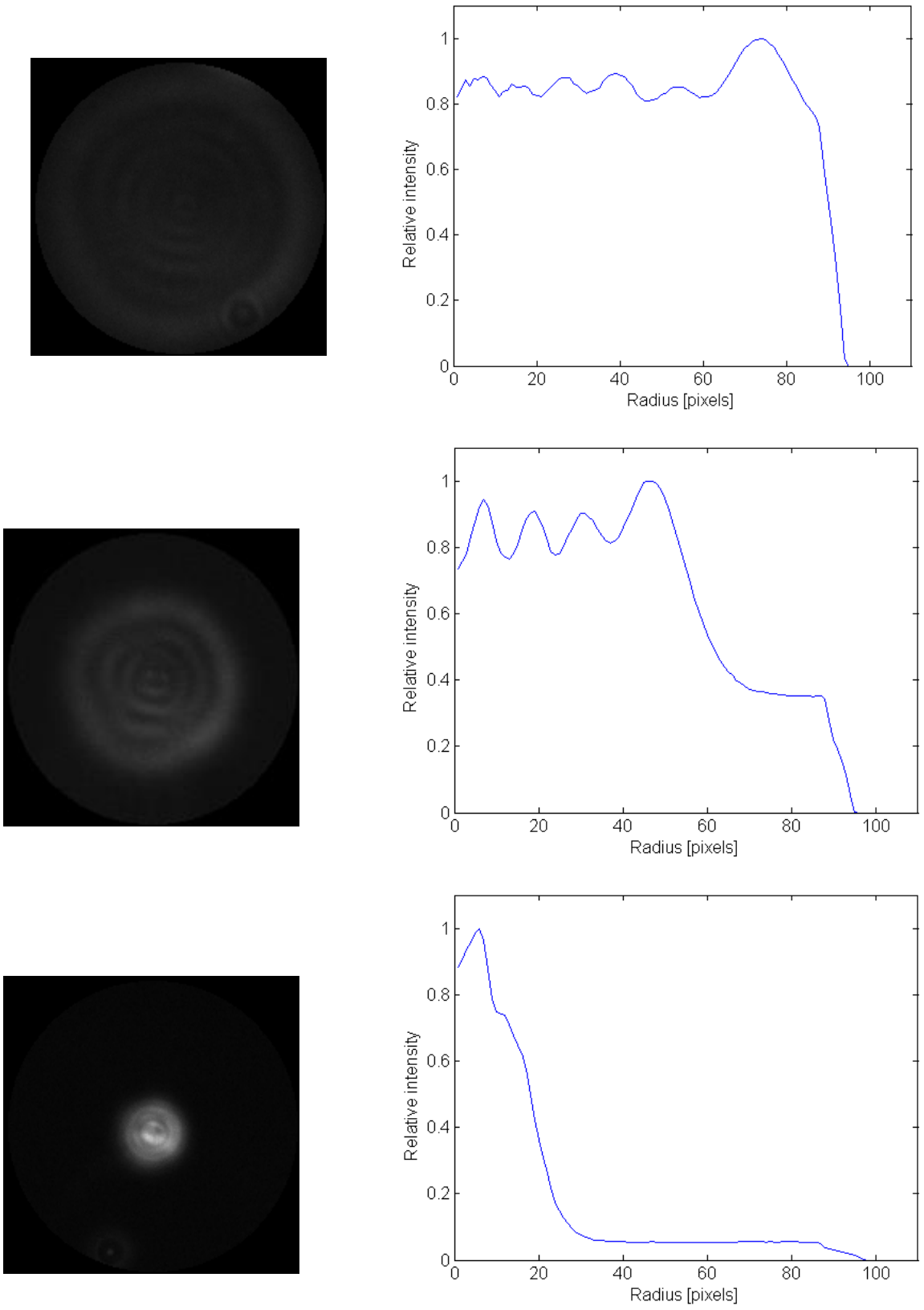


Figure 7.2: Images from the darkfield calibration series. First column: 1st, 10th and 22nd image. Second column: Corresponding radius vectors.

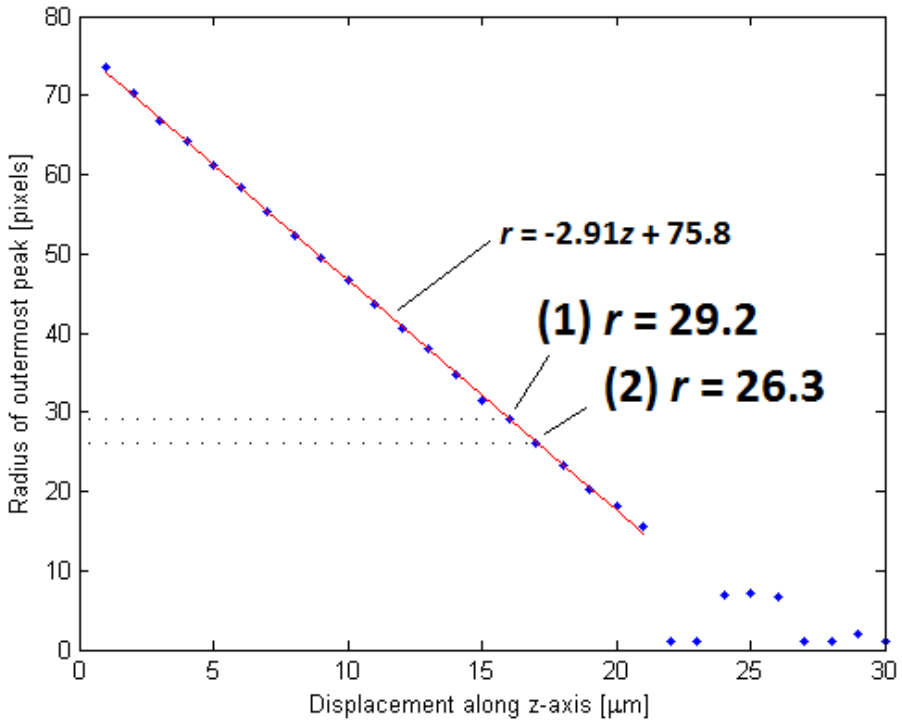


Figure 7.3: *rz* plot for the darkfield calibration series. Radius of outermost peak as function of decreasing distance from focus. (1) and (2) are points referred to in Figure 7.7.

7.2 Tracking particle trapped between waveguides from darkfield images

Several videos were recorded of a particle trapped between the waveguide ends. Many of them were not suitable for further analysis, mostly because the particle was not trapped for a long enough time, or because two particles were in the gap simultaneously. This section presents the results of the tracking algorithm used on a video where the particle was successfully trapped. **Figure 7.4** shows four images from the video. The particle is travelling along the waveguide in A), and is just about to enter the gap in B). In C), the particle is trapped between the waveguides. D) shows the particle as it drifts away after the laser has been switched off.

As can be readily seen in **Figure 7.4**, there is far more background noise within the gap than around the particle in the calibration series in **Figure 7.2** This is due to light scattered from the waveguide and diffraction patterns from other particles nearby. The video used here was the ninth we made. When recording the very first, there were fewer particles near the gap. As the laser used for trapping remained on, particles moved along the waveguide to the gap. None of the preceding eight videos

were usable, for reasons just mentioned. Many particles had, by the time of the ninth video, moved near the loop. A smaller concentration of particles would have helped a little.

Figure 7.5 shows a close-up of a typical image from the video, where the yellow circle shows the border of the diffraction pattern which almost overlaps with the waveguide ends. The tracking algorithm was applied to a 28 seconds excerpt (700 frames) from the video, and the detected radii are shown in **Figure 7.6**. The region from frame number 542 to 589 is shown in **Figure 7.7**. Within this region, the radii are close to two points in the calibration plot in **Figure 7.3**, marked (1) and (2). The image and the radius vector of two time instances, where the detected radius is close to values of the calibration curve, (3) and (4), are compared with the calibration series in **Figure 7.8** and **Figure 7.9**, respectively. The radius vectors from the video are entirely different from the ones in the calibration series. The reason is that the particle is near focus and the images are saturated. What is being measured is simply noise. Defocusing to get larger diffraction patterns would not have worked, as light scattered from the waveguides would have caused too much noise. From this, we concluded that darkfield could not be used for particle tracking, and decided to set up a fluorescence microscope.

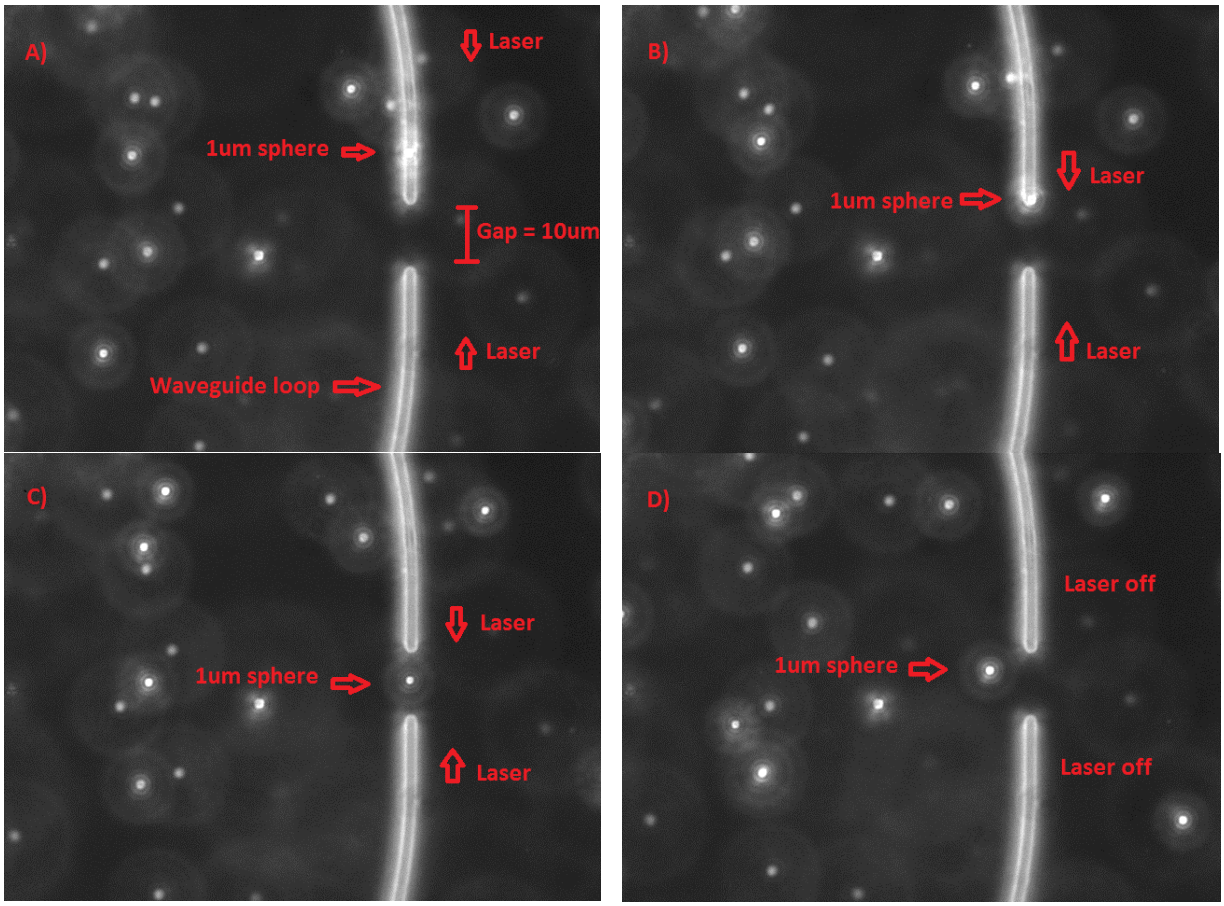


Figure 7.4: Darkfield images of 1µm particle: A) Being transported along waveguide; B) About to enter the gap; C) Being trapped within the gap; D) Leaving the gap as laser is turned off.

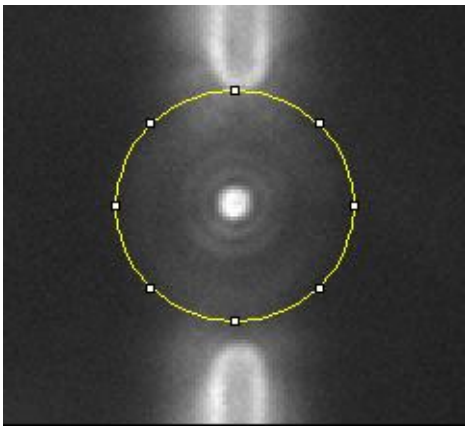


Figure 7.5: Particle trapped between waveguide ends. The yellow circle shows the outermost visible part of the diffraction pattern, which almost overlaps with the waveguides.

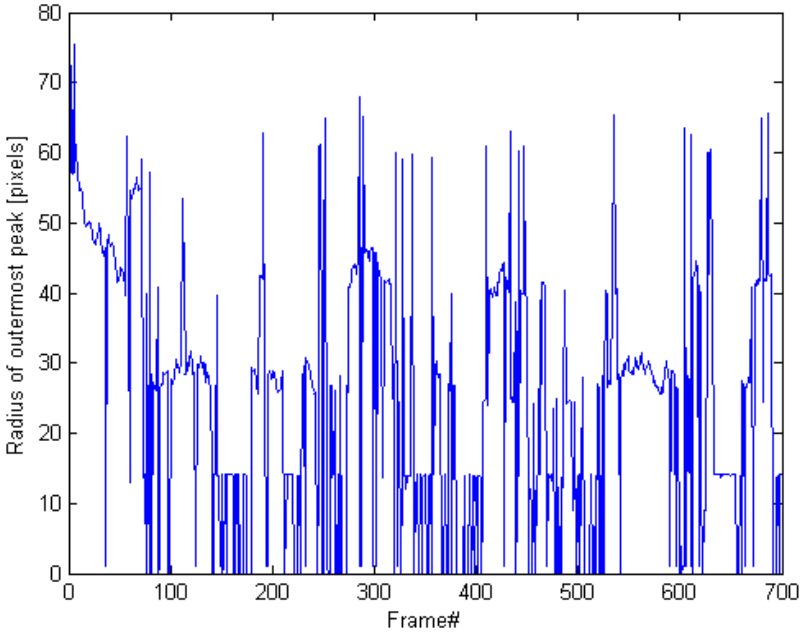


Figure 7.6: Measured radius of particle in a 28 seconds video (700 frames) where it is trapped within the gap.

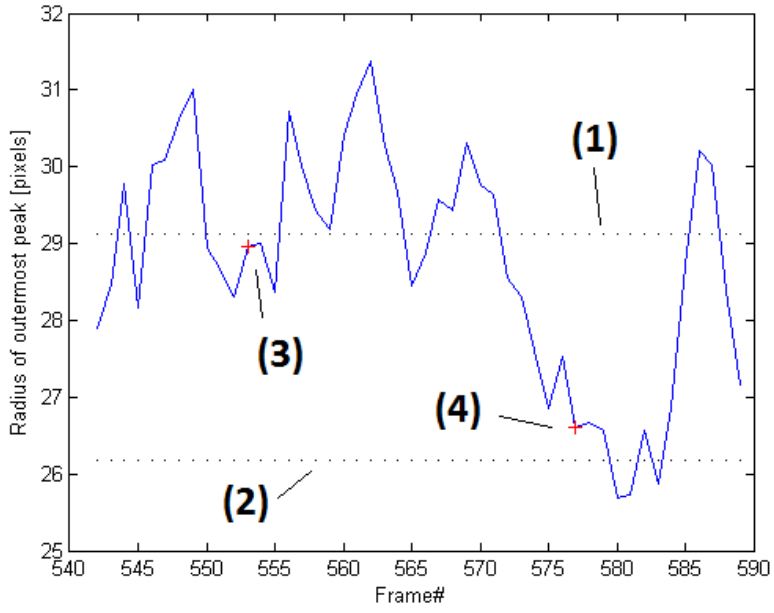


Figure 7.7: Calculated radius of the particle for a short section of the video – the region of Figure 7.6 between 542 and 589 with only small variations. The two dotted lines, (1) and (2), are radii of the calibration plot in Figure 7.3. The red crosses, (3) and (4), are instances in the video where the detected radius is close to that of the calibration series.

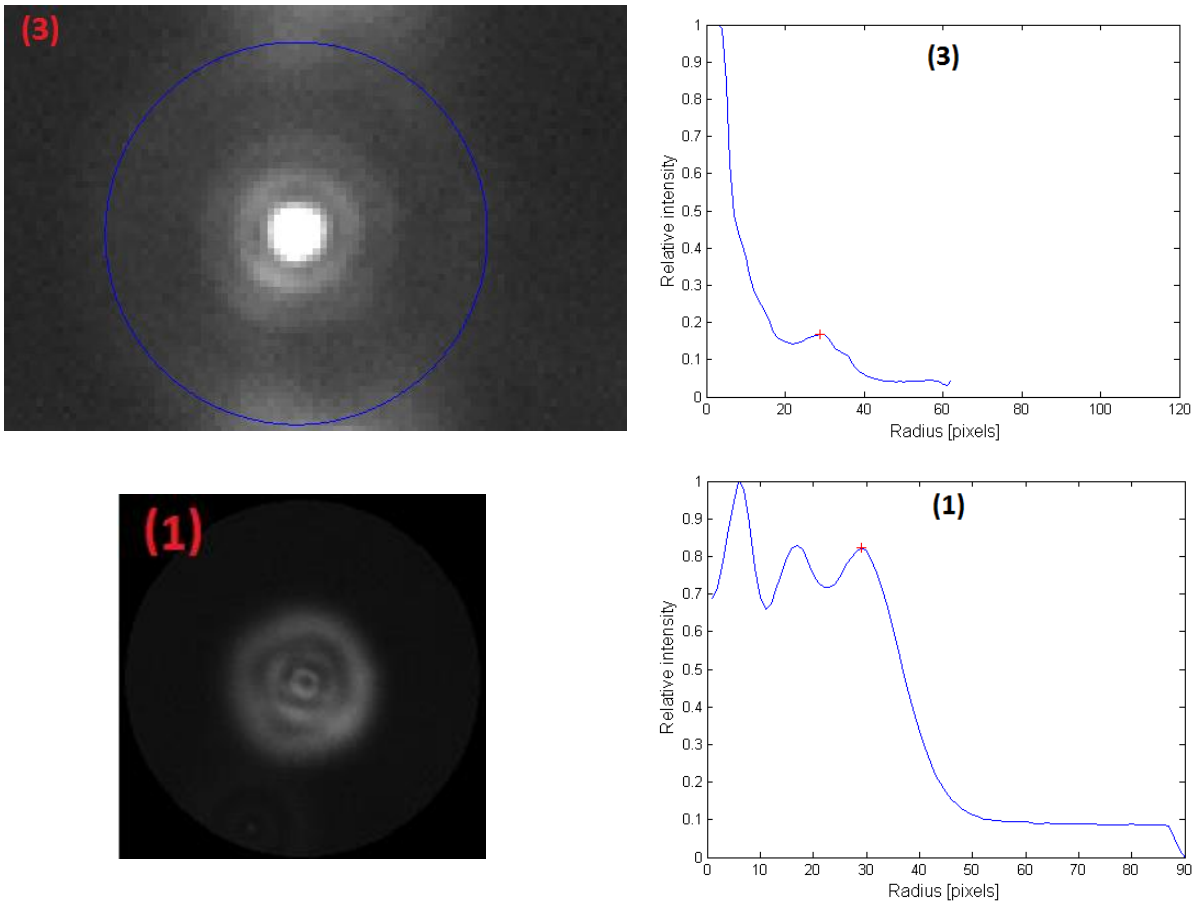


Figure 7.8: Image and radius vector from video and calibration series where the radius of outermost ring is 29.2.

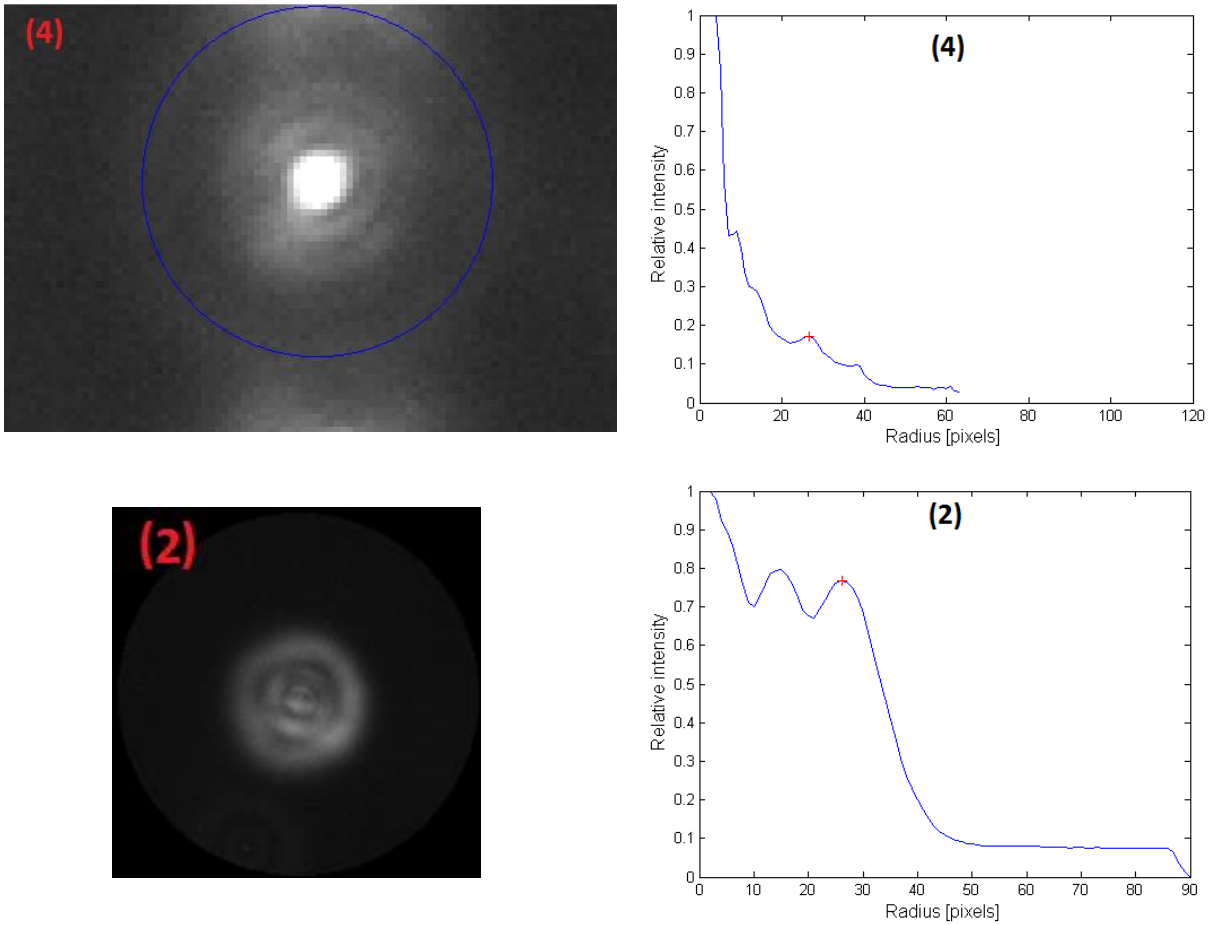


Figure 7.9: Image and radius vector from video and calibration series where the radius of outermost ring is 26.3.

7.3 Calibration curve with fluorescence images

In section 7.2, we saw that the radius vectors calculated from darkfield images of a 1µm particle trapped in the 10µm gap between the waveguides cannot be used for tracking. That is mainly because light scattered from the waveguide ends produces a lot of noise around the particle, and thus distorts the radius vector. We therefore decided to use fluorescence microscopy instead. A calibration series was made prior to trapping.

A 1µm fluorescent particle was imaged above focus, starting almost in focus and then moved further away. Images were taken in steps of 0.5µm. The first, 27th and 51st images in the series, along with the radius vectors, are shown in **Figure 7.10**. For the images with a large radius (like the 51st), the intensity before the outermost peak is quite noisy, while the curve which makes up the outermost ring has a large intensity and is quite distinct.

The radius of the peak was found with the final tracking algorithm. **Figure 7.11** shows the resulting *rz* plot. The radius was detectable for the 15th frame of the image series and upwards. The equation of the fitted line is $r = 6.0z - 9.4$ with an error $\sigma_r = 0.69$pixels. The slope of 6.0pixels/µm means that a change of radius of one pixel corresponds to a displacement along the *z*-axis of 0.2µm.

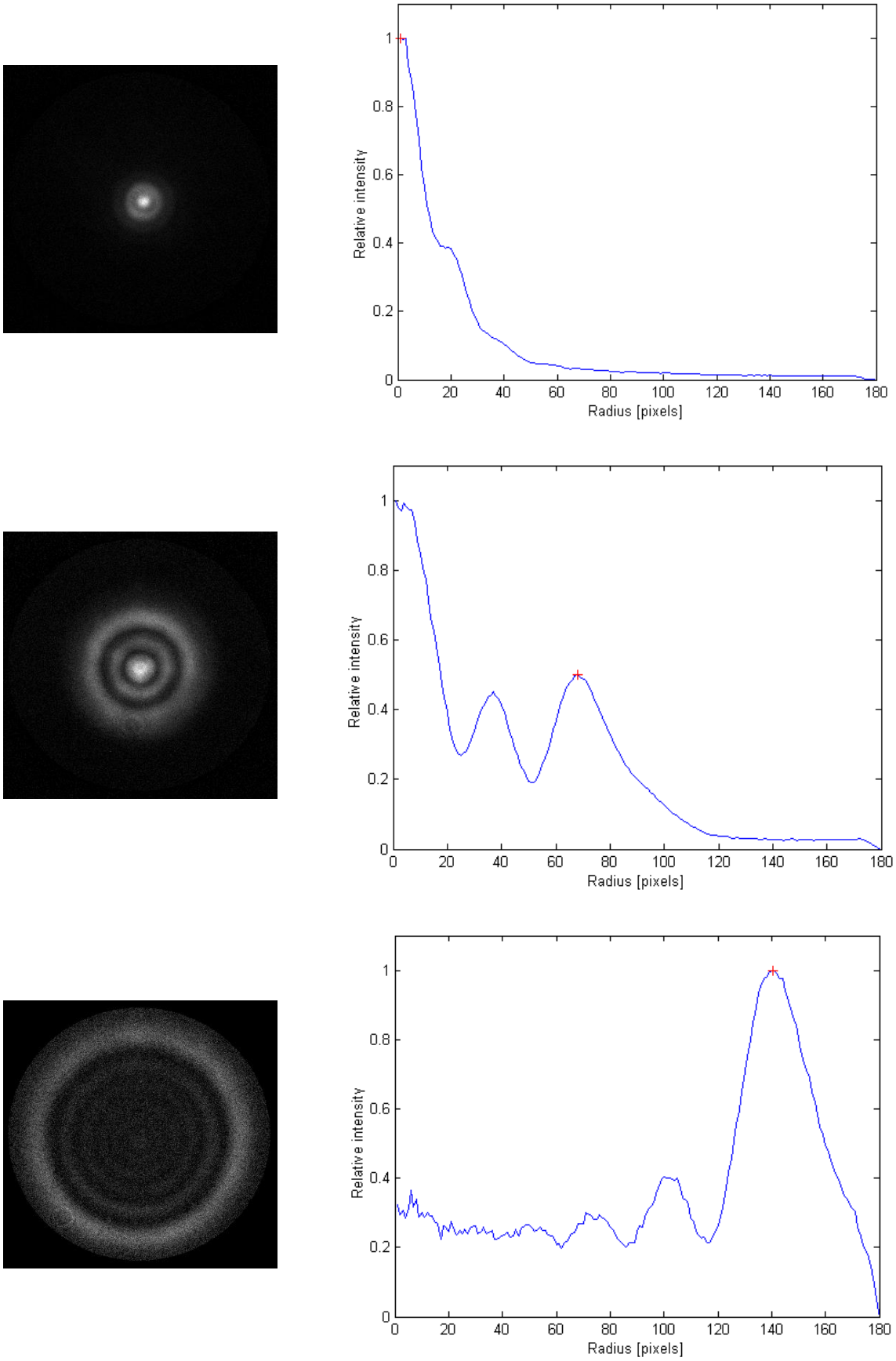


Figure 7.10: First column: 1st, 27th and 51st image frame. Background of the largest particle has been deleted for all images. Second column: Corresponding radius vectors with detected peak.

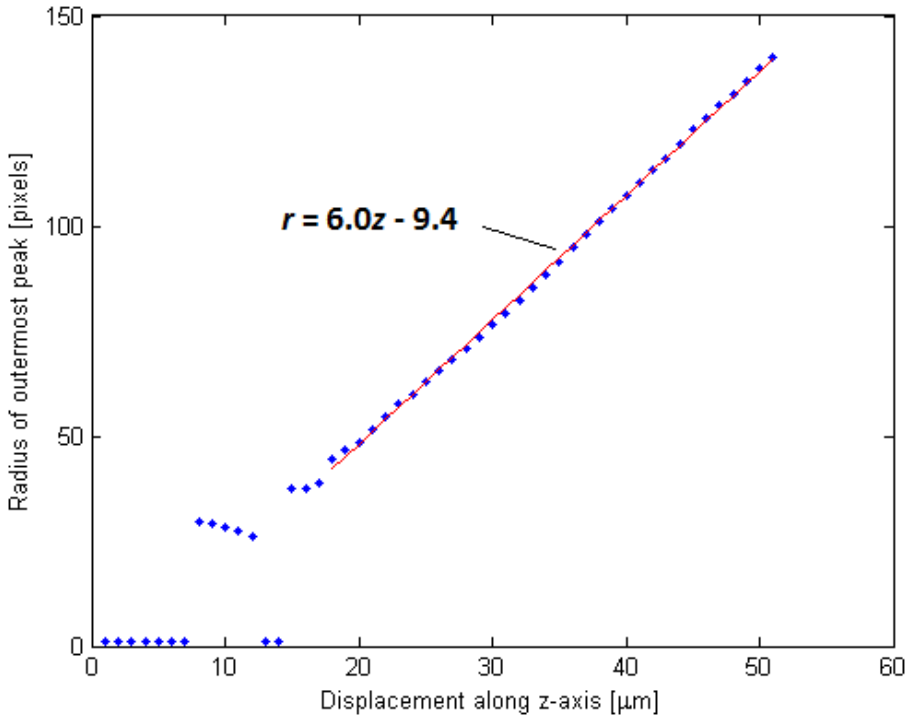


Figure 7.11: ***rz*** **plot for the fluorescence calibration series. Radius of outermost peak as function of increasing distance from focus.**

7.4 Application of tracking algorithm to fluorescence video of particle trapped and moving along the waveguide

Trapping a particle within the gap of the waveguides might take some time to achieve. The particle needs to be trapped on top of the waveguide and transported all the way to the gap. That requires the waveguide to be clean enough so that the particle is not blocked by another particle or object that has stuck to the waveguide. When the particle arrives in the gap the forces need to be strong enough so that it is held stably there. This requires the light transmitted through the waveguides to be strong enough and the light coming out of the two ends to have similar amplitude. This was achieved when imaging with darkfield, as shown in section 7.2, but thus far not with fluorescence.

In this section, the tracking algorithm is applied to six seconds of a recording where the particle is being propelled along the waveguide. The goal is to describe the displacement along the *z*-axis. From simulations, there is a strong gradient force pulling the particle down to the waveguide surface. This force is due to the evanescent field, which stretches some 150nm into the water. The first image in the series is shown in **Figure 7.12**. To illustrate the movement along the waveguide, **Figure 7.13** shows the isolated Airy disk for all 28 image frames in the series. For these small images the centres have been manually selected with Manual Tracking in Fiji.

All radius vectors look more or less similar to the one from frame number 15, shown in **Figure 7.14**. For the series of 28 images, the radius varies with 3.5 pixels, from 66.5 to 70 pixels. In the calibration series in section 7.3, we found the slope to be 0.2µm/pixel. If the smallest value of the radius in the video is subtracted from all the radii, and all values are multiplied with the slope, the relative displacement along the *z*-axis in this video is found, as shown in blue in **Figure 7.15**. It must be emphasized that the second axis does not show the actual height above the waveguide. The lowest value has been set to zero only to see the changes more clearly. The stability test of the microscope in section 6.2.1, where 20 images were taken without changing the setup, showed that the radius can easily vary two pixels. That is more than half of the observed displacement in this video.

The particle seems to be moving downwards along the *z*-axis. This is most likely due to some small angle between the objective and the waveguide. If we assume that the particle is stuck to the waveguide, the fluctuations are only due to errors related to the algorithm. The standard deviation of the fitted line is $\sigma_z = 74\text{nm}$.

Figure 7.12: First image in the series of the particle as it moves along the waveguide.

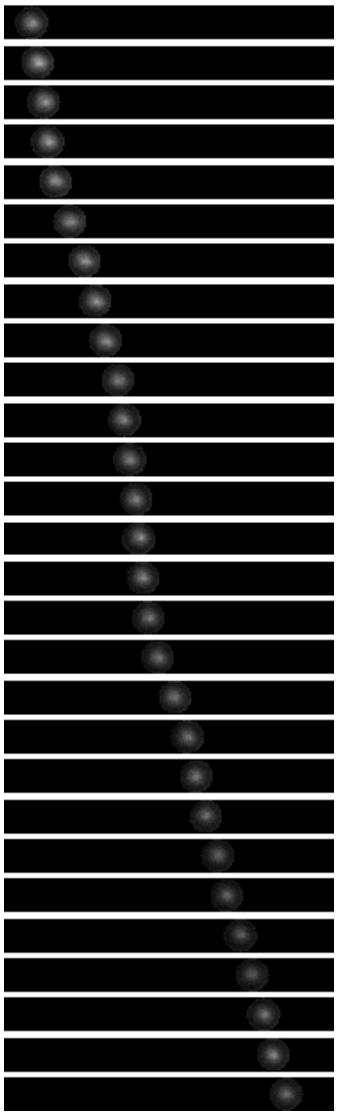


Figure 7.13: The isolated Airy disc of the particle as it moves along the waveguide.

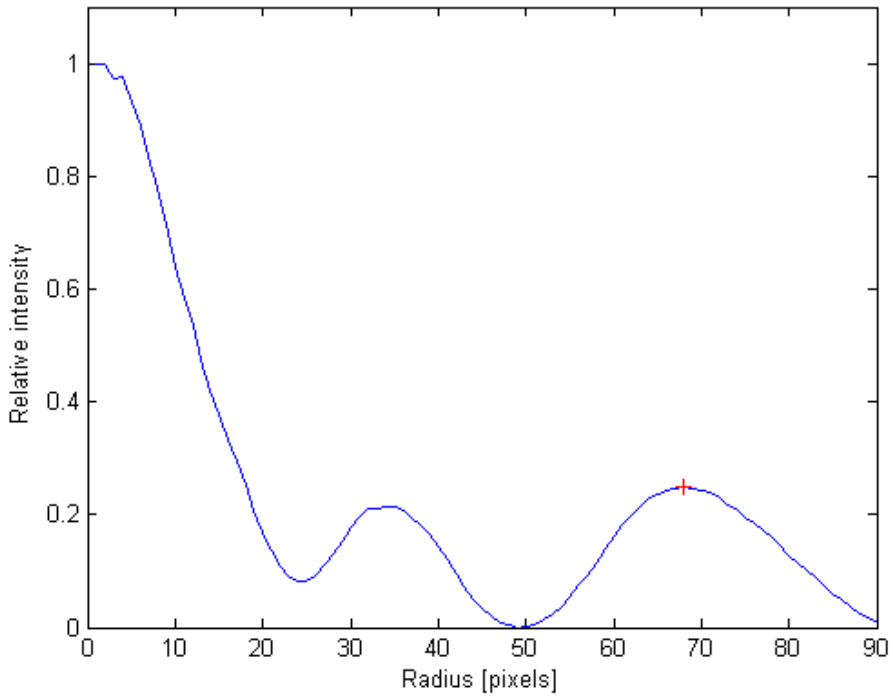


Figure 7.14: Radius vector and detected peak for frame number 15 in the video of the particle travelling along the waveguide.

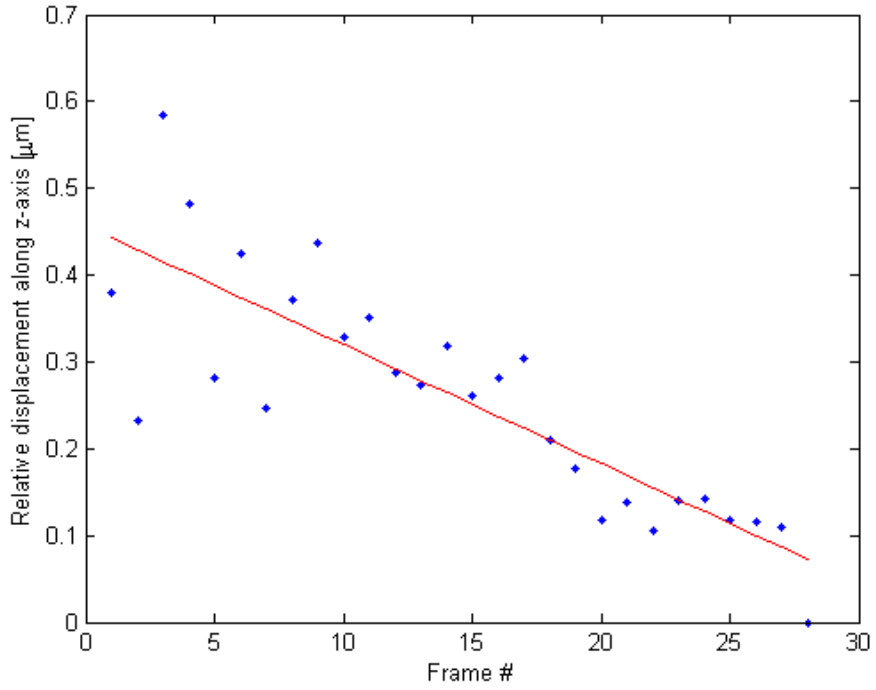


Figure 7.15: The blue dots show the detected relative displacement along *z*-axis for the video of six seconds (28 frames). The red line is fitted to the data points.

8 Conclusion

An algorithm for particle tracking in 3D has been developed in this thesis. It is based on off-focus images with diffraction patterns, and only requires images from a single *z* position. The *x* and *y* coordinates are determined by finding the centre. The *z* coordinate is found by exploiting the linear relationship between the displacement along the *z*-axis and the radius of the outermost ring. To find the z coordinate, the image of a particle is represented as a one dimensional radius vector. The outermost peak in the radius vector is detected. The radius is found by determining the peak of a fourth degree polynomial fitted around the estimate.

Darkfield microscopy was used to image a 1µm particle trapped in a gap between two waveguide ends. The tracking algorithm could not be used, due to too small diffraction rings. Increasing the defocusing would produce too much noise for adequate tracking. For the darkfield and fluorescence microscope setup, the algorithm found calibration curves with ±0.3µm and ±0.2µm precision, respectively. The precision of ±0.2µm is sufficient for tracking a particle that is elevated 3µm in a gap.

Further work can involve creating a LabVIEW program and linking it through a stepping motor to a piezoelectric actuator. This way, calibration series can be made quicker and with higher precision. Other methods for detecting the peak in the radius vector can be implemented and compared. One such option is to use kernel regression. Any ways to make the setup more stable will produce better images. Lenses, objectives and the CCD-camera can be cleaned to reduce noise from dust.

Bibliography

[1] E. Meijering, O. Dzyubachyk and I. Smal, “Methods for Cell and Particle Tracking,” in *Methods in Enzymology vol. 504: Imaging and Spectroscopic Analysis of Living Cells, ch 9*, Elsevier, 2012.

[2] R. Ghosh and W. Webb, "Automated Detection and Tracking of Individual and Clustered Cell Surface Low Density Lipoprotein Receptor Molecules," *Biophysical Journal,* May 1994.

[3] M. Wu, J. Roberts and M. Buckley, "Three-dimensional fluorescent particle tracking at micron-scale using a single camera," *Experiments in Fluids,* 2005.

[4] H. Bornfleth, P. Edelmann, D. Zink, T. Cremer and C. Cremer, "Quantitative Motion Analysis of Subchromosomal Foci in Living Cells Using Four-Dimensional Microscopy," *Biophysical Journal,* November 1999.

[5] K. D. Kihm, A. Banerjee, C. Choi and T. Takagi, "Near-wall hindered Brownian diffusion of nanoparticles examined by three-dimensional ratiometric total internal reflection fluorescence microscopy (3-D R-TIRFM)," *Experiments in Fluids,* 2004.

[6] M. Speidel, A. Jonás og E. Florin, «Three-dimensional tracking of fluorescent nanoparticles with subnanometer precision by use of off-focus imaging,» *Optics Letters,* 15 January 2003.

[7] O. Hellesø, P. Løvhaugen, A. Subramanian, J. Wilkinson and B. Ahluwalia, “Surface transport and stable trapping of particles and cells by an optical waveguide loop,” *Lab on a Chip,* 21 June 2012.

[8] B. Ahluwaliaa, O. Hellesøa, A. Subramanianb, N. Perneyb, N. Sessionsb og J. Wilkinsonb, «Fabrication and optimization of Tantalum Pentoxide waveguides for optical micro-propulsion».*Proc. SPIE 7604, Integrated Optics: Devices, Materials, and Technologies XIV, 76040W (February 11, 2010); doi:10.1117/12.840393.*

[9] B. Saleh and M. Teich, Fundamentals of Photonics, 2nd edition, Wiley Interscience, 2007.

[10] M. Abramowitz and I. Stegun, Handbook of Mathematical Functions, Dover Publications, 1964.

[11] H. Keller, K. Spring, J. Long and M. Davidson, "Molecular Expressions," 15 June 2006. [Online]. Available: http://micro.magnet.fsu.edu/primer/java/aberrations/spherical/.

[12] T. U. o. Birmingham, “The XMM Satellite Schoolpage,” Astrophysics and Space Research Group, [Online]. Available: http://www.sr.bham.ac.uk/xmm/images/waves/airy_400_246.jpg. [Accessed May 2013].

[13] "East Valley Ophthalmology," 2013. [Online]. Available: http://www.doctor-hill.com/patients/iq_lens_implants.htm.

[14] K. Spring and M. Davidson, "Microscopy U, The Source for Microscopy Education," Nikon, [Online]. Available: http://www.microscopyu.com/articles/fluorescence/fluorescenceintro.html. [Accessed June 2013].

[15] M. Abramowitz and M. Davidson, "Olympus Microscopy Resource Center," 2012. [Online]. Available: http://www.olympusmicro.com/primer/techniques/darkfield.html.

[16] M. Abramowitz and M. Davidson, "Olympus Microscopy Resource Center," [Online]. Available: http://www.olympusmicro.com/primer/lightandcolor/fluoroexcitation.html.

[17] «physikinstrumente.com,» Physik Instrumente, [Internett]. Available: http://www.physikinstrumente.com/tutorial/4_20.html. [Funnet 2013].

[18] J. Schindelin, «Fiji: an open-source platform for biological-image analysis,» *Nature Methods 9(7),* 2012.

[19] «Fiji,» [Internett]. Available: http://fiji.sc/Fiji.

[20] M. Cheezum, W. Walker and W. Guilford, "Quantitative Comparison of Algorithms for Tracking Single Fluorescent Particles," *Biophysical Journal,* October 2001.

[21] R. Parthasarathy, "Rapid, accurate particle tracking by calculation of radial symmetry centers," *Nature Methods,* July 2012.

[22] R. Parthasarathy, «ParthasarathyLab Biophysics & Soft Condensed Matter,» 2012. [Internett]. Available: http://physics.uoregon.edu/~raghu/Particle_tracking_files/radialcenter.m.

[23] Z. Zhang and C. Menq, "Three-dimensional particle tracking with subnanometer resolution using off-focus images," *Applied Optics,* 1 May 2008.

[24] M. Wand and M. Jones, Kernel Smoothing, Chapman and Hall, 1995.

[25] E. Billauer, "Eli Billauer. Freelance Electrical Engineer," 20 July 2012. [Online]. Available: http://www.billauer.co.il/peakdet.html.


UNIVERSITY
OF TROMSØ