

Jussi Kalliomäki

# **AUTOMATIC LASER BEAM ALIGNMENT USING COMPUTER VISION AND ACTUATED MIRRORS**

Master's thesis  
Faculty of Information Technology and Communication Sciences  
Examiners: Joni Kämäräinen  
Natalia Leinonen  
June 2024

## ABSTRACT

Jussi Kalliomäki: Automatic laser beam alignment using computer vision and actuated mirrors  
Master's thesis  
Tampere University  
Master's programme in electrical engineering  
June 2024

---

In this thesis, methods and software for an automatic laser beam alignment system are designed, implemented, and tested. The system utilizes actuated mirrors and computer vision interfaced with the software. Tested optimization methods were the Nelder-Mead method, Powell's method, and the Nelder-Mead method with adaptive parameters. Other possible methodologies such as mathematical modeling of optical systems and approaches based on reinforcement learning are also discussed.

The precision target for the alignment was 1  $\mu\text{m}$  for each laser beam on the system, which was achieved using a combination of Powell's method and the Nelder-Mead method. By combining the algorithms, reliable and precise alignment of the laser beams was achieved.

Keywords: Nelder-Mead method, Powells' method, Computer vision

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

# TIIVISTELMÄ

Jussi Kalliomäki: Lasersäteiden ohjaus käyttäen konenäköä ja ohjattavia peilejä  
Master's thesis  
Tampereen yliopisto  
Sähkötekniikan DI-ohjelma  
Kesäkuu 2024

---

Tässä diplomityössä suunnitellaan, toteutetaan ja testataan menetelmiä ja ohjelmisto automaattista laserkeilan kohdistusjärjestelmää varten. Järjestelmä hyödyntää liikuteltavia peilejä ja tietokonenäköä, jotka on liitetty ohjelmistoon. Testatut kohdistusmenetelmät olivat Nelder-Meadin menetelmä, Powellin menetelmä ja adaptiivisilla parametreilla varustettu Nelder-Meadin menetelmä. Käsittelemme myös muita mahdollisia toteutusmenetelmiä, kuten optisten järjestelmien matemaattista mallintamista ja vahvistusoppimiseen perustuvia lähestymistapoja.

Kohdistuksen tarkkuustavoite oli  $1\ \mu\text{m}$  jokaiselle järjestelmän lasersäteelle, mikä saavutettiin yhdistämällä Powellin menetelmä ja Nelder-Mead menetelmä. Menetelmien yhdistelmä tarjoaa luotettavan ja tarkan säteiden kohdistuksen.

Avainsanat: Nelder-Mead method, Powell's method, konenäkö

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

## **PREFACE**

Kiitos Natalia Leinoselle, joka ehdotti tätä työtä, sekä avusti ja neuvoi työn toteutuksessa.

Kiitos myös Professori Joni Kämäräiselle hyvien vinkkien antamisesta järjestelmän toteutukseen, sekä yleisesti diplomityön kirjoitusprosessissa avustuksesta.

Tampereella, 6th June 2024

Jussi Kalliomäki

## CONTENTS

1.	Introduction . . . . .	1
2.	Laser optics. . . . .	3
2.1	Laser . . . . .	3
2.2	Computer vision in laser applications . . . . .	4
3.	Low-level image processing . . . . .	5
3.1	Image acquisition. . . . .	5
3.1.1	Color image acquisition . . . . .	6
3.2	Color processing . . . . .	8
3.3	Image resizing . . . . .	8
4.	Mirror control methods . . . . .	10
4.1	Mathematical modeling . . . . .	10
4.2	Numerical optimization . . . . .	11
4.2.1	Nelder-Mead algorithm. . . . .	11
4.2.2	Nelder-mead algorithm with adaptive parameters . . . . .	13
4.2.3	Powell's conjugate direction method . . . . .	14
4.3	Reinforcement learning . . . . .	15
5.	Experimental setup and implementation . . . . .	16
5.1	Background . . . . .	16
5.2	Optical system. . . . .	17
5.2.1	Control optics . . . . .	17
5.2.2	Laser beams. . . . .	19
5.2.3	Beam steering . . . . .	19
5.3	Computer vision system . . . . .	20
5.3.1	Camera. . . . .	20
5.3.2	Image processing . . . . .	21
5.3.3	Beam detection and recognition . . . . .	23
5.4	Error calculation . . . . .	24
5.5	Control software . . . . .	24
6.	Results . . . . .	26
6.1	Initial attempts . . . . .	26
6.1.1	Algorithm improvements . . . . .	28
6.2	Choosing best control method. . . . .	29
6.3	Finding optimal parameters. . . . .	30
6.4	Final alignment precision. . . . .	32

6.5 Alignment duration . . . . .	33
7. Conclusions . . . . .	34
7.1 Future work . . . . .	35
References . . . . .	36
Appendix A: Final test errors . . . . .	40
Appendix B: Final test durations . . . . .	41

## LIST OF SYMBOLS AND ABBREVIATIONS

ADC	Analog-to-digital converter
CCD	Charge-coupled device
CMOS	Complementary metal oxide on silicon
DAC	Digital-to-analog converter
DOE	Diffractive optical element
ISO	International Organization for Standardization
MEMS	micro-electromechanical systems
PID	Proportional–Integral–Derivative

# 1. INTRODUCTION

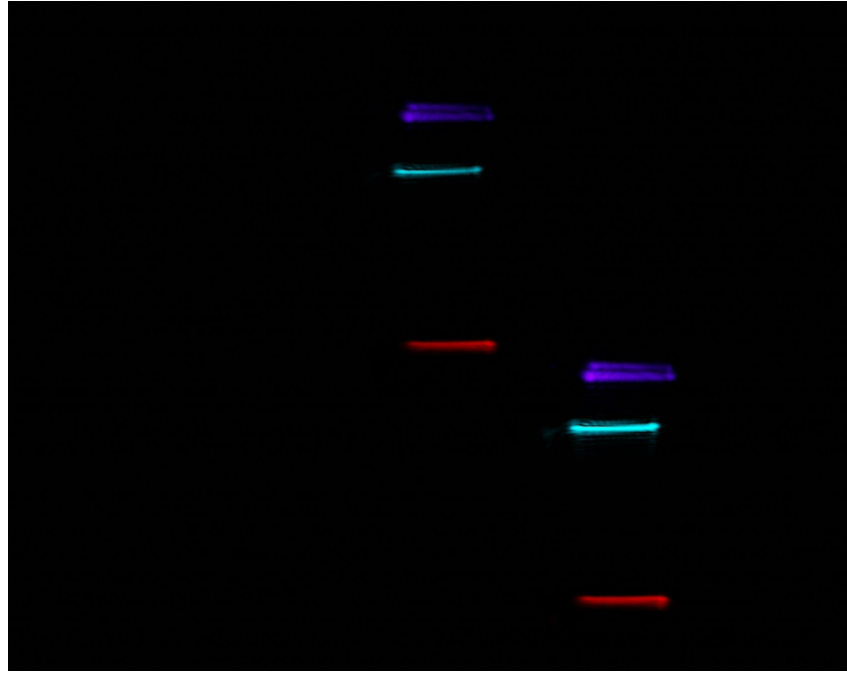
Flow cytometry is a cell research tool for characterizing cells. It is used mostly for medical research where cells need to be sorted. The laser system used in flow cytometers needs to have micrometer precise alignment for the flow cytometer to work properly and efficiently. The manual alignment of such systems is one of the main maintenance loads on flow cytometers.

In this thesis, a software system is designed and implemented. The software measures the positions of laser beams and controls them according to specified beam positions using a camera and actuated mirrors. For the implementation, we researched numerical optimization algorithms, mathematical modeling of the optical elements, and reinforcement learning-based methods, and compared their advantages and disadvantages.

The main improvement to existing solutions would be a new solution, that would reduce the need for manual beam alignment, which reduces maintenance costs and increases the system's up-time. Often the laser beams on such devices drift slowly, and this causes them to need manual aligning. With the system this thesis presents, it is possible to reduce service visits caused by drifted beams as the device can do the alignment automatically on demand. This thesis' goal is to design and implement the software for such a system, and the research question presented is "Is it possible to align laser beams to target positions automatically with sub-micrometer accuracy using computer vision and actuated mirrors?".

In Chapter 2, basic concepts of laser light are introduced with a literature review on the use of computer vision in laser applications. In Chapter 3 image acquisition's and processing's basic concepts are introduced and defined. The advantages and disadvantages of different approaches to implementing the alignment of the beams are discussed in Chapter 4.

After the background and theory for the implementation are discussed, in Chapter 5 the laser system's laser optics are described in addition to the imaging and beam steering on the device. The software implementation is also discussed on a broad level with used libraries and the limitations of the software. The laser beams of the system can be seen in Figure 1.1.



**Figure 1.1.** *Un-cropped image from the camera of the system. It includes three laser beams and their reflections caused by the laser beam splitting.*

The results and findings are shown in Chapter 6, which starts with discussing the initial tests with the numerical optimization methods, and what parts of the software implementation needed changes. After the initial attempts, the found solutions are discussed and the results from the comparison of the different algorithms and methods are presented. A bibliography and an appendix follow the main body of the thesis.

The target precision of the automatic alignment is  $\pm 1 \mu\text{m}$  for each laser beam, and the gaps between laser beams are in the range of  $200 - 400 \mu\text{m}$  in the actual output. In addition, the developed system must be reliable, as the implementation will be used in commercial devices.

## 2. LASER OPTICS

According to Joan Lisa Bromberg (1988), the first ideas for "a maser at optical frequencies" were written on paper in September 1957 by Charles H. Townes. A couple years after in 1960 Ali Javan, William Bennett, and Donald Herriot operated the first continuous laser at Bell Telephone Laboratories. After this, they have been used in multiple technology fields from communication technologies to the medical field.

### 2.1 Laser

Laser, which is an acronym for "Light amplification by stimulated emission of radiation", is a light type with unique properties compared to other types of light. The useful properties of laser include narrow frequency distribution, high intensity, and a great degree of collimation (Silfvast 2004). Lasers can be configured to have the effective wavelength span of a nanometer, which enables them to be used in very specific ways with the wavelength that is most optimal for the use case.

Laser light has high intensity, and a great degree of collimation, which means light's rays are parallel instead of scattering like in regular light. These properties allow the light to target a small area with high intensity accurately. This way they may transmit information or energy long distances with very high precision, thus they can be used in applications from space satellite communication (SpaceX 2024) to medical surgeries.

The basic principle of a laser is that the laser consists of an amplifying medium and mirror structure that reflects the light to the amplifying medium until the beam has developed enough intensity to pass through a partially transmitting mirror (Silfvast 2004). The stimulation by the passing photon is based on the theory by Einstein (1917), which theorized that because a photon can stimulate the emission to move from a lower energy state to a higher one, the photon stimulation should also be able to stimulate electron from upper energy level to lower. This form of photon emission is called stimulated emission.

The amplifying medium is a medium that has electrons that are excited in a way that the resonance from the passing electron can cause the electron to change to a lower energy state. The amplification medium needs to be chosen in a way that when the electron changes energy state, a photon is emitted so that the energy is conserved (Silfvast 2004).

According to the Planck-Einstein relation, the energy of the photo

$$\Delta E = hv, \quad (2.1)$$

where  $h$  equals to Planck's constant, and  $v$  is the frequency of light (Einstein 1917). From this, it can be concluded that to achieve stimulated emission for a certain wavelength, the amplification medium needs to be chosen so there is an energy difference between states such that it matches the wanted wavelength.

## 2.2 Computer vision in laser applications

Using computer vision to analyze laser beams is not a unique idea, but quite a rare one. There are articles about using computer vision with lasers for different purposes, but there are only a few where the laser light is directed to the image sensor for analysis. Most of the articles use lasers to illuminate points of interest. One of the articles where laser light is directed to an imaging sensor is (Xu et al. 2021), where Xu et al. used computer vision for focus detection of a laser machining system.

In their application Xu et al. used a CCD (charge-coupled device) sensor to detect defocus direction and distance in femtosecond laser machining application. They used multiple Gaussian peak function fitting with the function

$$F(x) = \exp\left(-\frac{x^2}{2\sigma_c^2}\right) + A \exp\left(-\frac{(x-p)^2}{2\sigma_s^2}\right) + A \exp\left(-\frac{(x+p)^2}{2\sigma_s^2}\right), \quad (2.2)$$

where  $A$  represents the height,  $p$  the position, and  $\sigma_s$  the width of the ring. Fitting this function on the sensor's image allows them to analyze this fitting function's coefficients to recognize the energy distribution of the focus pattern.

Most applications including lasers and computer vision use lasers to highlight points of interest, like in (Xianguo et al. 2018) where laser is used to illuminate conveyor belt. The conveyor belt is then imaged to detect inconsistencies on otherwise solid laser lines using image processing. Therefore it seems like using computer vision for laser beam quality evaluation and control is quite a new research field.

### **3. LOW-LEVEL IMAGE PROCESSING**

This chapter provides an exploration of the fundamental principles of computer vision. According to Richard Szeliski (2010), in computer vision, an attempt is made to describe the world that is seen and to reconstruct its properties like shapes and color distributions. This can be utilized for anything from simple pattern recognition like the laser beam identification implemented in this thesis to more complex things like measuring distances to objects for navigation use cases like modern autonomous cars do.

#### **3.1 Image acquisition**

For image acquisition, there are multiple different camera sensor technologies, but according to Szeliski (2010), the most common technologies are charge-coupled device (CCD) and complementary metal oxide on silicon (CMOS) cameras. The traditional CMOS camera was outperformed by CCD cameras when CCD technology was created. However, since then CMOS sensors have been improved such that today they are commonly used in digital cameras (Szeliski 2010). Multiple various schemes of both technologies have been created since then to enhance specific aspects. For example, there are versions where the coverage of the photosensitive area is improved.

In the older CMOS sensors, the photon hitting the sensor would directly affect the conductivity of the photodetector, which generates small voltage differences that are gated and amplified. After this, the individual signals are read by utilizing multiplexing (Szeliski 2010). After this, there have been multiple improvements to the basic idea of the CMOS sensor. The first of the improvements was to add a transistor to reset the signal between measurements. This makes the signal dependent on only the absorption of the current measurement, and previous measurements cause no noise. Further improvements added a PIN diode, which allows the decoupling of the photo-receptor to increase sensitivity and reduce the thermal noise. (Maître 2017)

CCD cameras work by having photosensitive "wells" called photosites, where photons accumulate electrical charge during the exposure time. The photosites are in a grid with guard regions, which are doped to have opposite charges between them. After the accumulation phase, the accumulated electrons need to be transferred to the analog-to-digital converter (ADC) to acquire the digital image. This is done by altering the potentials of the

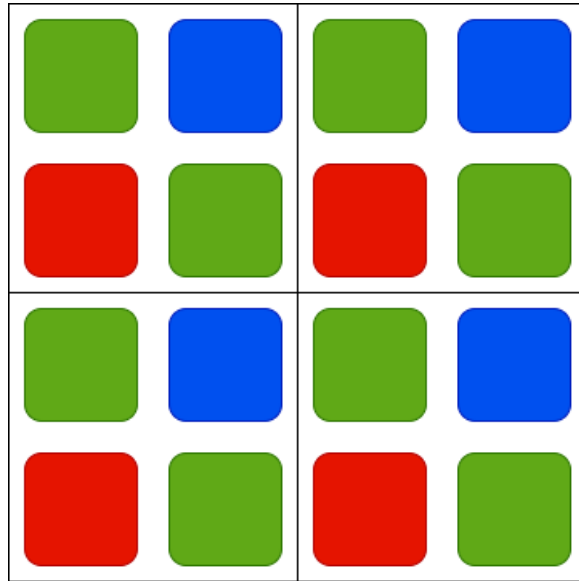
well-grid, which causes the electrons to move towards the wanted direction cyclically until they reach the voltage drain that converts the charge to voltages, which is then converted to digital form by the ADC converter. (Maître 2017)

There are a couple of advantages of CCD sensors over CMOS sensors. One of them is that as they use common ADC converters for all photosites or pixels, the noise is homogeneous across the image. An additional benefit of the CCD sensors is that the pixel readout circuit requires only a very small space. This improves the area of photon acquisition. The final advantage of CCD is that the sensor base is quite simple. This allows for achieving smaller pixel sizes and therefore higher number of pixels in smaller sensors. (Maître 2017) This is important in modern technologies that utilize miniaturization, like mobile phones and laptops.

### **3.1.1 Color image acquisition**

On digital cameras, the color acquisition is not a trivial thing. Almost always there are no three colors captured by each pixel. Instead, there is a chromatic array on top of the sensor pixels, where each pixel has one color that it acquires. When each pixel captures one color, there is a need for post-processing of the colors on the image to calculate the RGB value for each pixel. According to Maître (2017), the simplest form of color acquisition is to have vertical or horizontal bands for each color of red, green, and blue, but the most common filtering configuration on consumer cameras is the Bayer filter.

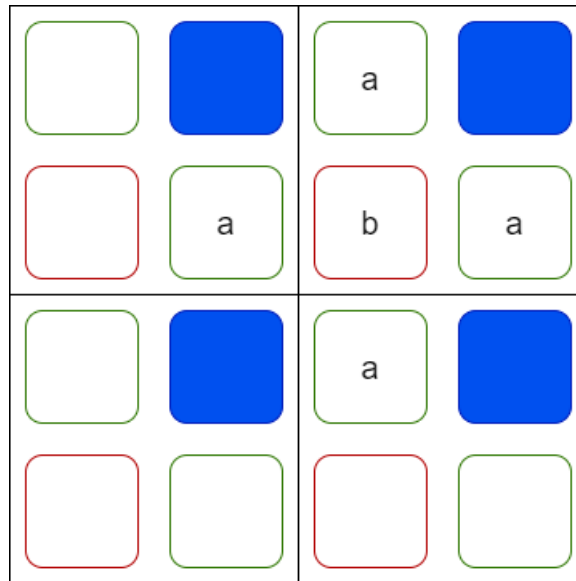
Bayer filter was invented by Bayer (1976), and has still retained its place in modern digital cameras. The filter's pixel pattern is drawn in Figure 3.1, where it can be seen that the green color is favored. This choice is made, as it was observed that the human eye is more sensitive to finer details in luminance that is closest to the green channel (Maître 2017).



**Figure 3.1.** Bayer filter pattern

After the image with the color filter array has been acquired, the missing color values need to be interpolated to all the pixels, so that each pixel has values for red, green, and blue channels. This interpolation is called demosaicing (Szeliski 2010). Bayer's color filter array is the industry standard for most consumer cameras, but for demosaicing there have been multiple techniques and solutions. For simple and more traditional algorithms Longere et al. (2002) compares their features and performance well, but there have also been progress (Tan, Chen, and Hua 2018) in using convolutional neural networks in image demosaicing.

The most straightforward approach for demosaicing implementation involves interpolation, notably utilizing the efficient and straightforward method known as bilinear interpolation. For blue and red channels the bilinear interpolation, which means using a kernel that is linear in  $x$  and  $y$  directions, would be implemented by calculating the mean of the two neighboring pixels of the same color for the pixels marked with  $a$ , and the mean of the 4 same colored neighbors on pixels marked with  $b$  in Figure 3.2.



**Figure 3.2.** Reconstruction of the Bayer filtered blue channel by interpolating the values for direct and diagonal neighbors.

### 3.2 Color processing

Often the first step of image processing is processing the colors. One common color processing task is highlighting or analyzing only specific colors which is called color slicing. Another common color processing task is balancing the intensity of all colors to be at the same level and at the same time increasing the contrast of the image by processing the histogram (Gonzalez and Woods 2007).

Colored images have a couple of commonly used color models: RGB which stands for red, green and blue, CMY which stands for cyan, magenta and yellow, and HSI which stands for hue, saturation and intensity. In RGB and CMY the three dimensions of the color model represent the intensity value of the base colors. For these color models, the easiest color slicing technique would be considering only one of the color channels.

### 3.3 Image resizing

There are multiple different reasons for needing to resize an image. There might be a need for image pyramids, which consist of differently-sized images of the base image. This may be useful in cases where one tries to match some part of the image to an image with a different resolution (Szeliski 2010). Another reason for up-sampling the image is to get more accurate measurements of objects in an image, which is why it is needed in the context of this thesis.

The basic tool for scaling up an image is using interpolation. Simply put, interpolation means estimating the most optimal value for a pixel at an unknown location when the

value somewhere in the pixel neighborhood is known (Gonzalez and Woods 2007). The simplest method is nearest neighbor interpolation, where the new pixel receives the pixel intensity from the pixel nearest to the new pixel. This method doesn't provide any new information, and according to (Gonzalez and Woods 2007), it also tends to produce undesirable artifacts, such as severe distortion of straight edges.

Nearest neighbor interpolation is rarely used in practice because of the aforementioned issues. Instead, a relatively simple bilinear interpolation method is used to achieve better results with a modest increase in required computational power. In bilinear interpolation, the intensity value of new pixel  $(x, y)$  is

$$v(x, y) = ax + by + cxy + d, \quad (3.1)$$

where the coefficients  $a, b, c, d$  are determined with four nearest neighbor pixels of the new pixel  $(x, y)$ . (Gonzalez and Woods 2007)

Bicubic interpolation is a moderately complex method for estimating pixel intensity values. The intensity value of some pixel at point  $(x, y)$  can be estimated to be the 4x4 pixel area sum

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j. \quad (3.2)$$

The coefficients  $a_{ij}$  are acquired by solving a system of 16 linear equations based on the neighborhood. Then those coefficients can be multiplied by the coordinates that are raised to powers of  $i$  and  $j$ . (Gonzalez and Woods 2007)

## 4. MIRROR CONTROL METHODS

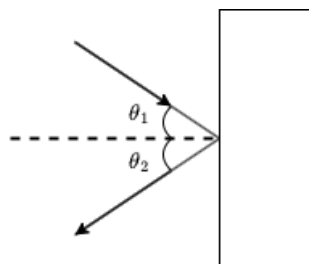
In this thesis, we focus on a mirror system composed of various optical elements including simple plane mirrors, actuated mirrors, and additional beam-shaping components. This chapter introduces multiple methods for implementing the control for the actuated mirrors.

### 4.1 Mathematical modeling

The most laborious but precise method for implementing the beam alignment is the mathematical modeling of the optical system. A complete optical system would provide a way to find the needed control values just by solving an equation that would be an easy job for a computer.

The optical properties of each optical element are needed to make a mathematical model of an optical system. An optical element is defined by Lin (2014) as a block of optical material possessing constant refractive index  $\zeta_{ej}$ . Multiple optical elements built together fixedly form an optical assembly. An optical system is defined to consist of several optical assemblies that function together.

Most of the optical elements in the optical system in this study are fundamentally simple plane mirrors, where the incident angle  $\theta_1$  equals the reflected angle  $\theta_2$ , as shown in Figure 4.1 (Walker 2008). However, all optical elements in the system do not have optimal parameters, as there are error tolerances in manufacturing the optical elements and the actuated mirrors. This causes the mathematical model of the system to have multiple variables that would need to be defined accurately for the model to be accurate in sub-micrometer precision.



**Figure 4.1.** Diagram of plane mirror

Another problem occurs when devices with the beam alignment developed in this study are produced. For mathematical modeling to work, all the devices would need to be assembled in the same way with small tolerances, and the choice for the used optical elements would also need to be locked. Vendor-locking of the components would not be a good thing for the production of the system. This is why some other control method that would be more generalized would be better for this application.

## 4.2 Numerical optimization

### 4.2.1 Nelder-Mead algorithm

As the problem can be interpreted as a minimization problem, a numerical method for solving the problem can be used. One such method is the Nelder-Mead method (Nelder and Mead 1965). The method that John Nelder and Roger Mead proposed uses the concept of a simplex. Simplex is a geometric figure that is the convex hull of  $n + 1$  vertices in an environment with  $n$  dimensions (Gao and Han 2012). In simpler terms, simplex is a generalization of the notion of a triangle to arbitrary dimensions. For example, the simplex in 2-dimensional space is a triangle, and in 3-dimensional space a tetrahedron.

Nelder-Mead method, and other optimization methods based on it, are widely known. They have been used a lot in optimization problems in multiple different fields. In the field of optics, Breckling et al. (2022) used the Stochastic Nelder-Mead simplex method (Chang 2012) for aligning compound refractive lenses. In their research Breckling et al. found out that the Stochastic Nelder-Mead method efficiently and repeatably was able to align lenses using motorized stages that controlled the compound refractive lens.

The Nelder-Mead method has been used multiple times in PID-controller's tuning (Izci 2021; Izci, Hekimoğlu, and Ekinci 2022; Blondin et al. 2018). For example Izci (2021) made a hybrid algorithm by adopting the Nelder-Mead algorithm with the Lévy flight distribution (LFD) method by Houssein et al. (2020) to develop a PID controller for DC motor's speed control. According to Izci, the proposed algorithm exploited the good global search property of the LFD algorithm, and the greater local search ability of the Nelder-Mead, and their method demonstrated efficient PID controller tuning for a DC motor speed control system. Nelder-Mead algorithm has been found to suit well in applications in the electronics field (Ekinci et al. 2023), geotechnical engineering (Yin et al. 2018) and process optimization (Ozturk, Aydin, and Celik 2018).

The method by Nelder and Mead has three steps per iteration. At first, the worst, second worst, and best indices of the simplex are recognized. Then the centroid  $c$  of the side with the two lowest values is calculated. After this, there is the transformation step, where three operations can be used to define the best way to proceed in the simplex minimization process. They are reflection, contraction, and expansion. Their visualization can be seen



The criterion for stopping the minimization process relies on the comparison of the so-called standard error of the  $y$ -values in the form  $\sqrt{\sum(y_i - \bar{y})^2/n}$  to a pre-defined value that is passed to the minimization algorithm. There have been suggestions to improving this termination criterion, for example, (Nash 1990) suggests adding a test for calculating how much the function value has decreased. This could be used in cases where the minimum value is not known.

The Nelder-Mead algorithm's performance can be influenced by the initial parameters. However, as the Nelder-Mead algorithm's iteration runs do not differ from each other when the initial simplex stays the same, the evaluation of the method needs to be done using different initial simplex between optimization runs. As the Nelder-Mead method is a heuristic search that can be used in any environment, in case the initial parameters are set close to local minima, the method may converge in that local minima. One way to find the global minimum with the Nelder-Mead algorithm in a difficult environment is to utilize restarting.

Nelder-Mead is one of the most widely used methods for nonlinear and unconstrained problems (Larson, Menickelly, and Wild 2019). It has been cited more than 38 thousand times according to Google Scholar (2024). Even though the method is popular, the convergence of the method has not been proved for the original method (Lagarias et al. 1998).

#### 4.2.2 Nelder-mead algorithm with adaptive parameters

Gao and Han (2012) stated that the Nelder-Mead method becomes inefficient in high dimensions. To improve the method's performance on problems with higher dimensions, they propose a method with adaptive parameters for the parameters for expansion, contraction, and shrinkage calculation on the Nelder-Mead method's corresponding phases.

In their method, Gao and Han proposes to choose the parameters adaptively according to the dimension  $n$  of the problem. Their suggestion is to choose transformation parameters as

$$\alpha = 1, \quad \gamma = 1 + \frac{2}{n}, \quad \beta = 0.75 - \frac{1}{2n}, \quad \sigma = 1 - \frac{1}{n}. \quad (4.4)$$

Here  $\alpha, \gamma, \beta, \sigma$  are reflection, expansion, contraction and shrink parameters.

Gao and Han claims that choosing expansion parameter  $\gamma$  prevents the simplex from distortion caused by high dimensions, and contraction parameter  $\beta$  is stated to alleviate the reduction of the simplex diameter with high dimension problems. The shrinking parameter  $\sigma$  is chosen so that with high dimensions there is no sharp reduction for the simplex diameter when compared to the usual  $\sigma = 0.5$ , which were also used in the original paper by Nelder and Mead (1965).

In their extensive testing of their ANMS (Adaptive Nelder-Mead Simplex) method, Gao and Han found that for problems with dimension  $n > 10$  their method converged 5-10 times quicker than the regular Nelder-Mead method on a uniformly convex function. When comparing ANMS and the original Nelder-Mead method on common optimization test problems with  $2 \leq n \leq 6$  dimensions, the ANMS method converged usually slower than the original. For common optimization problems with  $n > 10$ , the ANMS was usually quicker to converge when compared to the original method. (Gao and Han 2012)

### 4.2.3 Powell's conjugate direction method

A year before John Nelder and Roger Mead published their method, Powell (1964) published the minimization method called Powell's method, or Powell's conjugate direction method. Powell developed this method, as most of the solutions that existed at the time could not find the minimum from a general quadratic form in finite steps. Powell was also not satisfied in the disadvantage of the method by Smith (1962) that was slow in starting with bad initial approximation (Powell 1964).

Powell's method is not as widely used as the Nelder-Mead method and algorithms derived from it, but it has still been used in different fields for optimization. One example of Powell's method usage is work by Mota-Babiloni et al. (2018), where Mota-Babiloni et al. optimized high-temperature heat pumps performance using Powell's method to acquire lower greenhouse gas emissions. By using Powell's method, they identified optimal working conditions for the heat pumps and improved the coefficient of performance by 13% in best cases. Other fields where Powell's method has been used include astrophysics (Manthei et al. 2021), swarm intelligence algorithms (Sun et al. 2017), and fuel processing research (Zhou et al. 2022).

The method works by searching  $n$  linearly independent directions from the initial guess  $p_0$ . At first, the search directions are chosen to be linearly independent. This means that they are commonly chosen to be the coordinate directions from the initial guess. In practice, the method changes one parameter at a time. So for each  $r = 1, 2, \dots, n$  a  $\lambda_r$  is calculated from which the  $f(p_{r-1} + \lambda_r \delta_r)$  is minimum, after which the new approximation is defined for this direction  $p_r = p_{r-1} + \lambda_r \delta_r$ , until a new value  $p_n$  is achieved.

Next step is replacing  $\delta_r$  with  $\delta_{r+1}$  for all  $r = 1, 2, \dots, n - 1$ . This causes the previous  $\delta_n$  to be replaced by previous  $\delta_{n-1}$  after which the  $\delta_n$  is replaced by  $(p_n - p_0)$ . After this, the last minimization of the iteration is done by finding  $\lambda$  so that  $f(p_n + \lambda(p_n - p_0))$  is minimized.

This is the basic procedure for Powell's method. This procedure can then be iterated until proficient minimization results have been achieved. So in practice, the algorithm makes one multidimensional minimization problem into multiple one-dimensional minimization

problems, for which efficient algorithms exist.

### 4.3 Reinforcement learning

Another technology considered for the beam alignment was reinforced learning. Reinforcement learning problems involve the agent's acting in the target environment to maximize some numerical reward it is given (Sutton, Barto, and Bach 2018). The reward can be given for anything that may help the agent to achieve the target.

In this application, a temporal difference method named Q-learning was considered. It is a method where all possible actions in each state are given a score in the so-called Q-table. The training phase of such a system includes the agent exploring the environment it is in and testing the actions and while doing so, calculating new scores for the Q-table values of the actions. (Sutton, Barto, and Bach 2018)

The simplest form of Q-learning is called one-step Q-learning and it is defined by

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}) - Q(S_t, A_t)], \quad (4.5)$$

where  $S$  means state,  $A$  action,  $R$  reward,  $\alpha$  the learning rate and  $\gamma$  discount factor. It can be seen that the method updates the Q-value by adding the sum of the reward and discounted maximum Q-value of the new state which is subtracted with the Q-value of the taken action on the state. The learning rate  $\alpha$  limits how much the Q-value is raised for each tested action. (Sutton, Barto, and Bach 2018)

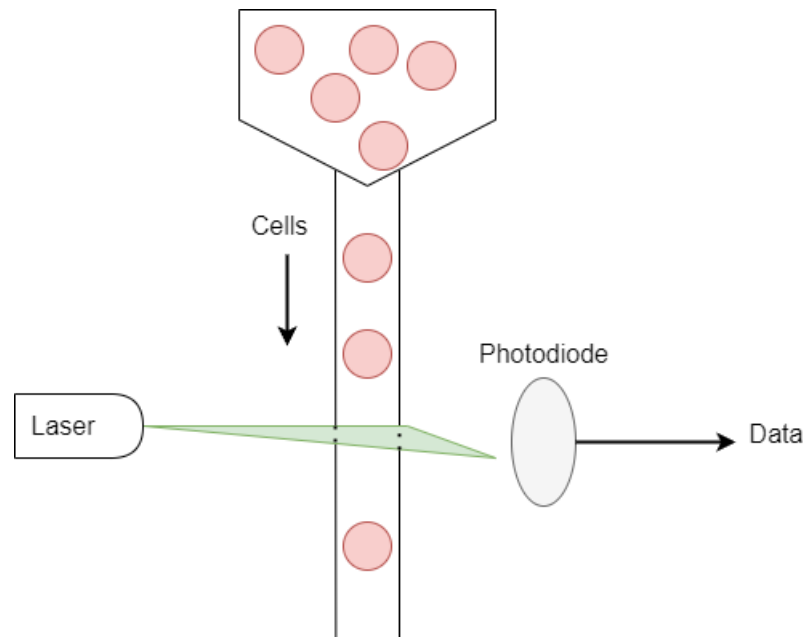
In practice, the Q-learning could be implemented by rewarding the agent when the alignment is successful. Then the agent could learn the environment by itself. Another reward behavior could be the distance between the laser beams, even though they would not be in the correct position.

Reinforcement learning could be a suitable solution if the device would not be developed to be produced in quantity. As the alignment parameters on different setups cannot be identical with a reasonable initial manual alignment of the device, each device needs to be trained individually. This would be a time-consuming part of the manufacturing process. Therefore the method was not even tested in practice and was left for future work.

## 5. EXPERIMENTAL SETUP AND IMPLEMENTATION

### 5.1 Background

This alignment system is developed for a laser device that would be the laser light producer for flow cytometry applications. Flow cytometry is a cell research method that interrogates the phenotype and characteristics of cells (Macey 2007). The basic functionality of a flow cytometer is that cells are sent one at a time through a narrow tube where they pass through laser light. The laser light passes through the cells causing the light to scatter, attenuate, or have different reflections. The scattering, attenuation, and reflections are captured with photodiodes, which produce data that can be captured with a computer. The basic structure of a flow cytometer is shown in Figure 5.1.



**Figure 5.1.** Basic principle of a flow cytometer. The cells that are studied flow through laser light. This causes scattering and attenuation which is recorded by photodiodes.

As different cells produce different scattering and reflections and have different fluorescence, the cells can be identified and their properties can be studied in multiple ways. The fluorescence can be measured simply by measuring how much of the intensity of the laser beam passes the cell to the photodiode, and what is the wavelength of the emitted

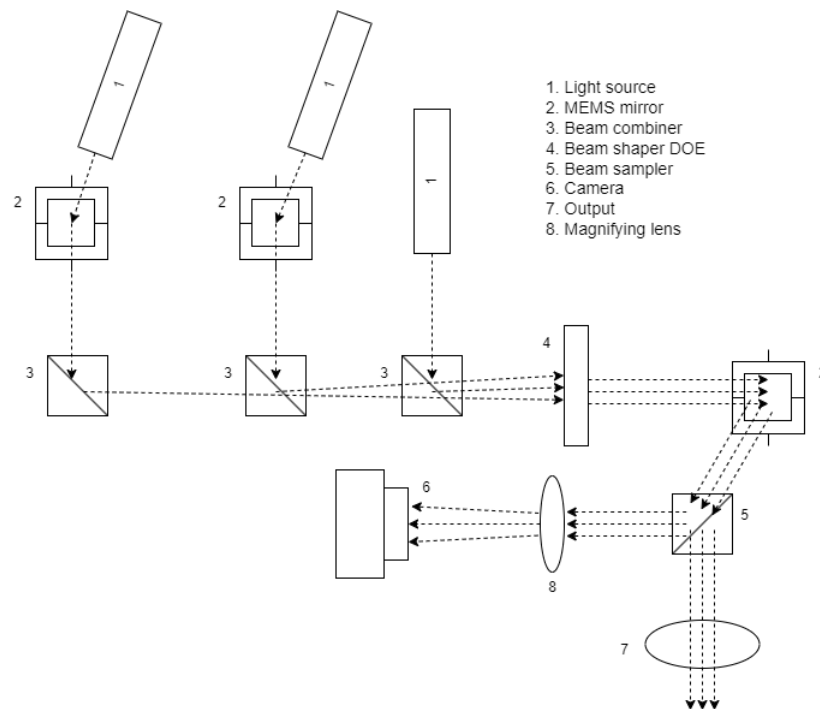
light. For measuring the scattering of the light, more photodiodes can be added to the system at locations where the scattering of the light is measured.

To measure the properties of the cells by flow cytometer, it is crucial to know the positions of the laser beams precisely. Knowing the positions of the laser beams allows knowing when the cell passes each laser light on the system, which allows collecting data about the cells through the photodiodes.

## 5.2 Optical system

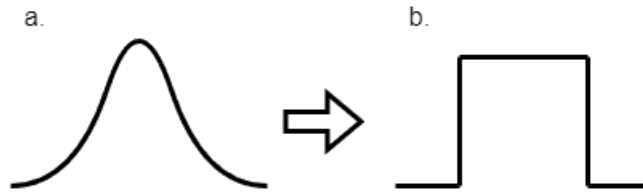
### 5.2.1 Control optics

The optical system consists of three lasers, which are formed elliptical, where the profile of the longer dimension is top-hat, and the shorter dimension is formed Gaussian. The camera image that was captured by the test setup can be seen in Figure 1.1. The optical diagram of the setup can be seen in Figure 5.2.



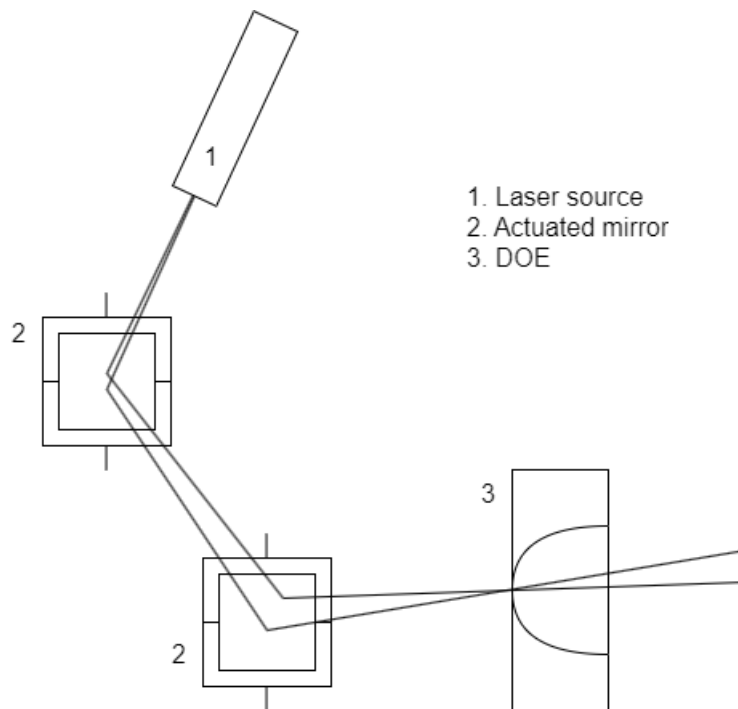
**Figure 5.2.** Optical diagram of the experimental setup

The DOE on the system is used to acquire the top-hat beam profile from the initial Gaussian laser beam profile, as described in Figure 5.3. This is important for the application of this system for research fields like flow cytometry, where a uniform intensity beam is necessary so there is uniform intensity where the flow cells pass.



**Figure 5.3.** Schematic of optimal Gaussian and top-hat beam profile intensity distributions. *a* marks the Gaussian profile, *b* marks the top-hat profile

It would be optimal if all of the laser beams cross through the DOE in the middle of it, as that is where the beam shaping is most efficient. For future work, a system could be made with two actuated mirrors for each beam as shown in Figure 5.4. Controlling each laser beam with two mirrors like this would allow the beam alignment so that the laser beams would go through the middle of the DOE to achieve optimal beam quality with the alignment functionality. The beams that do not pass the DOE through the middle of it have some imperfections on the beam profile like a non-linear top-hat profile.



**Figure 5.4.** System with two actuated mirrors for optimal beam shaping and control. This system would be able to steer the laser beam to pass through the middle of the DOE as well as control the angle at which the laser beam comes to the DOE.

The red beam is the rightmost in the diagram in Figure 5.2. It can be seen that it does not have individual control with a MEMS mirror. The two other lasers in the diagram, cyan and purple, are reflected from their actuated mirrors to their beam combiner elements, after which all three laser beams pass through the DOE. Then there is the third actuated mirror which can be controlled in the X- and Y-axis. After those elements, there is still a

beam splitter and a focus lens for the output. The beam splitter splits part of the laser beam's intensity and directs it to the camera. The focus lens is configured to focus the laser beams to the desired distance for integrating the device into flow cytometer systems.

To get the beams focused on the camera sensor, there is a focusing lens after the beam splitter on the optical path to the camera. The focal length of this lens is chosen in a way that there is optical magnification in the camera sensor when compared to the focus point of the actual output. This allows efficient usage of the whole camera sensor area for the measurements of the beams, which increases the measurement precision of the camera. In the development system, the optical magnification of the image was  $\frac{10}{3}x$ .

### 5.2.2 Laser beams

Laser beams in the tested system had the wavelengths of 405 nm, which is purple, 488 nm, which is cyan, and 638 nm, which is red, lasers. Figure 1.1 shows the beams on the camera image.

### 5.2.3 Beam steering

The beam steering on the system was done using actuated mirrors. The mirrors in the experimental setup were MEMS (micro-electromechanical systems) mirrors, and as mentioned in Section 5.2.1, only the cyan and purple laser beams had individual control mirrors. The red mirror was controlled by moving all the beams, after which the cyan and purple beams could be steered back to their intended place. MEMS mirrors work by inputting high voltages on the mirrors that cause their structure to twist because of electromagnetic forces. This causes them to have theoretically infinite precision, but in practice, the precision comes from the precision of the ADC of the control circuit. The rotation angle range in the mirrors was from around  $-5^\circ$  to  $5^\circ$  on the X- and Y-directions.

The MEMS mirrors used in the experimental setup did not have a completely linear control. Because of this, the application requires an optimization algorithm instead of just calculating the change in the control variable based on the displacement. With linear control, the beam steering and alignment implementation would have been trivial, as the movement of the beam per control value could have just been calculated. This would have made the optimization process almost instant instead of the 1-3 minutes it takes with the optimization algorithm on the experimental setup.

The control hardware in this system had a 16-bit DAC (digital-to-analog converter), which allows good precision in the controlling. As the angle control range of the MEMS mirrors was around  $10^\circ$ , the theoretical control precision on the system is  $10^\circ/65535 \approx 0,1526$  millidegrees. The MEMS mirrors used had a control system where two voltage values were inputted to set the position of the mirror on one axis. In practice, this was done by

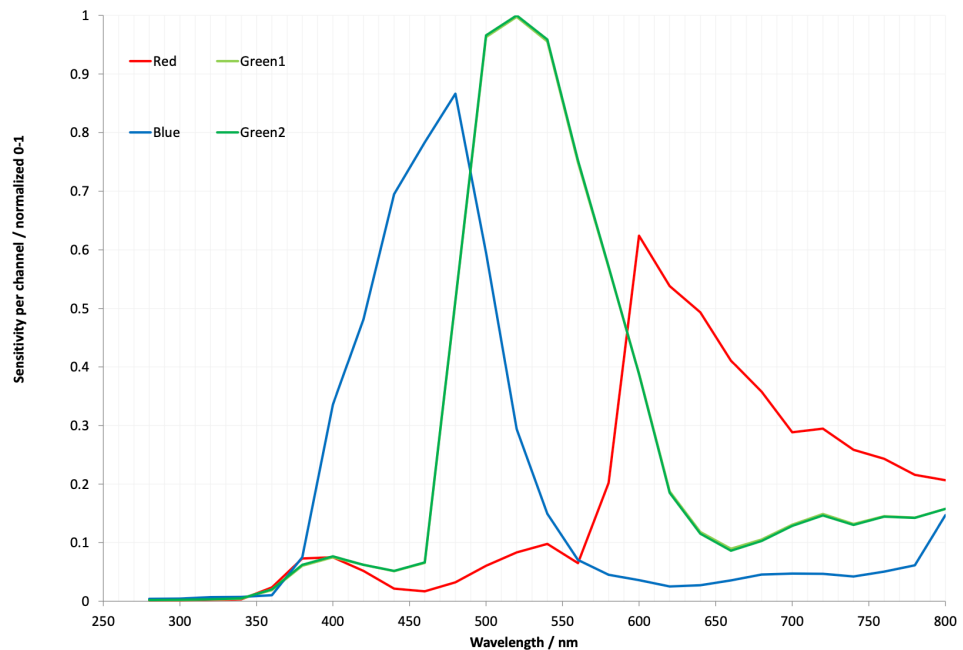
having 16-bit integers representing the voltage value to set. Value 0 was set to be the lowest possible input voltage, and 65534 was set as the highest input voltage.

The MEMS mirrors used in the system had a so-called bias voltage. It meant that to set the mirror position to zero, both control voltages for the mirror needed to be set to a certain value. After both control voltages are at bias voltage, the mirror angle was controlled by increasing the voltage of the first control and decreasing the voltage equally on the second control.

## 5.3 Computer vision system

### 5.3.1 Camera

The camera used in the experiment was a commercial Raspberry Pi High-Quality Camera, with a Sony IMX477 CMOS imaging sensor that has a pixel size of  $1,55\ \mu\text{m}$  (Raspberry Pi 2024). Even though the camera can capture still images in resolution  $4056 \times 3040$  pixels with the pixel size of  $1,55\ \mu\text{m}$ , the used resolution was  $2026 \times 1520$  pixels. With this resolution, the pixel size is  $3,10\ \mu\text{m}$ . The resolution reduction was because of software restrictions, which did not allow the still images to be taken efficiently. Because of this, the lesser resolution was used with the live image stream from the camera. After the magnification, 1 micrometer in the system's output is represented by  $3,333$  micrometers in the camera sensor. This means that one micrometer in the output corresponds to  $3,333\ \mu\text{m} / 3,10\ \frac{\mu\text{m}}{\text{px}} = 1,075\ \text{px}$  in the camera sensor.



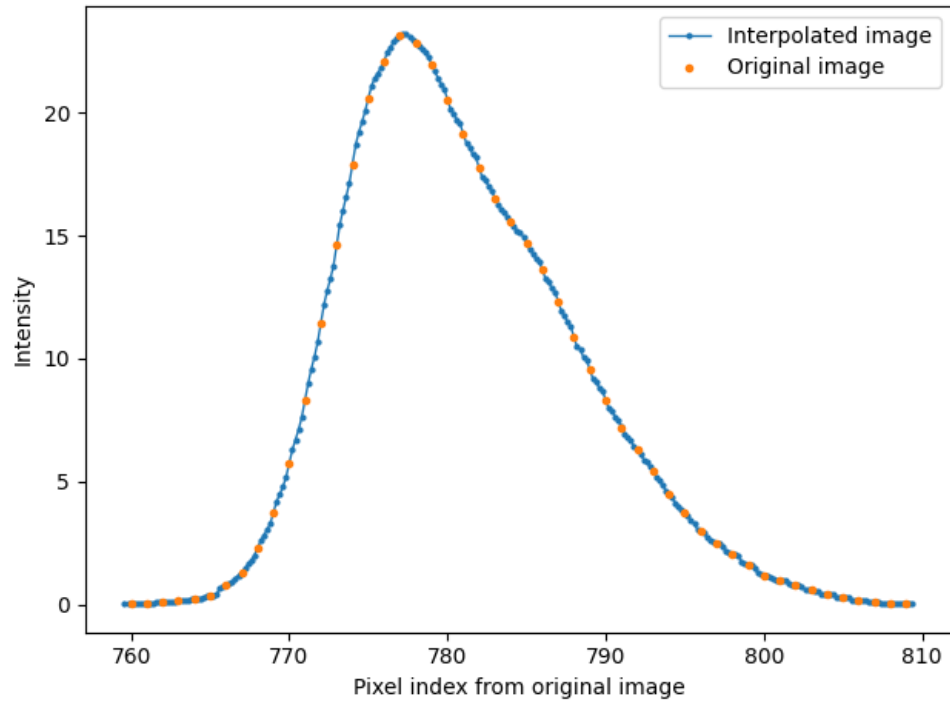
**Figure 5.5.** Spectral response of Sony IMX477 sensor

### 5.3.2 Image processing

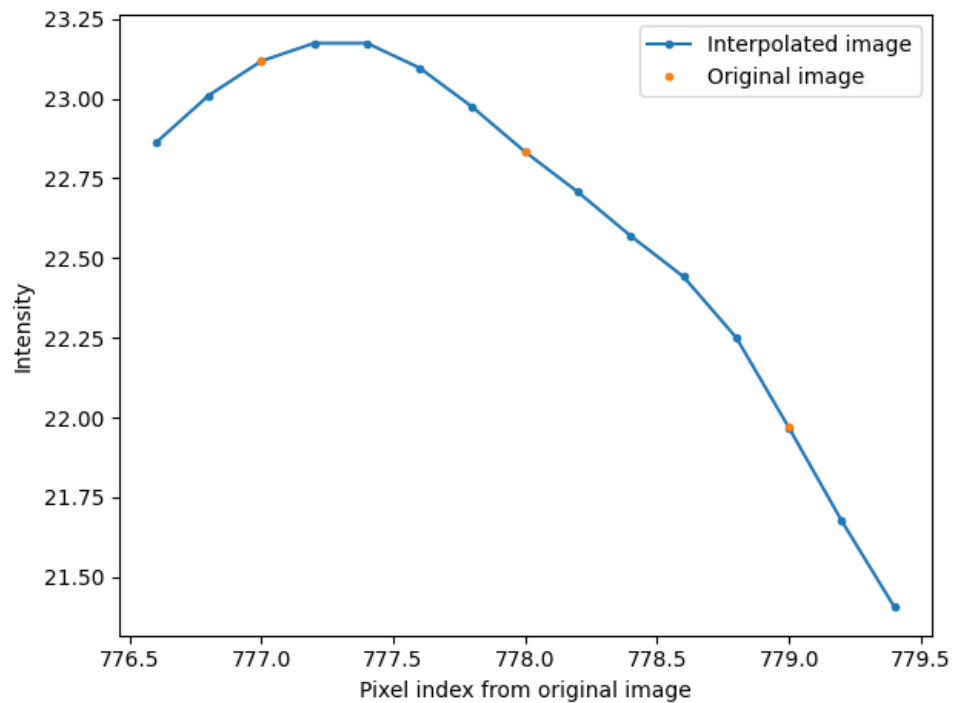
As shown in Figure 1.1, the initial image consists of reflections of the three laser beams. The beams that correspond to the actual output beams of the system are the three beams on the left side of the image. To avoid the effect of them to cause issues on the beam alignment, they are cropped out of the image. As the beams are aligned horizontally to the middle of the output and therefore the camera image, the cropping can be done by simply cropping the image to a certain width so only the real beams fit on the image.

After the cropping, the image is pre-processed to achieve sub-micrometer precision in aligning the beams. The first step in the image pre-processing was to interpolate the image. Based on the calculation in the previous section, the error of one pixel on defining the beam middle is  $1px / (3,333 \mu m / 3, 10 \frac{\mu m}{px}) = 0,93 \mu m$  without interpolation. As this would cause quite a big error margin on the precision when targeting the precision of  $1 \mu m$ , using interpolation is a good way to increase the theoretical maximum accuracy of the alignment system.

Different interpolation ratios and methods were tested with the system, and it was found that the interpolation ratio of 2 and bi-cubic interpolation, defined in Chapter 3.3, was a good trade-off between performance and resulting accuracy. The effect of interpolation can be seen in Figures 5.6 and 5.7. With an interpolation ratio of 2, the computational pixel size is  $0,93 \mu m / 2 = 0,465 \mu m$ . This makes it so that the beam offset of two computational pixels is still lower than the target precision of  $1 \mu m$ .



**Figure 5.6.** Image's vertical profile averaged over horizontal axis in an original and interpolated monochrome image. Interpolation ratio 5.



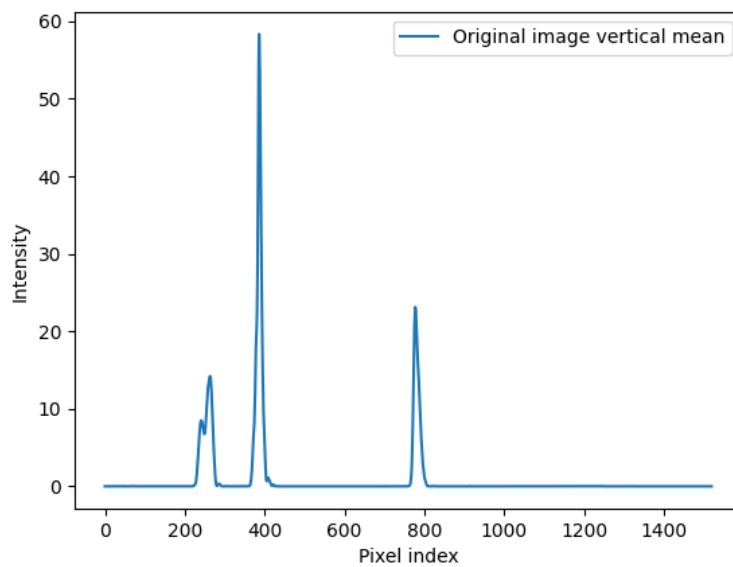
**Figure 5.7.** Image's vertical profile averaged over horizontal axis in an original and interpolated monochrome image. Interpolation ratio 5.

The interpolation process also gives a better estimate of the actual position of the peak

than what could have been acquired from the original image's profile. It can be seen from Figure 5.7 that the peak acquired has been estimated to not be at the peak of the original values but at an interpolated point.

### 5.3.3 Beam detection and recognition

Laser beams are detected by creating a vertical profile of the laser beams. This is done by calculating the mean of each pixel row of the image. This results in a profile shown in Figure 5.8.



**Figure 5.8.** Monochrome image's vertical profile averaged over horizontal axis

From the vertical profile, the intensity peaks can be easily detected from the otherwise black image background, excluding the small noise. As the system's laser beams had Gaussian profiles in the vertical direction, there is a possibility to increase the precision of measuring the beam middle position even further by fitting a Gaussian curve on the beam profile. However even though the Gaussian fitting was tested, it was not used in the experiment as the beam noises caused the Gaussian fits to have some offset compared to the vertical profiles acquired using the mean values of horizontal axis values. This could be fixed by fitting the Gaussian function based on only some part of the intensity peak of the laser beam, but implementing this is left for future work.

The initial method tested for laser beam identification is identifying them by their color. As the wavelengths of the beams are known, the scale between color channels can be estimated from the spectral response of the imaging sensor. The spectral response of the used camera can be seen in Figure 5.5. For example, the scale of RGB channels at a

wavelength of 500 nanometers can be read from the figure to be around  $[0.05, 0.95, 0.60]$ .

When the RGB scale of a color is known, the RGB values of pixels in the image in the middle of the laser beam can be captured. When taking the RGB values from each of the beam peaks, the nearest neighbor for matching the beams to the closest color can be found easily. This method was found to be a working solution with the used lasers.

## 5.4 Error calculation

The error is calculated by comparing the detected beam positions to predefined positions where the beams should be. The difference between the actual and target positions is calculated in pixels, and the pixel difference is then turned into micrometers as described in Section 5.3.2. The target positions of the laser beams are defined based on the needs of the flow cytometry application the device is part of, and for this research a beam pattern where the beam centers are  $400\ \mu\text{m}$  apart from each other was used.

## 5.5 Control software

The control software utilizes Python libraries for different tasks. NumPy (Harris et al. 2020) is a Python library with various matrix operations and therefore was used for matrix calculations and operations like image cropping and calculating mean. SciPy (Virtanen et al. 2020) is a Python library that provides algorithms for optimizations and statistics among other things, and here it was used as it implements the optimization algorithms specified in Chapter 4. The third main library used is OpenCV2 (Bradski 2000). It is a computer vision library consisting of thousands of algorithms for image and video processing. Its main usage in this research was image reading and the interpolation of images.

The software created for this experiment had a constraint. It needed to be integrated into existing laser driving software written in Python. This caused issues like the camera resolution issue mentioned in Section 5.3.

There was a ready-made implementation in the SciPy library for the Nelder-Mead method and Powell's algorithm. Using these algorithms was made simple, as the algorithms needed a function with the control values provided by the algorithm as inputs, and some scalar error value as output. In addition to this function, the algorithm needed an initial guess for the optimization, optionally minimum and maximum values for the control variables, and convergence parameters.

As mentioned in Chapter 5.2.3, the actuated mirrors needed a bias voltage that needs to be set to both control values to set the mirror in a zero position. Because of this it was easy to implement normalized controlling, where a control value of zero means the zero position, -1 means that the mirror is set to maximum negative angle, and 1 means that

the mirror is set to maximum positive angle. These values are then transformed to control voltages on the lower level of the control software. The algorithm uses these normalized control values, so the bounds of the values were also easy to define.

When the optimization algorithm comes up with new values to try, the application works very sequentially. First, the algorithm generates control values, which are sent to the lower level of the control system, which calculates the voltages to set for the mirror control. The voltages are then communicated to the driver hardware which then applies the voltages. After the mirrors have steered the beam, an image from the camera video stream is taken. As video streaming is relatively compute-intensive, a little artificial delay is added before capturing the image to ensure that the captured frame is taken after the movement of the mirrors.

After the image has been captured, the beam detection and recognition are done according to Section 5.3.3. In the software, the beam recognition phase returns information on recognized beams with their locations on the image. However, there is a possibility that a beam has not been detected, which affects the error calculation by increasing the error. This tells the algorithm that losing a beam from the image is not a wanted behavior.

The coordinate system used in the image calculations defines the origin in the middle of the image, with the Y-axis being the vertical axis, and the X-axis being the horizontal axis. Positive ends of the axis were defined to be right and up. The error is calculated using the image coordinate values received from beam recognition to calculate the offset between predefined target positions. As the target positions have been defined by a micrometer offset from the center of the image/output, they need to be transformed to the pixel coordinates with the origin in the middle of the image. After this, the error between the beam and the target positions can be easily calculated using subtraction. It was wanted for the error to be offset as micrometers in the output, thus the pixel values from the subtraction needed to be converted to micrometer values. The error as micrometers

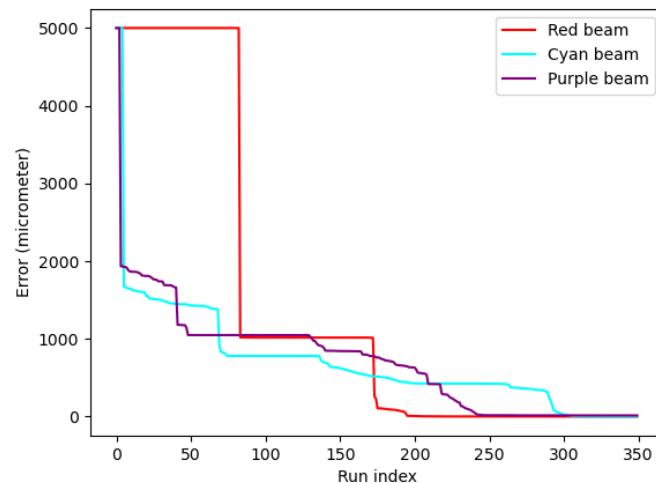
$$E_{\mu\text{m}} = \frac{E_{\text{px}}c}{Im}, \quad (5.1)$$

where  $I$  is the interpolation scale,  $c$  the camera sensor's pixel size, and  $m$  the optical magnification of the system.

## 6. RESULTS

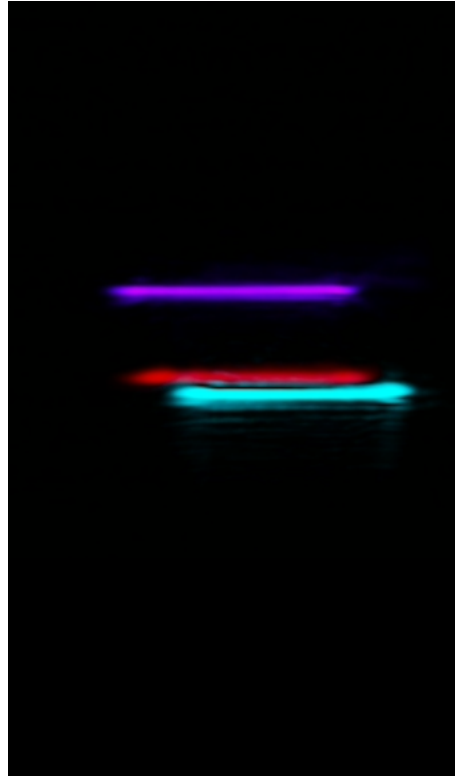
### 6.1 Initial attempts

For the initial attempts, there was an attempt to control one mirror at a time in the optimization order red, cyan, and purple using the Nelder-Mead method. A data collection study was done with random initial guesses for the control values. Different convergence parameters were used for the study to discover how the system works when running the optimization in this order. The results of this test can be seen in Figure 6.1.



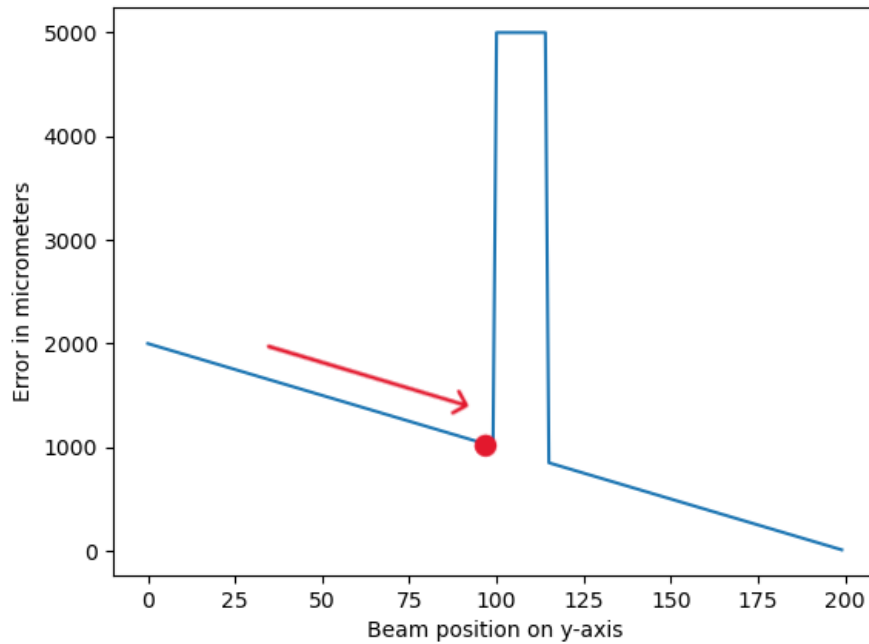
**Figure 6.1.** Optimization results of 350 random initial guesses with several convergence parameters. Errors are sorted in descending order.

The error function is just the distance between the positions of beams and the target positions in micrometers. For positions when a beam couldn't be found, like when two beams are on top of each other or when a beam is steered out of the camera sensor, an error of 5000  $\mu\text{m}$  was set. It can be seen from the results that this was most often a problem for the alignment of the red beam. This was because as described in Section 5.2.3, the red beam cannot be moved individually, so in cases where the random initial position of two beams are on top of each other, the red beam cannot escape this position by changing its control, and will have the error value 5000  $\mu\text{m}$ .



**Figure 6.2.** Cyan beam unable to cross past the red beam seen on the system's camera.

Overall, it was witnessed that there were multiple occasions where the cyan or purple beams could not pass on the other side of another beam. This effect can be seen in Figure 6.2, where moving the beam on top of the red beam would cause a peak in the objective function value. This is understandable, as the Nelder-Mead method thinks this direction is not good to proceed when the value of the objective function is suddenly very high. This is one disadvantage of the Nelder-Mead method and is illustrated in Figure 6.3, where the red spot marks the spot where the algorithm will converge, and the red arrow represents the path that the optimization has already gone.



**Figure 6.3.** Illustration of the error caused by another beam. The red marker marks the spot where the algorithm converges, and the arrow shows the path that optimization has traversed. If the beam that is being optimized crosses on top of another beam, the beam is not recognized anymore which results in an error of 5000  $\mu\text{m}$ .

The Nelder-Mead method was found to work for the beam alignment when aligning mirrors one beam at a time, but the alignment process was slow. Alignment of all mirrors at the same time was attempted to improve the speed of alignment. The results of that were not good with the initial error function and color recognition method as the beams had problems passing each other more than what they had when moving one by one. The difficulties increased most likely because when moving two beams to be at the same spot the algorithm moved both of the beams to the location that they were coming from. There was also a problem with just the overall performance, as the beams couldn't find the target positions. This was suspected to happen because the error is calculated the same way for each beam that is being aligned. This way the priority for the beam aligning is equal, but this has a problem as the alignment of the red beam also affects the alignment of other beams.

### 6.1.1 Algorithm improvements

As mentioned, there were two major problems with the initial implementation. The first one is that it is hard for the beams to pass over each other, and the second problem is that when aligning all beams simultaneously, the overall performance is bad as the alignment of the red beam affects the other beams too.

We came up with two possible solutions for the problem with beams getting on the other sides of each other. One is restarting the failed beam's alignment until it achieves the target position. When using random initial positions, it is probable that at some point the initial position is such that the alignment process converges successfully. This method was tested for reliability, and it worked to some extent. However, as that method depends on probabilities and drastically increases the alignment's duration, another method could be preferred.

The error calculation function was improved by instead of having the linear error that depends on the offset of a beam from the target, the offset values were raised on different powers. Different powers were tested, and it was found that having the error for the cyan and purple beam to be squared, and the error for the red beam in the power of three had good results. These modifications to the error calculations made the alignment of the red beam to be prioritized, and the faster decrease of the error function value on all the beams made the alignment process overall quicker.

Changing the error function also had an effect in that the beams did not get stuck on each other as much as before. The beam detection function could be improved further by recognizing if two or more beams are on top of each other. For this, improvements in the color recognition algorithm would be needed. A method was devised where the RGB values of the beams individually, and on top of each other, were captured. Then when the beam was recognized, it was identified by looking at the nearest neighbor from the colors of individual beams and stacked beams to identify stacked beams and individual beams. These RGB values were captured simply by manually aligning the beams on top of each other and capturing the image with the system's camera. However, this was not a perfect method as many possible color hues can emerge when two beams are on top of each other in different amounts.

## **6.2 Choosing best control method**

After the algorithm for error calculation and beam detection were improved, there was a comparison between methods to see the advantages and disadvantages of each. In the following figures, success is defined as having a total alignment error of less than one micrometer, and failure is defined so that the error value is over 1000. With the revised error calculation the error value goes easily to over 1000 when any problem occurs in the alignment process. It can be seen from Table 6.1 that Powell's algorithm did not fail the convergence at all, and the Nelder-Mead method did, which makes Powell's algorithm more reliable on this problem.

Method	Mean error value ( $\mu m$ )	Failure rate	Success rate
Nelder-Mead	0,66	30%	70%
Powell	1,77	0%	100%
ANMS	0,70	20%	70%

**Table 6.1.** Mean error values from runs that did not fail and failure rates for each method. Failure is defined by a run having an error value over 1000 micrometers. Success is defined as a run with an error of less than a micrometer.

Table 6.1 shows that ANMS ( Adaptive Nelder-Mead Simplex) method (Gao and Han 2012) did not have any good properties on this problem, as it was slower to converge and didn't converge successfully more often than original Nelder-Mead method. The mean error on successful runs was also a little higher, with a bigger 95% confidence interval. Because of these findings, the ANMS method was not considered viable for the alignment process further in the research. It was stated in the paper by Gao and Han (2012) that the ANMS method suits better high dimensional problems. This case had an optimization of only three dimensions, so it is not surprising that the ANMS method was not optimal for this problem.

As seen in the results in Table 6.1, Powell's algorithm is more reliable as it did not fail any time. However, it does have a higher error value on convergence on successful runs compared to the Nelder-Mead method. Nelder-Mead's successful convergence also has an error that is lower than the target error value of 1  $\mu m$ , where Powell's algorithm's mean error is almost 2  $\mu m$  which is twice as big as the target precision of the alignment. The importance of initializing the Nelder-Mead algorithm with proper initial values has been studied in (Wessing 2019), where it was found that constructing the initial simplex by hand and not from the starting point is advisable for improved performance of the method.

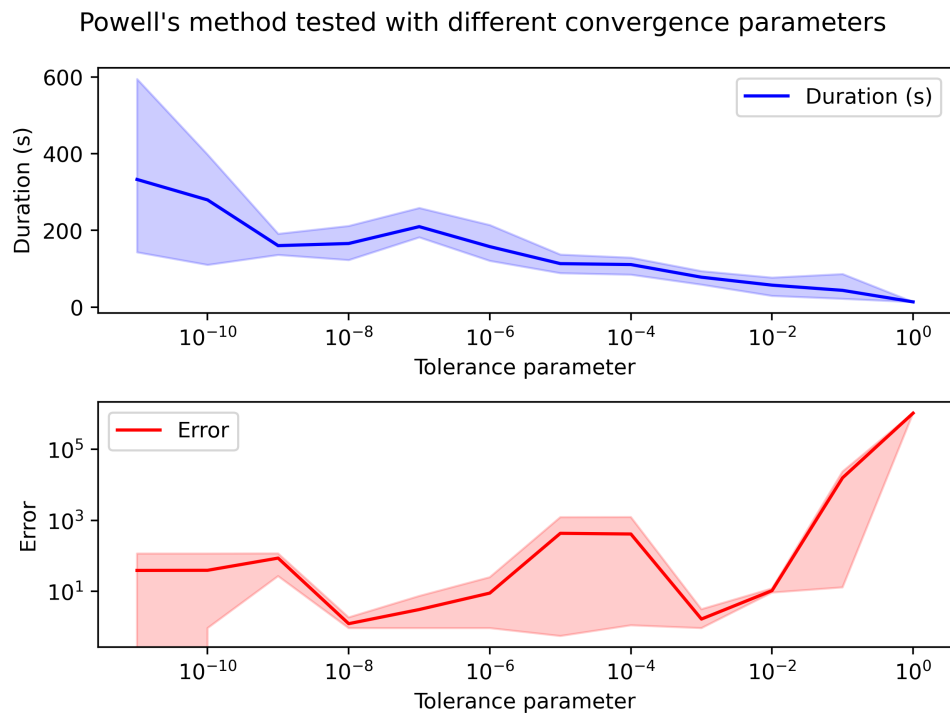
It was found that Powell's method was more reliable in getting alignment close to the target precision, and the Nelder-Mead method achieved lower error scores more consistently. Therefore it was decided to use a combination of both of these methods. Powell's method will do the initial alignment to align the beams close to the optimal place, so the error is around 10  $\mu m$ . After this, the Nelder-Mead method is used, which should reliably achieve successful alignment from a position that is close to a successful alignment.

### 6.3 Finding optimal parameters

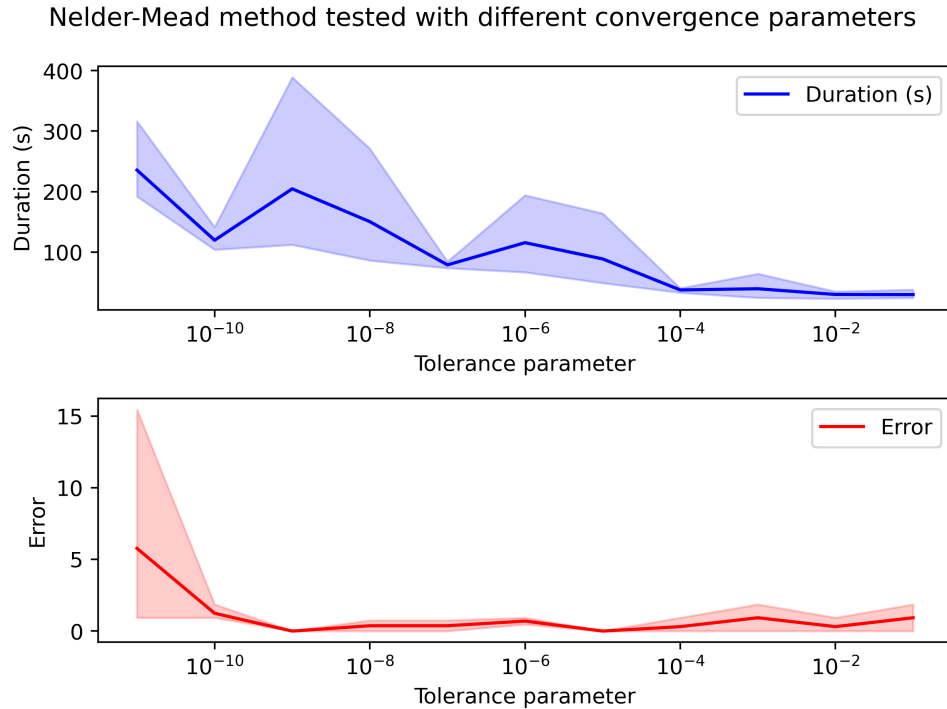
The convergence parameters of the Nelder-Mead method and Powell's method change the alignment precision that the methods converge. This can be used to optimize Powell's method to converge quicker at a good enough alignment, after which the Nelder-Mead method can have convergence parameters that reliably achieve successful alignment.

Finding the optimal parameters was done simply by testing multiple parameters, and after that by looking at the results and choosing parameters that offer a good balance for precision and alignment duration.

The test was implemented with initial values close to the target alignment, with random noise added each time. This was to simulate the long-term drift of the beam alignment. The results of the tests can be seen in Figures 6.4 and 6.5. In Figures, the solid line shows the mean value achieved over a couple of tests with the same parameters. The colored area shows a 95% confidence interval for the value.



**Figure 6.4.** Powell's algorithm optimization duration and achieved error on different tolerances



**Figure 6.5.** the Nelder-Mead algorithm optimization duration and achieved error on different tolerances

Based on the results of these tests, Powell's algorithm tolerance values  $1e-3$  and  $1e-2$  provided quick convergence close to the target which is the target of Powell's algorithm. The value  $1e-3$  was chosen to be best as the duration difference between them is not big, and it is overall a bit safer choice as it can be seen that bigger tolerance values seem to cause an exponential increase on the achieved error.

For the Nelder-Mead algorithm, it can be seen that with tolerance values less than  $1e-7$  the durations of the optimizations start to increase drastically, and with tolerance values higher than  $1e-4$  the resulting error value from the optimization starts to increase over the target of  $1\ \mu\text{m}$ . Based on these tests, the tolerance value of  $1e-5$  was deemed the best tolerance value.

## 6.4 Final alignment precision

The alignment precision target of  $1\ \mu\text{m}$  was achieved. The mean estimate of precision achieved from automatic alignments is  $56,9\ \mu\text{m}$ , but this includes the one high value seen on the table in Appendix A. If that run is excluded, the mean of the beam offsets is  $0,27\ \mu\text{m}$ . The results in Appendix A also show that out of the 27 runs, two runs had an error of over 1 micrometer. From these two, one run is very unsuccessful with the error value of  $1530\ \mu\text{m}$ , and the other is just barely over the 1-micrometer target with an error value of  $1,4\ \mu\text{m}$ .

This is only an estimate of the actual precision, as the measurement of a more accurate precision would need an external measurement system, which could take into account problems in the internal system, like getting absolute truth on the beam positions instead of trusting the camera that is used for the alignment.

Another source of unreliability for the measurement is that interpolation is used for assessing the beam positions. As it is just a way of estimating the missing pixel intensity value, there would be a need for a system where the optical magnification would be bigger, or the camera sensor's pixel size would be a lot smaller. This would allow using the raw images from the sensor, and remove the need for utilizing interpolation or other methods for calculating the beam positions. As such a system was unavailable for the evaluation, these estimated values for the results need to be used.

## **6.5 Alignment duration**

The alignment duration didn't have any specific target, but it was thought that a duration of a couple of minutes would be good. The achieved mean duration for Powell's method's convergence was 120,4 seconds, and the Nelder-Mead method's duration was 135,1 seconds. Therefore the mean total duration of the alignment is 255,5 seconds, a little over 4 minutes. The results can be seen in Appendix B.

The duration is a bit longer than what would be optimal. One reason for this long alignment duration is software compatibility issues with the camera driver. This caused a need for adding an artificial delay in the image acquisition. Another camera-based increase in the duration is that the interpolation is one of the longer calculations that are part of the optimizer algorithm's control loop. A camera with a smaller pixel size could eliminate the need for interpolation, decreasing the duration of an iteration of the control loop.

## 7. CONCLUSIONS

It was found that the numerical optimization methods are suitable for aligning laser beams accurately. The combination of Powell's method with the Nelder-Mead method gives a reliable method for the alignment problem. The initial optimization with Powell's method provides a close initial estimate of the alignment after which the Nelder-Mead method's precise alignment finishes the offset minimization.

The alignment precision target for each beam was  $1\ \mu\text{m}$ , and the target for the alignment duration was that it would be a couple of minutes at maximum. The precision target was achieved with the combination of Powell's method and the Nelder-Mead method very reliably. The nature of Powell's method is such that it converges to the target's proximity, where the Nelder-Mead can continue and finish the alignment from Powell's algorithm's result. The numerical results were discussed more thoroughly in chapter 6.4.

The alignment's duration with the combination of algorithms depends on the convergence parameters. They were optimized for a good balance of duration and precision. This was discussed more thoroughly in chapter 6.5. The achieved duration for the alignment was over 4 minutes, which is quite a long duration.

Overall the alignment's duration is reduced to a small part of what it was with manual alignment. The software was not optimized for the duration, and as mentioned in chapter 6.5, hardware changes could immediately reduce the duration. The duration could also be reduced by not running Powell's method every time, but only when the initial offset is over some threshold. This way small drifting of the beams would be corrected with only the Nelder-Mead method, which would cut the duration approximately in half. However, as there is no data on what the long-term drift looks like with this setup, experimenting with this must be left for future work.

Overall the implementation is considered successful and the achieved reliability and precision are proficient for the intended use of the device. The target of the development of the whole device was to reduce the maintenance load and downtime caused by the long-term drift of the alignment. This implementation is considered highly successful in that regard.

## 7.1 Future work

The study found multiple interesting things that could be investigated more. First, it would be interesting to see how a Gaussian fit on the Gaussian beam profile would increase the precision of finding the middle of the beam. If it would provide good results, it could even remove the need for the bicubic interpolation used in our approach.

As explained in Chapter 5.2.1, there is a DOE that would work optimally if there were two actuated mirrors for each laser beam. This would provide a more challenging environment for the aligning, and it would be interesting to test the methodology presented in this thesis on that more challenging environment. This kind of environment would probably need to have an error metric for the beam quality, which decreases when the laser beam does not pass through the DOE in the middle. A system with a more complex control algorithm might be needed for that type of system.

A more complex control algorithm for the system could be implemented using reinforcement learning. The basic principle of reinforcement learning is similar to the numerical methods, as the reinforcement learning agent uses trial and error to find the optimal solution. Q-learning and even a deep learning version called Deep Q-learning could be tested on this problem to see if reinforcement learning would provide a quickly converging and precise solution to this problem.

## REFERENCES

- Bayer, Bryce E. (July 1976). "Color imaging array". US3971065 A. URL: <https://worldwide.espacenet.com/patent/search/family/019057066/publication/US3971065A?q=pn%3DUS3971065>.
- Blondin, M.J. et al. (2018). "New optimal controller tuning method for an AVR system using a simplified Ant Colony Optimization with a new constrained Nelder–Mead algorithm". eng. In: *Applied soft computing* 62, pp. 216–229. ISSN: 1568-4946.
- Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.
- Breckling, Sean et al. (2022). "An automated approach to the alignment of compound refractive lenses". eng. In: *Journal of synchrotron radiation* 29.4, pp. 947–956. ISSN: 1600-5775.
- Bromberg, Joan Lisa (1988). "The birth of the laser". eng. In: *Physics today* 41.10, pp. 26–33. ISSN: 0031-9228.
- Chang, Kuo-Hao (2012). "Stochastic Nelder–Mead simplex method – A new globally convergent direct search method for simulation optimization". eng. In: *European journal of operational research* 220.3, pp. 684–694. ISSN: 0377-2217.
- Cheng, Jade Yu and Thomas Mailund (2015). *An iteration of the Nelder-Mead method over two-dimensional space*. Online; accessed February 12, 2024. This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>. URL: <https://commons.wikimedia.org/w/index.php?curid=122326033>.
- Einstein, Albert (1917). "Zur Quantentheorie der Strahlung". In: *Physikalische Zeitschrift* 18. Translated as "On the Quantum Theory of Radiation", pp. 121–128.
- Ekinci, Serdar et al. (2023). "An effective control design approach based on novel enhanced aquila optimizer for automatic voltage regulator". eng. In: *The Artificial intelligence review* 56.2, pp. 1731–1762. ISSN: 0269-2821.
- Gao, Fuchang and Lixing Han (Jan. 2012). "Implementing the Nelder-Mead simplex algorithm with adaptive parameters". English. In: *Computational Optimization and Applications* 51.1. Copyright - Springer Science+Business Media, LLC 2012; Document feature - ; Last updated - 2023-11-22, pp. 259–277.
- Gonzalez, Rafael C. and Richard E. Woods (2007). *Digital image processing*. eng. 3rd ed. Upper Saddle River (N.J.): Prentice Hall. ISBN: 0-13-168728-X.
- Google Scholar (2024). *Google scholar: A simplex method for function minimization*. URL: <https://scholar.google.com> (visited on 03/11/2024).

- Harris, Charles R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- Houssein, Essam H. et al. (2020). "Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems". eng. In: *Engineering applications of artificial intelligence* 94, pp. 103731–. ISSN: 0952-1976.
- Izci, Davut (2021). "Design and application of an optimally tuned PID controller for DC motor speed regulation via a novel hybrid Lévy flight distribution and Nelder–Mead algorithm". eng. In: *Transactions of the Institute of Measurement and Control* 43.14, pp. 3195–3211. ISSN: 0142-3312.
- Izci, Davut, Baran Hekimoğlu, and Serdar Ekinçi (2022). "A new artificial ecosystem-based optimization integrated with Nelder-Mead method for PID controller design of buck converter". English. In: *Alexandria engineering journal* 61.3, pp. 2030–2044. ISSN: 1110-0168.
- Lagarias, Jeffrey C. et al. (1998). "Convergence properties of the Nelder-Mead simplex method in low dimensions". eng. In: *SIAM journal on optimization* 9.1, pp. 112–147. ISSN: 1052-6234.
- Larson, Jeffrey, Matt Menickelly, and Stefan M. Wild (2019). "Derivative-free optimization methods". eng. In: *Acta numerica* 28.2010, pp. 287–404. ISSN: 0962-4929.
- Lin, Psang Dain (2014). "Modeling an Optical System". In: *New Computation Methods for Geometrical Optics*. Singapore: Springer Singapore, pp. 49–86. ISBN: 978-981-4451-79-6. DOI: 10.1007/978-981-4451-79-6\_3. URL: [https://doi.org/10.1007/978-981-4451-79-6\\_3](https://doi.org/10.1007/978-981-4451-79-6_3).
- Longere, P. et al. (2002). "Perceptual assessment of demosaicing algorithm performance". eng. In: *Proceedings of the IEEE* 90.1, pp. 123–132. ISSN: 0018-9219.
- Macey, Marion G. (2007). *Flow cytometry principles and applications*. eng. 1st ed. Totowa, N.J: Humana. ISBN: 1-280-97216-5.
- Maître, H. (Henri) (2017). *From photon to pixel : the digital camera handbook*. eng. Revised and updated second edition. Digital signal and image processing series. Hoboken, New Jersey: Wiley. ISBN: 1-119-40244-1.
- Manthei, Alina C. et al. (2021). "Refining pulsar radio emission due to streaming instabilities: Linear theory and PIC simulations in a wide parameter range". eng. In: *Astronomy and astrophysics (Berlin)* 649. ISSN: 0004-6361.
- Mota-Babiloni, Adrián et al. (2018). "Optimisation of high-temperature heat pump cascades with internal heat exchangers using refrigerants with low global warming potential". eng. In: *Energy (Oxford)* 165, pp. 1248–1258. ISSN: 0360-5442.
- Nash, J.C. (1990). *Compact numerical methods for computers: linear algebra and function minimisation*. CRC Press.
- Nelder, John and Roger Mead (1965). "A simplex method for function minimization". eng. In: *Computer journal* 7.4, pp. 308–313. ISSN: 0010-4620.

- Ozturk, Savas, Levent Aydin, and Erdal Celik (2018). "A comprehensive study on slicing processes optimization of silicon ingot for photovoltaic applications". eng. In: *Solar energy* 161, pp. 109–124. ISSN: 0038-092X.
- Powell, M. J. D. (1964). "An efficient method for finding the minimum of a function of several variables without calculating derivatives". In: *Computer Journal* 7.2, pp. 155–162. DOI: 10.1093/comjnl/7.2.155. URL: <https://doi.org/10.1093/comjnl/7.2.155>.
- Raspberry Pi (2024). *Raspberry Pi High Quality Camera*. Online; Accessed: 2024-02-16. URL: <https://www.raspberrypi.org/products/raspberry-pi-high-quality-camera/>.
- Silfvast, William Thomas (2004). *Laser fundamentals*. eng. Second edition. Cambridge: Cambridge University Press. ISBN: 1-316-08607-0.
- Smith, C. S. (1962). *The automatic computation of maximum likelihood estimates*. Scientific Department, National Coal Board.
- SpaceX (2024). *Starlink technology*. <https://www.starlink.com>. Accessed: 2024-04-13.
- Sun, Fengyang et al. (2017). "Improving Multi-layer Particle Swarm Optimization Using Powell Method". eng. In: *Advances in Swarm Intelligence*. Vol. 10385. Cham: Springer International Publishing, pp. 166–173. ISBN: 9783319618234.
- Sutton, Richard S., Andrew G. Barto, and Francis Bach (2018). *Reinforcement learning : an introduction*. eng. Second edition. Adaptive computation and machine learning. Cambridge, MA ; The MIT Press. ISBN: 9780262039246.
- Szeliski, Richard (2010). *Computer Vision: Algorithms and Applications*. eng. 1st ed. Texts in Computer Science. London: Springer Nature. ISBN: 9781848829350.
- Tan, Daniel Stanley, Wei-Yang Chen, and Kai-Lung Hua (2018). "DeepDemosaieking: Adaptive Image Demosaicking via Multiple Deep Fully Convolutional Networks". eng. In: *IEEE transactions on image processing* 27.5, pp. 2408–2419. ISSN: 1057-7149.
- Virtanen, Pauli et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17, pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- Walker, Bruce H. (2008). *Optical engineering fundamentals*. eng. 2nd ed. Tutorial texts in optical engineering ; v. TT82. Bellingham, Wash: SPIE Press. ISBN: 1-62870-863-8.
- Wessing, Simon (2019). "Proper initialization is crucial for the Nelder–Mead simplex search". eng. In: *Optimization letters* 13.4, pp. 847–856. ISSN: 1862-4472.
- Xianguo, Li et al. (2018). "Laser-based on-line machine vision detection for longitudinal rip of conveyor belt". eng. In: *Optik (Stuttgart)* 168, pp. 360–369. ISSN: 0030-4026.
- Xu, Si-Jia et al. (Sept. 2021). "Machine vision-based high-precision and robust focus detection for femtosecond laser machining". In: *Opt. Express* 29.19, pp. 30952–30960. DOI: 10.1364/OE.438537. URL: <https://opg.optica.org/oe/abstract.cfm?URI=oe-29-19-30952>.
- Yin, Zhen-Yu et al. (2018). "Optimization techniques for identifying soil parameters in geotechnical engineering: Comparative study and enhancement". eng. In: *International journal for numerical and analytical methods in geomechanics* 42.1, pp. 70–94. ISSN: 0363-9061.

Zhou, Yi et al. (2022). "Numerical study for influence of ozone on the combustion of biodiesel surrogates in a homogeneous charge compression ignition engine". eng. In: *Fuel processing technology* 225, pp. 107039–. ISSN: 0378-3820.

## APPENDIX A: FINAL TEST ERRORS

Test index	Powell (um)	Nelder-Mead (um)
0	1.4	0.93
1	1.86	0.46
2	3.26	0.0
3	0.93	0.46
4	0.46	0.0
5	2.79	0.0
6	0.0	0.0
7	1.4	0.46
8	0.46	0.0
9	21.39	0.0
10	1.4	0.0
11	416.64	1530.32
12	0.93	0.46
13	0.93	0.0
14	1.4	0.0
15	0.93	0.0
16	0.93	0.93
17	21.86	0.46
18	0.93	0.0
19	7.9	0.46
20	36.27	0.0
21	0.46	0.0
22	2.79	1.4
23	0.93	0.46
24	0.46	0.0
25	1.4	0.0
26	24.64	0.46

## APPENDIX B: FINAL TEST DURATIONS

Test index	Powell (s)	Nelder-Mead (s)	Total (s)
0	185.9	394.8	580.8
1	142.7	80.8	223.5
2	90.9	82.7	173.6
3	89.9	78.8	168.7
4	359.2	104.2	463.5
5	120.3	116.2	236.5
6	88.2	140.2	228.4
7	87.2	82.0	169.1
8	145.1	104.1	249.2
9	57.3	103.7	161.1
10	93.5	96.6	190.1
11	64.4	397.4	461.9
12	104.4	80.7	185.1
13	130.0	85.2	215.3
14	89.8	85.8	175.5
15	113.2	78.5	191.7
16	110.6	395.1	505.7
17	83.8	82.1	165.9
18	101.0	83.8	184.9
19	308.3	77.5	385.8
20	60.0	316.5	376.4
21	117.7	94.2	211.9
22	93.9	121.3	215.1
23	108.9	95.6	204.5
24	133.0	111.6	244.6
25	107.3	78.2	185.5
26	64.0	80.1	144.0