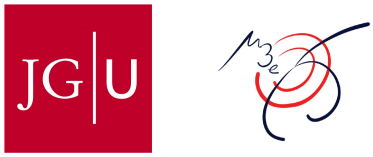
Johannes Gutenberg University

Master's Thesis

A Camera Alignment System for the Mu3e Experiment



Author:
Goran Stanić

Supervisor:
Prof. Dr. Niklaus Berger

Examiner:
Jun.-Prof. Dr. Florian Hug

A thesis in fulfillment of the requirements
for the degree of Master of Science

in the Group

Mu3e

at the

Department of Physics, Mathematics and Computer Science

Institute for Nuclear Physics

Johannes Gutenberg University

September 2021

Declaration of Authorship

I, Goran Stanić, declare that this Master's Thesis titled "A Camera Alignment System for the Mu3e Experiment" is my own work and was written without help or any aids other than those specified therein. I also affirm that all text passages, drawings, sketches, illustrations, and similar that were taken from other works are quoted either literally or correspondingly have been identified as such.

I declare and confirm that this Master's Thesis has not been submitted in this or any other form to any institution in the context of examination requirements.

Signed: 

Date: 13. 9. 2021

To my late grandpa Ljubo

JOHANNES GUTENBERG - UNIVERSITÄT

Zusammenfassung

Fachbereich Physik, Mathematik und Informatik
Institut für Kernphyik

Master of Science

Ein Kamerasystem zur Ausrichtung für das Mu3e-Experiment

von Goran Stanić

Das Mu3e-Experiment wird am PSI (Paul Scherrer Institut) im schweizerischen Villigen durchgeführt und hat es zum Ziel den Lepton-Flavor-verletzenden Zerfall $\mu \to e^+e^+e^-$ bei Verzweigungsfraktionen von über 10^{-16} zu finden oder auszuschließen. Der Mu3e-Detektor ist hauptsächlich ein Spurdetektor, der aus dünnen monolithischen aktiven Pixelsensoren mit Hochspannung (HV-MAPS) besteht. Diese werden durch szintillierende Fasern und Kacheln ergänzt, die präzise Zeitmessungen ermöglichen. Die einzelnen Detektorteile sind in einem Detektorkäfig integriert, der in einen 1 T-Solenoidmagneten eingesetzt ist.

Eine der größten Herausforderungen des Experiments liegt in der präzisen Ausrichtung der Detektorelemente. Eine Fehlausrichtung (Misalignment) kann durch verschiedene Faktoren verursacht werden. Dazu zählen die Konstruktion der einzelnen Detektorelemente sowie deren Integration bis hin zu den Umgebungsfaktoren während der Betriebszeit des Experiments. Um die bestmögliche Impulsauflösung zu erreichen und die Auswirkungen der Fehlausrichtung zu reduzieren, wird ein spurbasiertes Ausrichtungsprogramm verwendet. Dieses hat jedoch den Nachteil, dass sogenannte schwache Moden, d. h. Verformungen des Detektors, die Spuren gleicher Qualität erzeugen, nicht aufgelöst werden können.

Das Ziel der vorliegenden Arbeit ist es die schwachen Moden zu korrigieren, indem ein hochpräzises kamerabasiertes Ausrichtungssystem entworfen und entwickelt wird, das die Detektorposition von außen überwacht. Das System besteht aus mehreren Infrarotkameras, die sich gegenseitig und den Aufbau des Detektors beobachten. Das Hauptaugenmerk besteht darin, die Messgenauigkeit der Kameras so weit zu erhöhen, dass sie mit der einzelnen Pixel des Tracking-Detektors vergleichbar ist, die bei 80 µm liegt.

JOHANNES GUTENBERG UNIVERSITY

Abstract

Department of Physics, Mathematics and Computer Science
Institute for Nuclear Physics

Master of Science

A Camera Alignment System for the Mu3e Experiment

by Goran STANIĆ

The Mu3e experiment is going to be conducted at PSI (Paul Scherrer Institute) in Villigen, Switzerland, and it aims at finding or excluding the lepton flavor violating decay $\mu \to e^+e^+e^-$ at branching fractions above 10^{-16}. The Mu3e detector is mainly a tracking detector built from thin High-Voltage Monolithic Active Pixel Sensors (HV-MAPS) complemented by scintillating fibers and tiles for precise timing measurement. The individual detector parts are integrated and placed on a detector cage which is inserted in a 1 T solenoid magnet.

One of the main challenges of the experiment lies in the precise alignment of detector elements. Misalignment can be driven by multiple factors ranging from the construction of individual detector elements, their integration, all the way to the environmental factors which are present during the on-time of the experiment. In order to achieve the best possible momentum resolution and reduce the effects of misalignment, a track-based alignment program will be utilized. Track-based alignment cannot, however, resolve so-called weak modes - deformations of the detector that produce tracks of equal quality.

This thesis aims at correcting for the weak modes by designing and developing a high-precision camera-based alignment system that monitors the detector position from the outside. The system will consist of multiple infrared cameras, which will observe each other and the detector set-up. The main goal is to drive the camera measurement precision to be comparable to the individual tracking detector pixels, which is at 80 µm.

List of Figures

1.1 The Standard Model of particle physics. . . . 2
1.2 Allowed and forbidden decay in the Standard Model due to lepton flavor conservation. . . . 3
1.3 Neutrino-less decay in the Standard Model . . . 3
1.4 $\mu \to eee$ decay in a SUSY model. . . . 4
1.5 Conducted and proposed experiments searching for lepton flavor violation in muonic experimental channels . . . 4

2.1 Representation of a $\mu \to eee$ decay. . . . 6
2.2 Dominant background processes in the Mu3e experiment. . . . 7
2.3 Detector schematics: Cross-sectional view . . . 7
2.4 High-Voltage Monolithic Active Pixel Sensor (HV-MAPS) . . . 8
2.5 Main parts of the time-of-flight detector. . . . 8
2.6 CAD rendering of the detector cage and the detector . . . 9

3.1 Schematics of a particle traversing three sensor planes . . . 12
3.2 Weak mode in the case of elliptical deformations . . . 13
3.3 Different possible weak modes . . . 13
3.4 Muon tracks during the elliptical deformation . . . 14

4.1 Geometric optics . . . 15
4.2 Schematics of the basic camera set-up with camera modules: cross-sectional view . . . 16
4.3 Schematics of the basic camera set-up with the coordinate system . . . 17
4.4 Raspberry Pi NoIR camera and a depiction of a camera module . . . 18
4.5 Heating up and cooling down of the camera and the surrounding PCB. 19
4.6 Investigation of the dependence between temperature and object position on the image . . . 20

5.1 Basic simulation scene . . . 23
5.2 Rendered image of the detector and the camera set-up . . . 25
5.3 Rendered image of the detector and the camera set-up from a different angle . . . 26
5.4 Simulation results for a test of the longitudinal resolution . . . 27
5.5 Simulation results for a test of the transverse resolution . . . 28
5.6 Simulation depicting a random scan of the 3D space . . . 29
5.7 Simulation depicting a random scan of the 3D space: x-axis projection 29

6.1 Experimental set-up for testing the transverse resolution of the camera 32
6.2 Experimental set-up for testing the longitudinal resolution of the camera 32

6.3 Depiction of the rotation problem 33
6.4 Sketch of the rotation problem 34
6.5 Experimental set-up for testing the rotation of the camera module . . 35
6.6 Three representative rotational positions of the module 35
6.7 Experimental set-up for testing the complete system: side view 36
6.8 Expanded view of the experimental set-up for testing the complete system 36

7.1 Test results of the transverse resolution of the camera 40
7.2 Test results of the longitudinal resolution of the camera 41
7.3 Results of the 3D error analysis 42
7.4 Results of the 3D error analysis: z-projection 42
7.5 Results of the experiment regarding the camera module rotation: position 43
7.6 Results of the experiment regarding the camera module rotation: proof of principle 44
7.7 Results of the experiment with 3 pairs of LEDs 45
7.8 The experimental set-up with 3 pairs of LEDs 46
7.9 Results of the experiment with 5 pairs of LEDs 48
7.10 Results of the experiment with 5 pairs of LEDs: different position . . 48
7.11 Overlap of the plots of experiments with 5 pairs of LEDs 49

List of Tables

4.1 Hardware specifications of the Raspberry Pi v2 NoIR camera 18

5.1 Results of the simulation for a test of the longitudinal resolution . . . 27
5.2 Results of the simulation for a test of the transverse resolution 27

7.1 Error values for the 3 points on the tracker 47

Contents

1 Introduction and Physical Motivation **1**
- 1.1 The Standard Model . . . 1
- 1.2 Leptons and lepton flavor violation . . . 2

2 The Mu3e Experiment **5**
- 2.1 Signal and background . . . 5
- 2.2 The Mu3e detector . . . 6

3 Detector Alignment **11**
- 3.1 Misalignment . . . 11
- 3.2 Track-based alignment . . . 11
- 3.3 Weak modes . . . 12

4 Camera Alignment System **15**
- 4.1 Basic principles . . . 15
- 4.2 Camera modules . . . 17
- 4.3 Object recognition . . . 21

5 Simulation **23**
- 5.1 Ray-tracing . . . 23
- 5.2 Simulation results . . . 25

6 Experimental Set-up **31**
- 6.1 Camera resolution . . . 31
- 6.2 Camera module rotation . . . 31
- 6.3 The complete system . . . 34

7 Results and Discussion **39**
- 7.1 Camera resolution . . . 39
  - 7.1.1 Transverse resolution . . . 39
  - 7.1.2 Longitudinal resolution . . . 40
  - 7.1.3 Discussion . . . 40
- 7.2 Camera module rotation . . . 43
- 7.3 The complete system . . . 44

8 Conclusion and Outlook **51**

A Error propagation **53**

Chapter 1

Introduction and Physical Motivation

> "*Observation, reason, and experiment make up what we call the scientific method.*"
>
> Richard P. Feynman

1.1 The Standard Model

Over the previous couple of decades, in the search for the fundamental building blocks of matter and interactions between them, physicists have been developing and continuously testing a currently dominant description of nature, the Standard Model (SM). Since 2012, after the discovery of the SM's crown jewel - the Higgs boson - it has been the most successful, self-consistent theory, predicting and explaining countless modes of decay, the nature of interactions between different particles, and the general way in which the world around us unfolds.

As seen in Fig. 1.1, the SM contains two overarching categories of elementary particles: *fermions* - the half-integer spin constituents of matter, and *bosons* - the integer spin force-carriers which are responsible for mediating interactions between fermions, giving rise to the electromagnetic force, the weak force and the strong force. The Higgs boson is a category in itself, as it is a manifestation of the Higgs field, which is responsible for giving mass to all the other fundamental particles. In addition, fermions can be subdivided into two categories, namely into quarks and leptons, each consisting of six fundamental particles, two in each of the three generations of matter. Up quarks and down quarks make up protons and neutrons, which together with electrons account for all the visible matter in the Universe. Lest we forget, each of the fermions has its antimatter counterpart, bringing the number up to 12 in total.

Even though the Standard Model represents a pinnacle of physics research to date, it does not account for some of the greatest scientific mysteries, such as the existence of Dark Matter and Dark Energy and the matter-antimatter asymmetry in the Universe. There is also an obvious problem of merging the SM with General Relativity, which would include gravity into the force-carrier family. This is precisely why looking Beyond the Standard Model (BSM) in search of answers is what

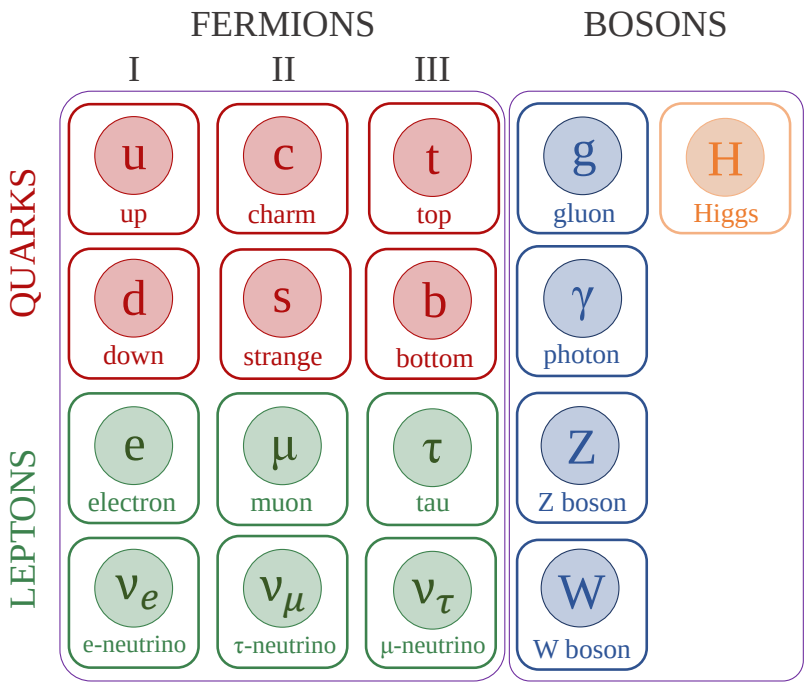


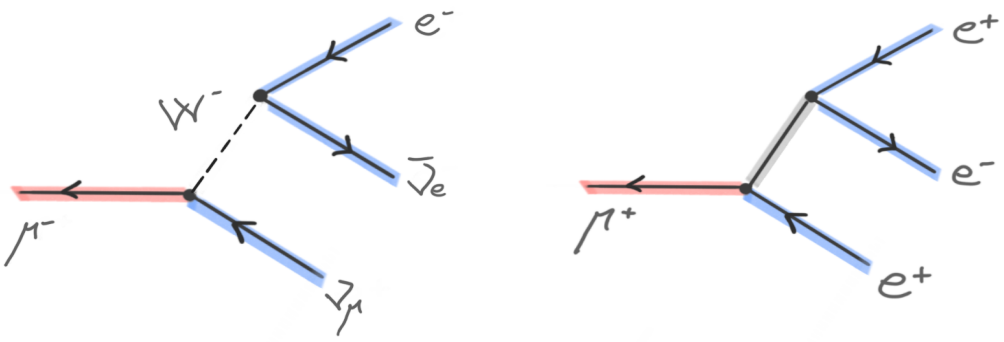
Figure 1.1: The Standard Model of particle physics.

occupies physicists all around the world. It is a fertile ground both for known and unknown unknowns that lie before us.

1.2 Leptons and lepton flavor violation

Leptons are a group of half-integer spin elementary particles that do not interact strongly and can be either charged (electron-like) or neutral (neutrino-like). They come in three generations, the *electronic*, the *muonic*, and the *tauonic*, and each generation forms a weak isospin doublet which allows us to assign them different *flavors* and *flavor numbers*. Particles from the electronic generation have an electronic number $L_e = 1$, from the muonic generation the muonic number $L_\mu = 1$, and from the tauonic generation the tauonic number $L_\tau = 1$. Their antiparticle counterparts have the negative values of the flavor numbers. In the Standard Model, the lepton flavor number is a conserved additive quantum number at the tree-level. This means that the sum of the flavor numbers of each generation has to be the same before and after the reaction. In Fig. 1.2, we can see two decays with their corresponding Feynman diagrams, the one on the left being allowed in the SM and the one on the right forbidden.

This, however, is not the complete story. Experiments in the late 1990s and the early 2000s, such as SuperKamiokande, SNO, and KamLAND [1–3] have produced evidence of lepton flavor violation (LFV) in the neutrino sector through the mechanism of the so-called neutrino oscillation, or mixing. These results point to the existence of massive neutrinos, which, when accounted for in the extensions of the SM, allow for a possible, yet incredibly suppressed LFV decay in the charged



(a) Allowed (b) Forbidden

Figure 1.2: Allowed and forbidden decay in the Standard Model due to lepton flavor conservation.

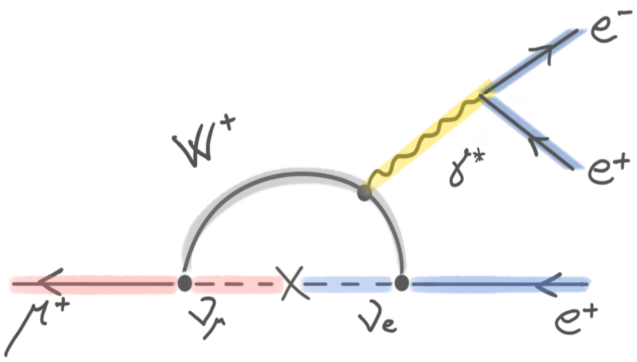


Figure 1.3: Highly suppressed neutrino-less decay of a muon to two positrons and an electron in the Standard Model.

lepton sector. For example, the branching ratio of a positive muon decaying into two positrons and an electron ($\mu \to e^+e^+e^-$) would be of the order of BR $\sim 10^{-55}$ [4, 5], making it inaccessible to any form of experimental investigation (Fig. 1.3). The reason for such a small branching ratio, even though the mixing angles in the neutrino matrix are measured to be large, lies in the fact that the neutrinos are much lighter than leptons. Additionally, the differences between the squares of masses of neutrinos are tiny compared to the W-boson mass [6].

Since the mechanism for and the size of LFV are unknown, this process opens a door for investigating neutrino mass generation and physics beyond the Standard Model. This is why theoretical physicists have introduced multiple new elementary particles in models such as Grand Unified models and Supersymmetric models (SUSY) [7], Triplet Higgs Model [8], Extended Gauge Sector [9], or Leptoquarks [10]. All of these give a much higher chance for a lepton flavor violating decay to occur, making it reasonable to look for it with the current and developing detector technology. An example of one such decay in the SUSY model is presented in Fig. 1.4.

As it is starting to become clear from the decays listed as examples above, one of the prime ways to look for lepton flavor violation in the charged lepton sector is to observe forbidden muon decays. The decision becomes clear when we realize

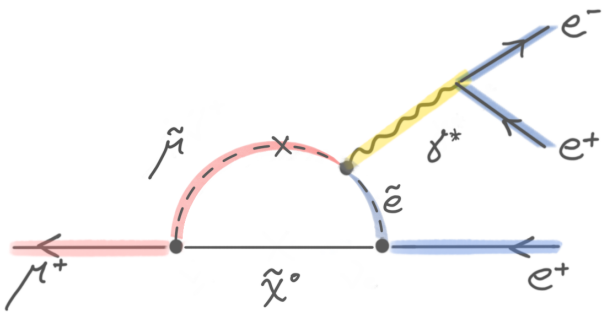


Figure 1.4: $\mu \to eee$ decay in a SUSY model.

that electrons, as the first-generation particles, do not decay any further, whereas the amount of taus that can be produced is several orders of magnitude smaller than that of muons. This is the reason why the path of searching for lepton flavor violation in muon decays has been taken by many experiments so far in different experimental channels, e.g., $\mu \to 3e$, $\mu \to e\gamma$, $\mu N \to eN$ (Fig. 1.5). They have been pushing the limits year after year, however, without success. Now, the time has come for a new generation of experiments to make the scene and push the frontier further.

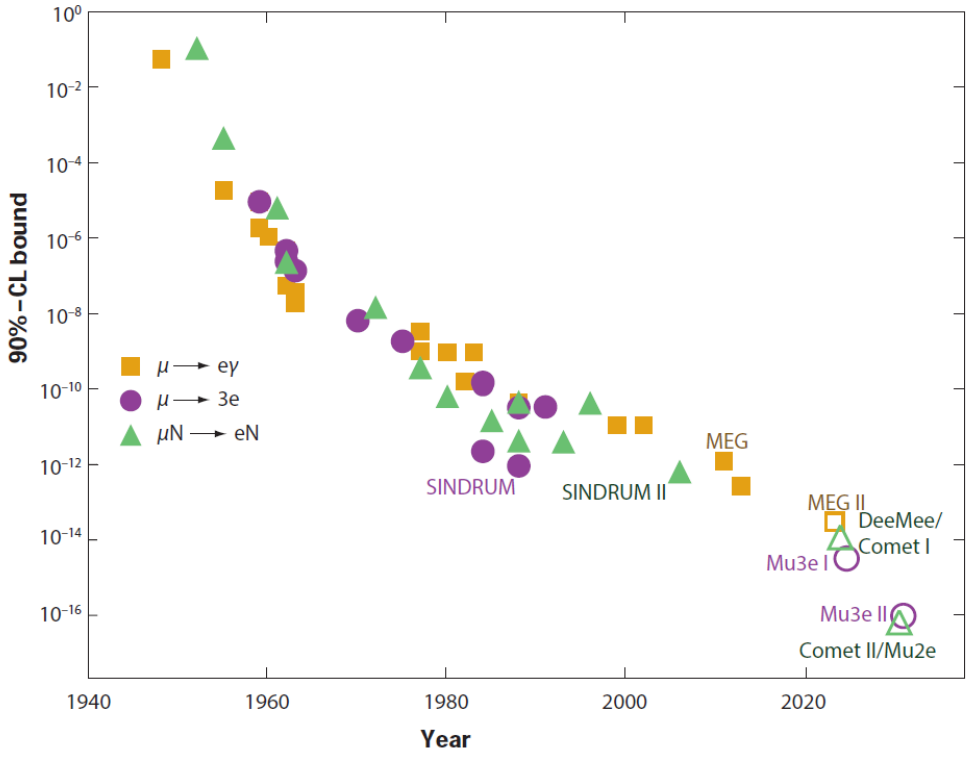


Figure 1.5: Conducted and proposed experiments searching for lepton flavor violation in muonic experimental channels. Solid markers stand for currently running or concluded experiments, whereas the hollow markers represent experiments under development. Adapted from [11].

Chapter 2

The Mu3e Experiment

The Mu3e experiment is one of the few experiments currently being built that aim at detecting a lepton flavor violating event or excluding a branching fraction of decay to a certain level. As its name states, Mu3e will hunt for a $\mu \to eee$ decay, or more specifically for a positive muon decaying into two positrons and an electron:

$$\mu^+ \to e^+e^+e^- \ .$$

In the case of the Mu3e experiment, single event sensitivity lies at a branching fraction of 2×10^{-15} for Phase I and 10^{-16} for Phase II of the experiment. In order to achieve the final goal, more than 10^{17} muons have to be stopped in the detector with a muon stopping rate of around 2×10^9 Hz and a background suppression to below 10^{-16}. Luckily, Mu3e will be located at the Paul Scherrer Institute (PSI) in Villigen, Switzerland, which houses the πE5 beamline, the highest intensity low-energy muon beam in the world (in use for Phase I), as well as the future HiMB (high-intensity muon beam) which is currently under study at PSI (planned for Phase II). In essence, the PSI synchrotron serves as a proton accelerator which accelerates the proton beam to 590 MeV. The beam impinges on a carbon target wheel, producing pions that then decay to muons which are guided to the experiment where they are implanted in a stopping target, and their decay products are observed. This results in a vast amount of muon decays which need to be sieved, recognizing and eliminating the background and finding the potential signal of a $\mu \to eee$ decay (Fig. 2.1).

2.1 Signal and background

Since the $\mu \to eee$ decay is prompt, observation of two positrons and an electron emanating from a single source at the same point in time is expected. Apart from this condition, two more apply, namely:

- The sum of four-momenta has to be equal to the muon mass;
- The three-momenta of the electron and the positrons has to add up to zero in the muon rest frame, which is equal to the lab rest frame, i.e.,

$$\sum \vec{p}_{tot} = \sum \vec{P}_i = 0 \, . \tag{2.1}$$

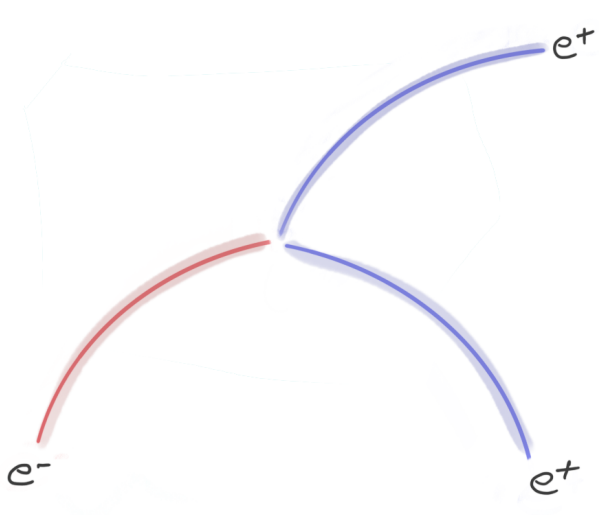


Figure 2.1: Representation of a $\mu \to eee$ decay.

As the decay happens at rest, the energies of the decay electron and positrons is in the range 0-53 MeV and the decay products all lie in a plane.

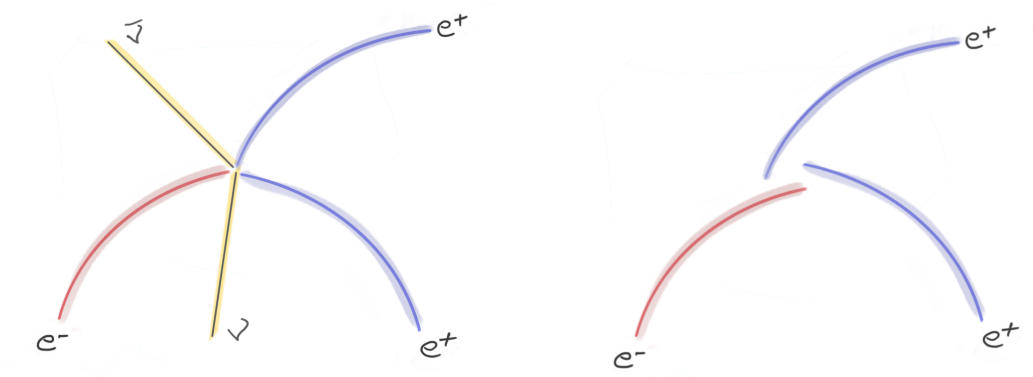
When searching for a rare decay event such as this one, potential backgrounds and their discrimination must be well understood. Two main background sources influence the sensitivity of the detector at hand, *irreducible backgrounds* and *accidental backgrounds*.

Irreducible backgrounds The dominating irreducible background effect arises from the internal conversion process $\mu^+ \to e^+e^+e^-\nu\overline{\nu}$ with a branching fraction of 3.4×10^{-5} (Fig. 2.2a). This process can be distinguished from the desired signal only by relying on momentum and energy conservation to reconstruct the neutrinos, which the detector cannot detect.

Accidental backgrounds The main contributor to the accidental background is also the main decay channel of the positive muon - the Michel decay $\mu^+ \to e^+\nu\overline{\nu}$. As it does not contain any negatively charged particles as decay products, it can only significantly contribute to the background if it coincides with additional processes that provide negatively charged particles, such as Bhabha scattering, photon conversion, or Compton scattering. An accidental combination of such processes can lead to a decay topology similar to the one expected from the signal (Fig. 2.2b). Luckily, these processes do not and cannot originate from the same point in space and time, which means that with superb vertex and time resolution, the background noise can be reduced, allowing the silent hum of the true signal to be found.

2.2 The Mu3e detector

The Mu3e detector set-up focuses on particle tracking, which allows for momentum and vertex reconstruction, as well as timing measurements - all integral parts of the $\mu \to eee$ decay detection and background suppression. The whole detector is located in a 1 T solenoidal magnetic field and consists of various modules, which can be seen



(a) Irreducible (b) Accidental

Figure 2.2: Dominant background processes in the Mu3e experiment.

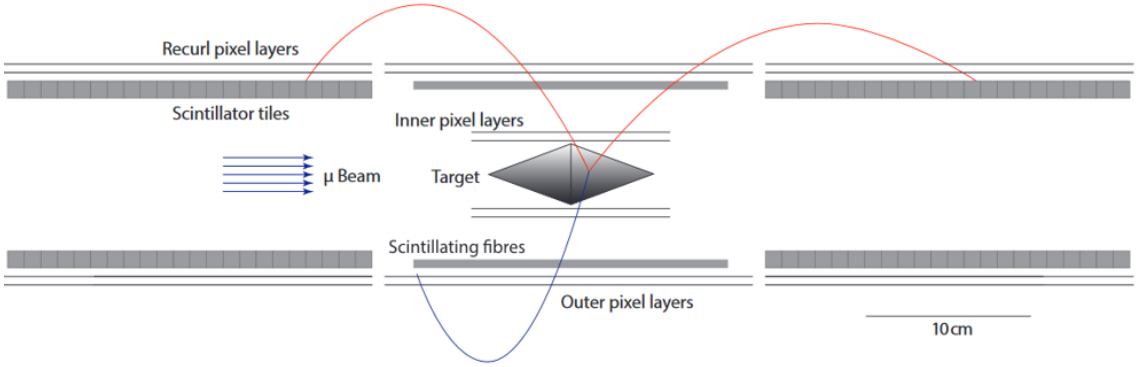


Figure 2.3: Detector schematics: A longitudinal cross-section of the central and first recurl stations - one on each side. Positrons are depicted in red and the electron in blue.

in Fig. 2.3. After the muons reach the experiment, they are implanted inside a hollow double-cone target, after which they decay at rest. Their decay products spread outwards and describe helical tracks as they are guided by the magnetic field. While curling, they traverse layers of pixel sensors that track their position, and scintillation fibers and tiles, which are used for precise timing measurements. Because of the detector geometry, at the end of their journey, they recurl back and hit the detector once again, which significantly reduces the dominating effects of multiple Coulomb scattering on the momentum resolution. The following paragraphs will be focusing on detector parts mentioned above in slightly more detail.

The pixel detector The pixel detector consists of two cylindrical double layers, i.e., the inner and outer pixel layers, which at their core consist of a multitude of silicon pixel sensors, or more specifically High Voltage Monolithic Active Pixel Sensors, so-called HV-MAPS (Fig. 2.4). HV-MAPS were excellent candidates for the Mu3e detector, as they integrate sensor and readout functionalities in the same device, allowing them to be thinned to just 50 µm, thus significantly reducing the material budget and with it the amount of Coulomb scattering present. Moreover, individual pixels are 80 × 80 µm which is much smaller than the contribution multiple scat-

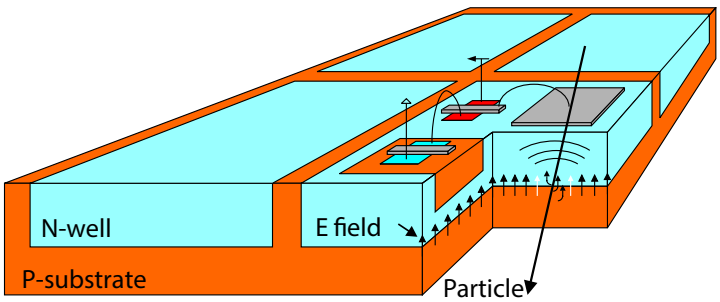
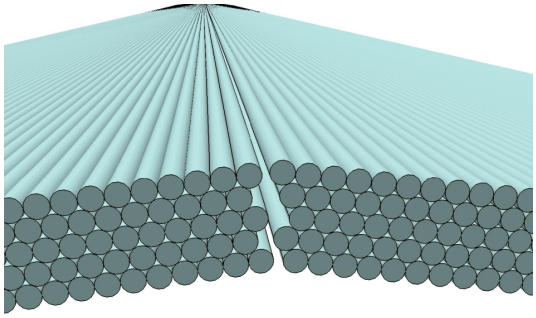
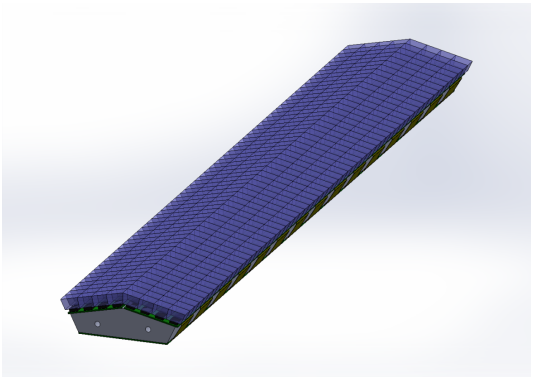


Figure 2.4: High-Voltage Monolithic Active Pixel Sensor (HV-MAPS) - key segment of the tracking detector.



(a) Cross section of Scintillating Fibres (Sci-Fi)



(b) Scintillating tiles module

Figure 2.5: Main parts of the time-of-flight detector.

tering has on the momentum resolution. In the context of the Mu3e experiment, pixel sensors are implemented in the form of MuPix chips which contain 256 × 250 sensors each. Intelligent design (4 individual layers) and positioning with respect to the target (one double-layer very close, and one further away, but not too far) allow for both redundancy during track-fitting (four points for fitting a curve instead of the usual three) as well as best possible vertex and track resolution for both lower and higher momentum particles [6].

The Time-of-Flight (ToF) detector In order to precisely measure the arrival time of particles, which helps with matching the hits detected in the silicon detectors, a two-part cylindrical time-of-flight detector is being constructed. The first part will consist of scintillating fibers (Sci-Fi) whose expected time resolution is of several 100 ps and the detection efficiency around 100% (Fig. 2.5a). The Sci-Fi detector will be placed in the central station on the inside of the outer pixel layer, which will help reject pile-up events and allow for charge measurements of recurling tracks. The second part of the ToF system will consist of scintillating tiles, which will be placed in the recurl stations on the inside of the recurl pixel layer (Fig. 2.5b). It aims at a resolution below 100 ps and an efficiency close to 100%, which will allow for identification of a coincidence signal from electrons and positrons and the suppression of combinatorial background. Both the scintillation fibres and scintillation tiles will be read out by sillicon photomultipliers (SiPM) [6].

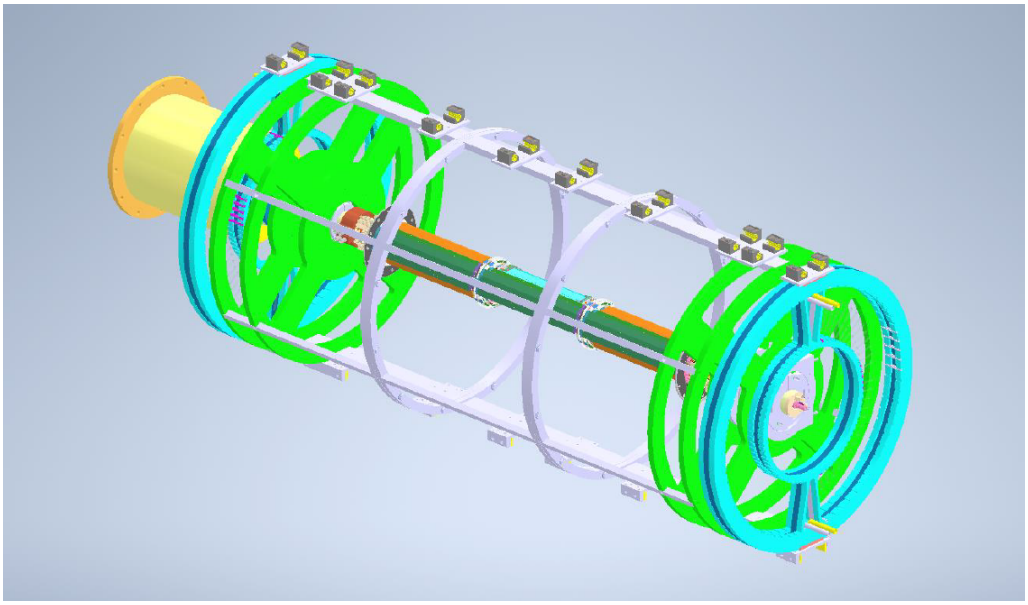


Figure 2.6: CAD rendering of the detector cage (white) and the detector inside of it.

The detector cage All of the detector parts are later integrated and mounted on a removable frame called the detector cage, whose diameter is 852 mm. It consists of two main ring frames on each side, made of pairs of glass-fiber reinforced polymer wheels, as well as aluminum struts connecting the two wheels and forming the cylinder structure (Fig. 2.6). Gliders on the wider struts help insert and extract the cage in and out of the magnet. Additionally, the cage carries the infrastructure such as crates for the power converters and the front-end FPGA boards, and provides support for all the cabling and piping. The cage with the detector can be seen in Fig. 2.6 [6].

In order to precisely reconstruct particle tracks and deduce their momenta and origin, it is vital to have individual detector elements (described in the paragraphs above) aligned as accurately as possible. The following chapters will be focusing on the misalignment effects, general detector alignment, and specific alignment techniques together with their benefits and drawbacks.

Chapter 3

Detector Alignment

The construction of a tracking detector is a highly demanding task. Research and development aside, the very process of machining, gluing, and assembling different detector elements, is paramount for a high precision experiment such as Mu3e. This is why knowledge of the quality and quantity of misalignment, together with the ideas that would lead to resolving them, has to be investigated.

3.1 Misalignment

An optimal scenario for the detector would be if all the detector elements ended up where they should and stayed in their place during the whole run of the experiment. However, this most certainly cannot be the case not only because of the limited mechanical assembly precision but also because of the environmental factors such as thermal expansion and gravitational effects.

Simulations using the misalignment software Mu3eMisal have been used to investigate different misalignment modes and their effects on track reconstruction. These have shown, that the misalignment originating from multiple factors during the construction, assembly, magnet ramping, and during the on-time of the detector, requires an alignment algorithm [12]. This algorithm would have to be able to correct for the position, orientation, and potential deformation of all the detector elements, as well as the general movements of whole detector parts relative to each other. This procedure needs to be carried out since the algorithms in charge of track and vertex reconstruction require knowledge of all of the variables above for all the active detector parts in order to perform track reconstruction with the largest possible precision.

3.2 Track-based alignment

One of the proposed alignment strategies is the so-called *track-based alignment* with the help of the Millipede II software [13]. The procedure for the barrel-like, multiple-layer detector is as follows. After individual particle tracks are observed and fitted with a model, *residuals* - differences between the measured hit and its corresponding prediction obtained from the fit - can be defined and calculated:

$$r_{ij} = m_{ij} - f(\boldsymbol{q}_j, \boldsymbol{p})\,. \tag{3.1}$$

Here, m_{ij} stands for one measurement and f, which depends on local track parameters $\boldsymbol{q}_j$ and global parameters $\boldsymbol{p}$, represents the track model, with i and j running over all hits in the track, and all tracks in the model, respectively. A graphical representation of all the variables can be seen in Figure 3.1. After the residuals have been defined, we can go one step further by summing over i and j, and define the χ^2 function

$$\chi^2(\boldsymbol{q}_j, \boldsymbol{p}) = \sum_{j}^{tracks} \sum_{i}^{hits} \left(\frac{r_{ij}}{\sigma_{ij}} \right)^2 , \tag{3.2}$$

where σ_{ij} denotes the uncertainty of the measurement and r_{ij} are the residuals defined in the previous step. The next and final stage is the point where the MILLIPEDE II software steps in and solves the minimisation problem of χ^2 in a single step obtaining an optimal combination of track and global parameters [12].

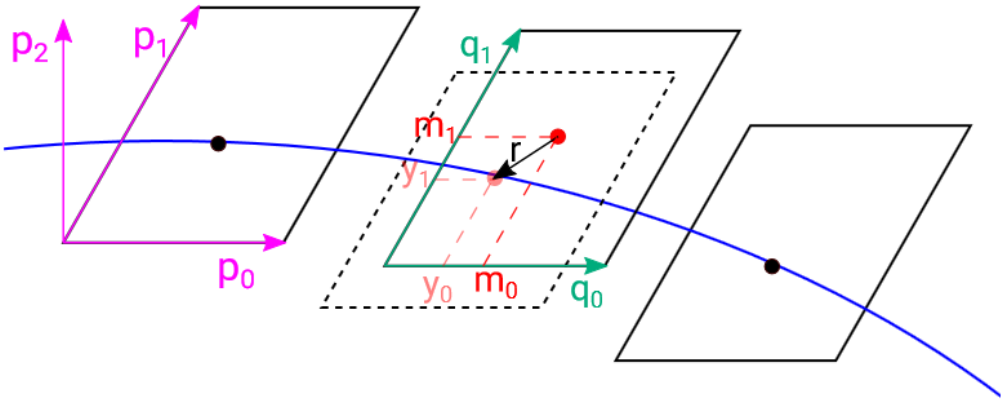


Figure 3.1: Schematics of a particle traversing three sensor planes. In magenta, a coordinate system representing one set of global parameters, in red and light red, the measured hit position and the predicted hit position, respectively, and in green, the local track parameters. The residual is represented as a black arrow [12].

3.3 Weak modes

The track-based alignment technique, however useful, has its drawbacks. One of them is the fact that there exist specific tracker deformation modes which do not influence the χ^2 (Equation 3.2), so-called *weak modes*. Weak modes can be visualized in an example of elliptical tracker barrel deformations (Figure 3.2). In this case, the true track will have a worse χ^2 than before the deformation; however, another track can be reconstructed with the same χ^2 as the original one, which is precisely the main characteristic of the weak mode.

Multiple weak modes might occur during the course of the experiment, some of which are merely expected from the geometry of the detector and some of which have already been observed in the simulation. In Figure 3.3 an incomplete list of weak modes which might appear has been presented.

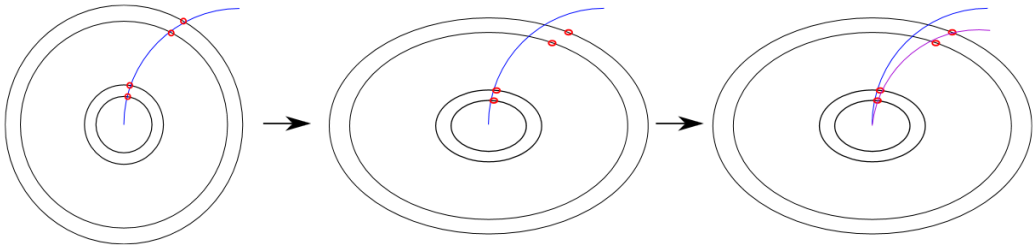


Figure 3.2: Weak mode in the case of elliptical deformations. The original track is depicted in blue, measured hits are the red circles and on the very right, after the deformation, a new track with the same χ^2 as the original one is represented in purple [12].

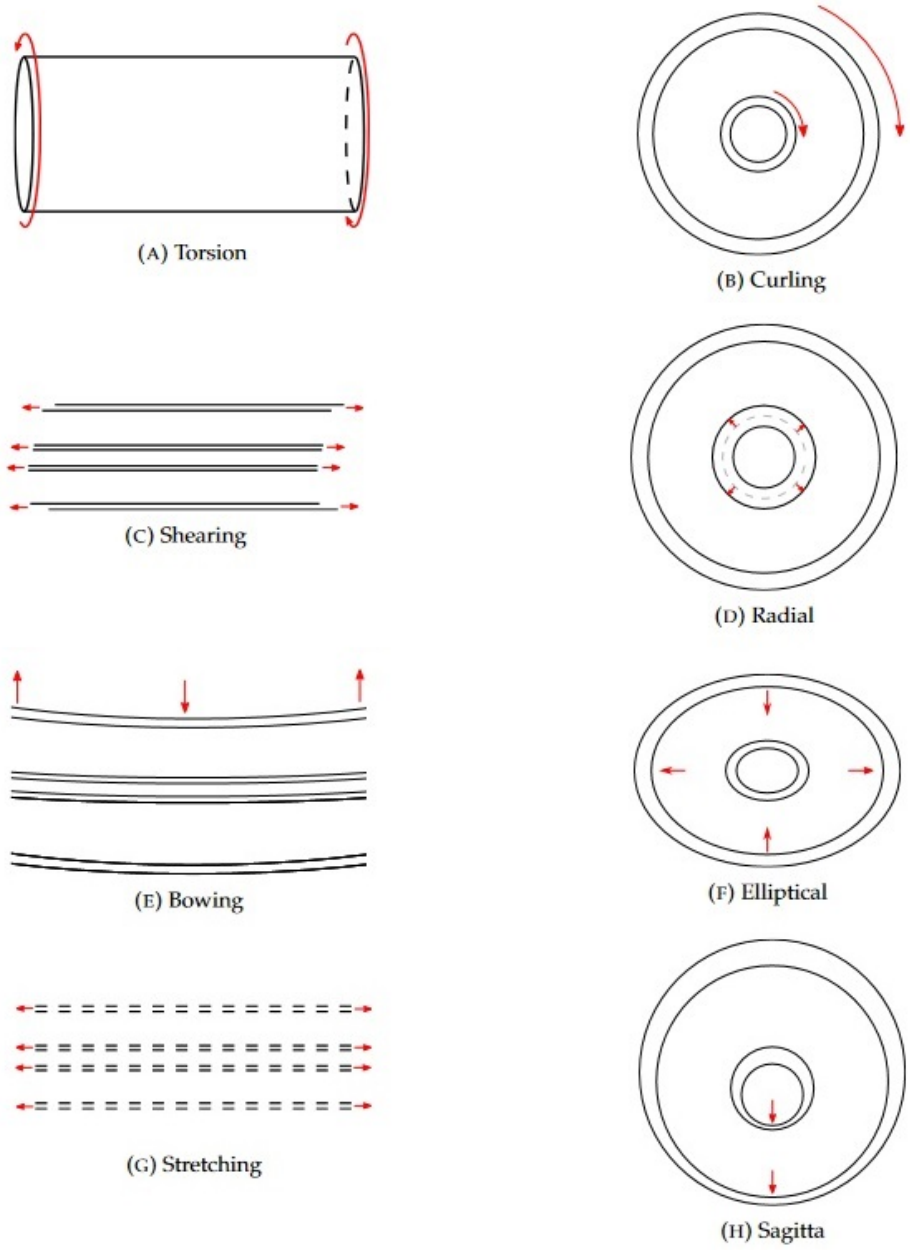


Figure 3.3: Different possible weak modes [12].

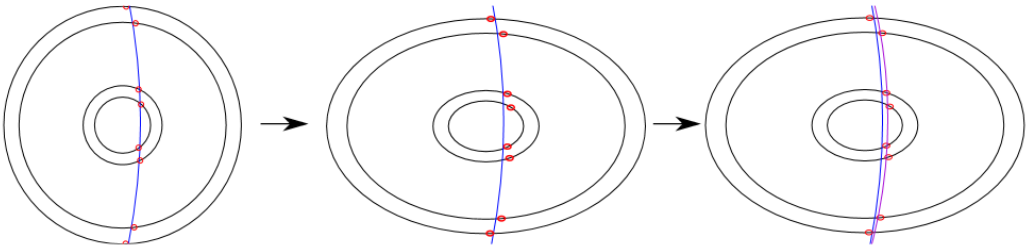


Figure 3.4: Muon tracks during the elliptical deformation. In this case, the χ^2 function is different before and after deformation (blue and purple) [12].

There are, however, several techniques that can be used to suppress weak modes in the Mu3e experiment. One of them utilizes tracks from cosmic muons, which, because of their very different topology from the expected tracks coming from muon decays in the target, offer insight into detector deformations by connecting detector parts that would otherwise not be connected (Figure 3.4). The same advantage comes from using reconstructed tracks from Mott scattering, as they have strong forward momentum.

Except for the above-mentioned techniques, in order to include even more information into the χ^2 function and the Millipede II software, and eliminate as many weak modes as possible, independent high precision position measurements of detector segments using cameras can be performed, which is exactly the main topic of this work. In the following chapters, the construction of a camera alignment system, its basic principles, as well as results obtained will be presented.

Chapter 4

Camera Alignment System

As discussed in previous chapters, proper alignment is crucial in high precision experiments such as Mu3e, where the results highly depend on how precisely particle tracks can be reconstructed and, therefore, to what precision the momenta and energies of particles can be estimated. To aid in this endeavor, a camera alignment system was constructed as part of this thesis. The core idea behind the system lies in the clever positioning of cameras around the detector cage, seen in Fig. 2.6, which could, by observing the detector and each other, infer the absolute position of detector elements in space. Within this chapter, basic principles of the position calculation, together with a method for detector-specific measurements will be explained, and the camera system that was used will be introduced.

4.1 Basic principles

At the center of position calculations lies a very well-known equation from the field of geometric optics, namely the *magnification formula*. In the case when an optical system with a lens is observing an object (Fig. 4.1), this formula connects the actual size of the observed object with its size on the image plane and the distance between it and the lens:

$$\frac{d_I}{d_{real}} = M = \frac{f}{f - L}\,. \tag{4.1}$$

Here, d_I and d_{real} represent the size of the image of the object and the real size of the object, respectively, M stands for the magnitude of magnification, f is the focal

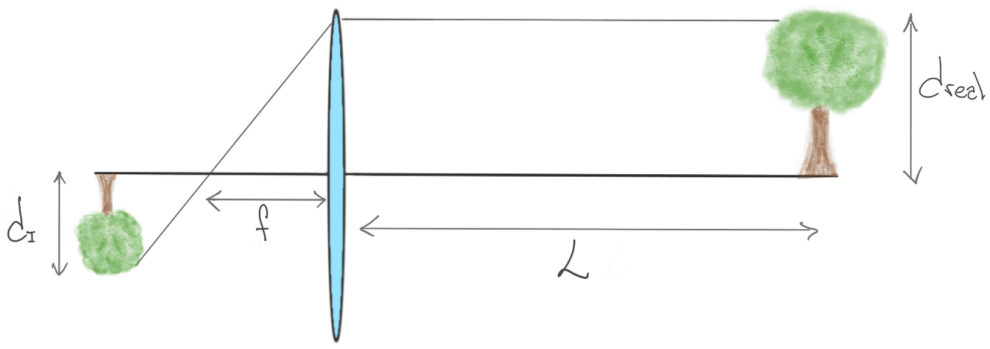


Figure 4.1: Geometric optics. Variables match those in the Eq. (4.1).

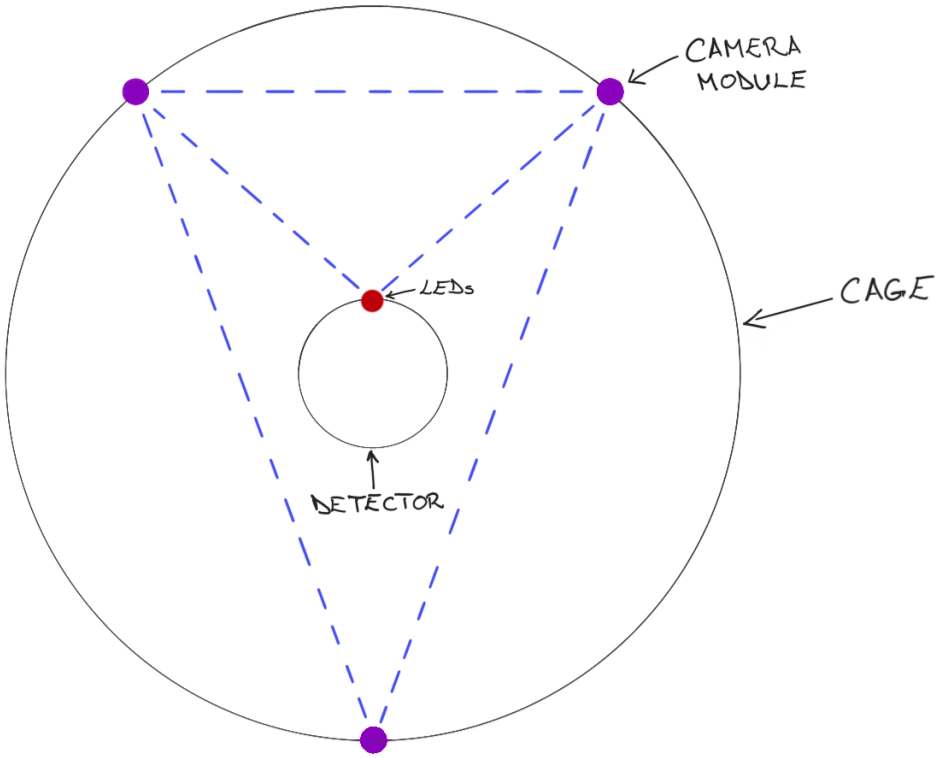


Figure 4.2: Schematics of the basic camera set-up with camera modules (IR-camera and two LEDs) on the detector cage and pairs of LEDs attached on the detector. It is important to notice that the cross-sectional view of the set-up is depicted.

length and L the distance between the camera lens and the object. Eq. (4.1) allows us to do the following. If the size of the observed object and the focal length of the camera lens are known, the distance between the camera and the object can be easily calculated:

$$L = f\left(1 - \frac{d_{real}}{d_I}\right). \tag{4.2}$$

Having this in mind, we can imagine the following scenario. By placing camera modules, which contain a camera and a pair of LEDs (purple), around the detector, and pairs of LEDs (red) on the detector (Fig. 4.2), multiple distances could be calculated, which would allow for precise determination of detector position in space. In this case, the distance between the pairs of LEDs would represent the size of the observed object mentioned in Eq. (4.2). A more detailed description of the procedure will follow in Chapter 6. Next, by placing the bottom and the top-left camera module along the y-axis of a 2D coordinate system, with the bottom camera module in the center of the coordinate system, as in Fig. 4.3, the following equations can be written down

$$x = \frac{c}{2} + \frac{c(r_0^2 - r_1^2)}{2D^2} + 2\frac{b-d}{D^2}\,\delta, \tag{4.3}$$

$$y = \frac{b+d}{2} + \frac{(d-b)(r_0^2 - r_1^2)}{2D^2} + 2\frac{c}{D^2}\,\delta, \tag{4.4}$$

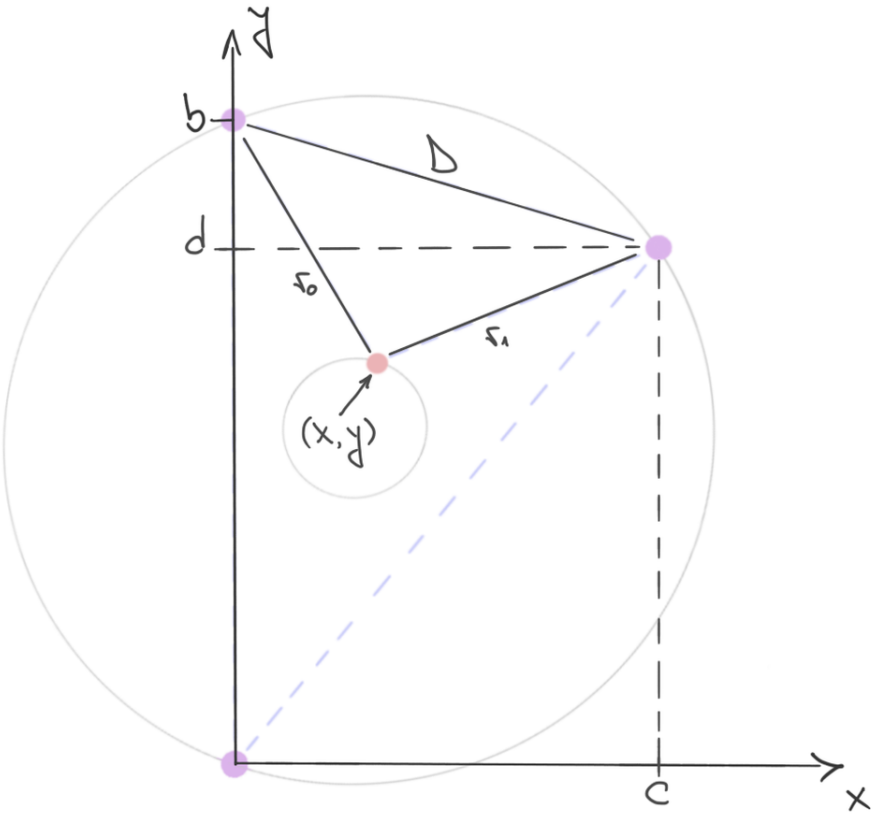


Figure 4.3: Schematics of the basic camera set-up with the corresponding coordinate system.

where $\delta = \frac{1}{4}\sqrt{(D+r_0+r_1)(D+r_0-r_1)(D-r_0+r_1)(-D+r_0+r_1)}$, with all of the variables matching those in Fig. 4.3. As a consequence, Eq. (4.3) and Eq. (4.4) allow us to determine the position of one of the points on the detector. If one repeats the method for multiple points it will result in a precise determination of detector shape and position in space.

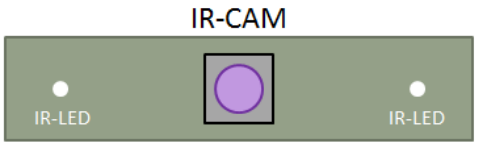
4.2 Camera modules

When constructing a camera module for alignment purposes in an experiment such as Mu3e, there are several limitations one should keep an eye on, e.g., size and weight because of spatial constraints, or the fact that the module should not disrupt any active components of the detector. The fact that the free space around the detector and on the detector cage is relatively narrow and that the ToF modules contain scintillating materials which respond to visible light, a decision was made to have the first iteration of the camera alignment system equipped with two main components:

- Raspberry Pi v2 NoIR cameras
- Camera modules

The Raspberry Pi cameras are capable of detecting infrared light, and the camera module will be equipped with a pair of infrared LEDs on each side of an infrared

(a) Raspberry Pi v2 NoIR camera



(b) Camera module

Figure 4.4: Main part of the camera alignment system - the Raspberry Pi NoIR camera, and a depiction of a camera module with the camera in the middle and two IR-LEDs on each side.

camera (Fig. 4.4). The LEDs will represent an object recorded by other cameras, whose size will help infer the distances between them. The basic characteristics of the IR camera are shown in Table 4.1.

One of the more important findings regarding the Raspberry Pi cameras was the fact that the factory focal length ($f = 3.04\,\text{mm}$) does not match the measured focal length and also fluctuates slightly from one camera to the next. This situation is not optimal as it is paramount to know the camera's focal length to high precision if the distances are to be calculated appropriately. Therefore, a camera calibration procedure is necessary before the cameras are mounted and the measurements had begun. The procedure does not require any special equipment, and all that is needed is to know the distance between the camera and a recorded object, together with the object's actual and image sizes (see Eq. (4.2)). Calculations have shown that the optimal scenario for determining the proper focal length of the camera is to

Table 4.1: Hardware specifications of the Raspberry Pi v2 NoIR camera.

Specs	Pi NoIR v2
Size	25 × 24 × 9 mm
Weight	3 g
Still resolution	8 Megapixels
Sensor	Sony IMX219
Sensor resolution	3280 × 2464 pixels
Sensor image area	3.68 × 2.76 mm (4.6 mm diagonal)
Pixel size	1.12 × 1.12 µm
Focal length	3.04 mm
Horizontal field of view	62.2 deg
Vertical field of view	48.8 deg
Price	$25

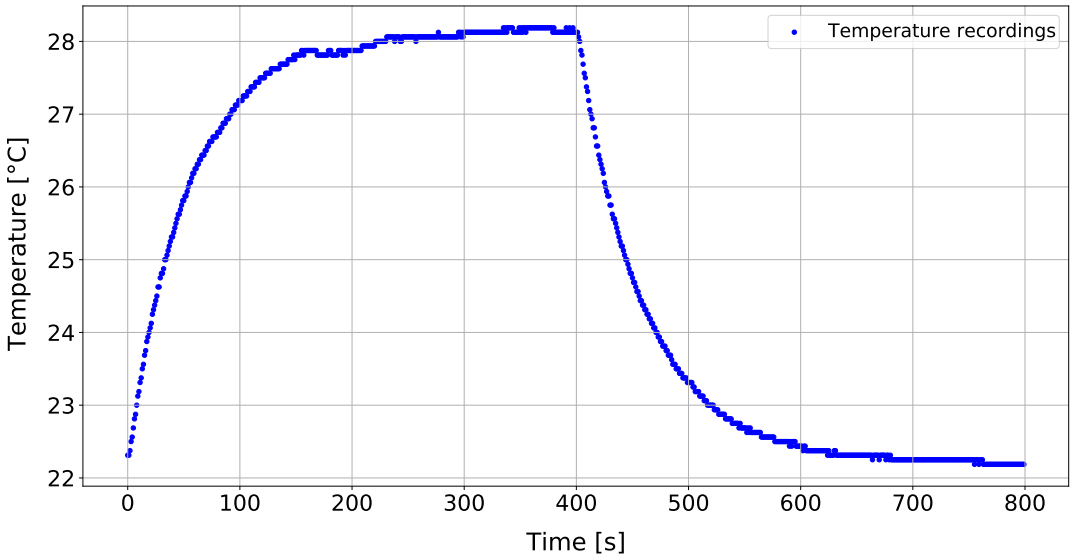


Figure 4.5: Heating up and cooling down of the camera and the surrounding PCB.

have it placed around 700 mm away from a 250 mm long object and to know the distances with a 5 µm precision (a caliper would suffice). An object of such a size would correspond to around 950px on the image plane, and if the size of the object in pixels is known to a precision of 0.033px (see Section 7.1.2), this will yield an error of the focal length of about 0.14 µm. Such precision is sufficient for all the calculations that include the focal length as a variable.

Another element of the puzzle one should keep an eye on is the heating up of the camera sensor and the PCB (Printed Circuit Board) it is attached to. The effect of the warm-up and the cool-down has been measured by attaching a temperature sensor on the PCB next to the camera, and it can be seen in Fig. 4.5. The conclusion is that the camera quickly heats up and then levels out at a certain point, with the same behavior occurring when the cool-down begins. However, when trying to capture a scene with an object in it (e.g., LED) while the camera heats up, the coordinates of the object change unpredictably (see Fig. 4.6). A correlation between the temperature change and the distortion of the camera sensor could not be clearly defined. This is why additional measurements had been done where it was shown that the camera is the most stable and unperturbed by temperature changes a few seconds after it has been turned on. As a result, it was decided to implement a recording procedure where the camera is turned on, left to heat up for 3 s, the photo is taken, and finally, the camera is left to cool down for around 30 s. After that, the recording procedure can be repeated.

Additionally, during the *Mu3e Integration Run 2021* at PSI, one Pi NoIR camera was placed in the same room as the detector, approximately 1 m from the experiment, and was continuously taking pictures while the beam was on, for $\sim$ 60 h. The camera shows no effects of radiation - no dead, stuck or hot pixels - which is encouraging but requires a more serious stress test where it will be mounted on the cage and inserted inside the magnet.

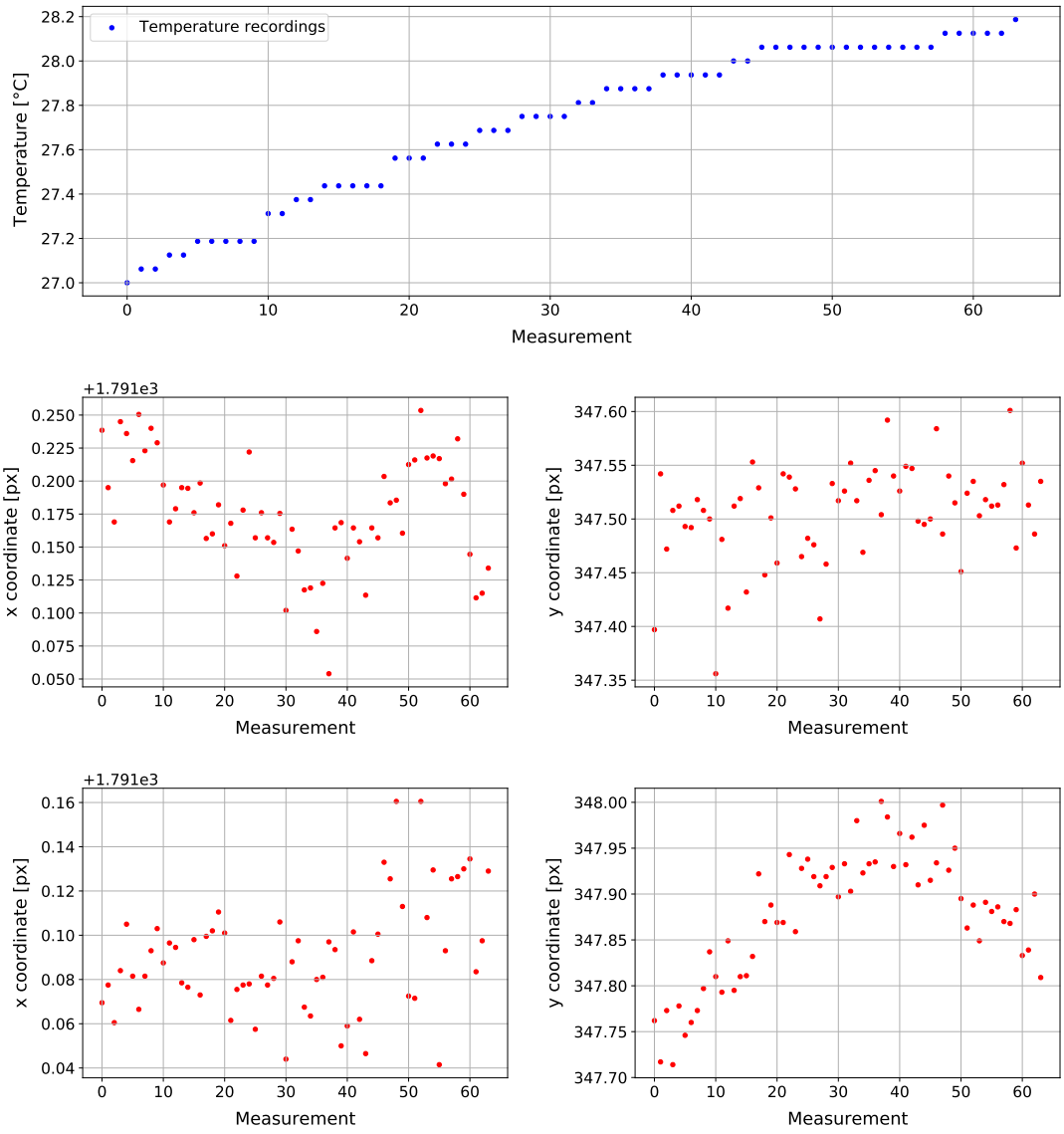


Figure 4.6: Investigation of the dependence between temperature and object (LED) position on the image. The single figure on the top represents the temperature range along which the test was done. Below are two different test runs. Figures on the left represent the x-coordinate of the LED, whereas the figures on the right depict changes in the y-coordinate. It is evident that as the camera heats up, the position of the LED fluctuates in two different ways for two different runs. Therefore, a clear correlation could not be drawn.

4.3 Object recognition

In trying to pinpoint the exact location of LEDs attached to camera modules, a need arises to develop an object recognition tool, as well as a tool capable of determining the distance between the LEDs. There are several steps involved. Firstly, a region of interest (ROI) in the image containing the bright spot that is the LED has to be selected. Secondly, the coordinates (in pixels) of the center of gravity of the LED need to be calculated. Moreover, the distance between two LEDs placed on the module has to be calculated, which in essence, gives us the final piece of Eq. (4.2).

Let us individually analyze the previously mentioned steps. In this work, the ROI was determined by taking a photograph of the scene and manually selecting the bounding box. The following images used the same ROI over and over again. Suppose that the individual detector elements and the detector cage do not move substantially during the course of the experiment. In that case, ROIs for LEDs can potentially be set only once at the beginning of the experiment and then slightly adjusted if the need arises. Scanning the whole image, discriminating the LEDs from the noise, and drawing bounding boxes for each camera shot would be much more error-prone.

Once the ROI is set, the center of gravity of the LED can be calculated. This is done with the help of a PYTHON-based computer vision package OPENCV. First of all, the noise in the ROI is subtracted using a thresholding method provided by the package, which allows the center of gravity to be calculated using image moments. In its essence, determining the center of gravity boils down to the following two equations:

$$x_c = \frac{\sum_i x_i I_i}{\sum_i I_i} \quad \text{and} \quad y_c = \frac{\sum_i y_i I_i}{\sum_i I_i}\,. \tag{4.5}$$

Here, I_i is the intensity of a pixel and it is calculated as $(I_R + I_G + I_B)_i/3$ where R, G and B stand for red, green and blue contributions to the pixel intensity, x_i and y_i represent x- and y-coordinates in pixels, respectively, and the index i runs over all pixels in the ROI. This procedure allows us to calculate positions of LEDs in the image to a sub-pixel precision.

Finally, once both the LEDs on the camera module have had coordinates associated with them, the distance between them in pixels can be determined. If the distance on the PCB between them is known, the distance between the camera taking the photo and the camera module, which is the object of the image, can be determined.

Chapter 5

Simulation

The alignment system at hand is one where there are multiple cameras involved in the set-up. Calculating their position and the position of a detector depends on what the cameras see and how well they can resolve different details in the image. Before the system is set up, simulating different scenarios is preferable in order to determine whether an idea is implementable and, if so, to have certain expectations considering the system's behavior. As we are dealing with photographs and projections, utilizing a ray-tracing algorithm proves itself to be very useful. The following sections will give a short guide to the ins and outs of such a technique and present the obtained results.

5.1 Ray-tracing

Ray-tracing is a rendering technique often used in generating video game graphics, which can realistically simulate a 3D scene. To do so, a 3D space, encompassing objects of the scene, light sources, a point-like camera, and a projection screen, has to be simulated (Fig. 5.1).

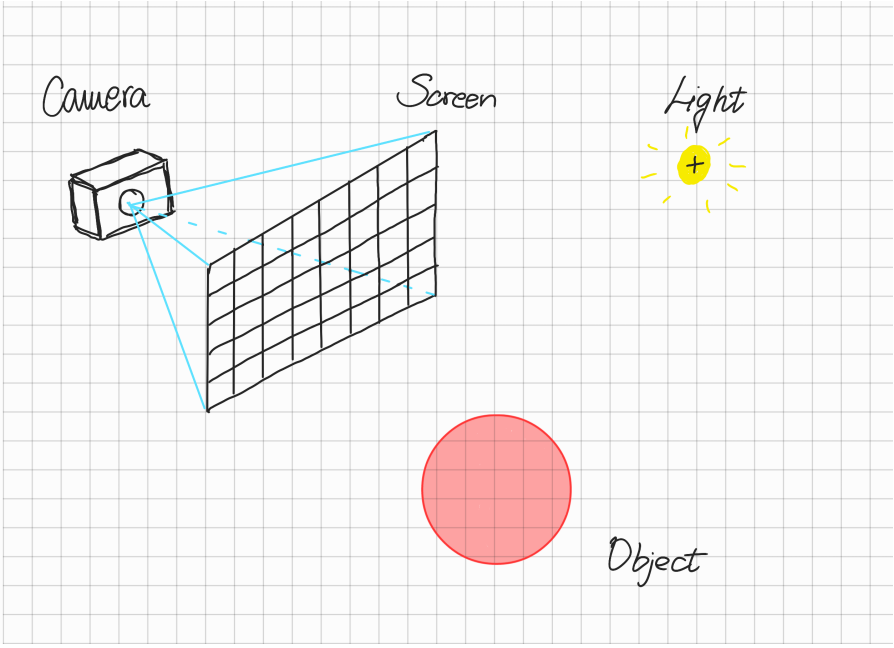


Figure 5.1: Basic simulation scene.

In the following lines, an idea of the ray-tracing algorithm is presented as pseudo-code.

Algorithm 1: Ray-tracing algorithm

```
Result: 2D projection of a 3D scene
for each pixel p(x, y, z) of the screen: do
    associate a black color to p;
    if ray from the camera going towards p intersects any object of the scene
     then
        calculate intersection point to nearest object;
        if no object between intersection point and light then
            calculate color of intersection point;
            associate color to pixel p;
        end
    end
end
```

In the case of a ray-tracing algorithm, one might notice that the ray does not exactly follow the natural path from the light source, bouncing off an object and hitting the camera sensor. Instead, only the rays that are sure to hit the sensor (screen in our case) are taken into account, and their paths are traced backwards to objects of the scene and the light source in the end. This is done in order to reduce the computational load, which occurs when the light illuminates the whole 3D space, now and then finding an object on which it will scatter and potentially hit the sensor. The number of rays would be too much to handle, which is why the process is essentially reversed.

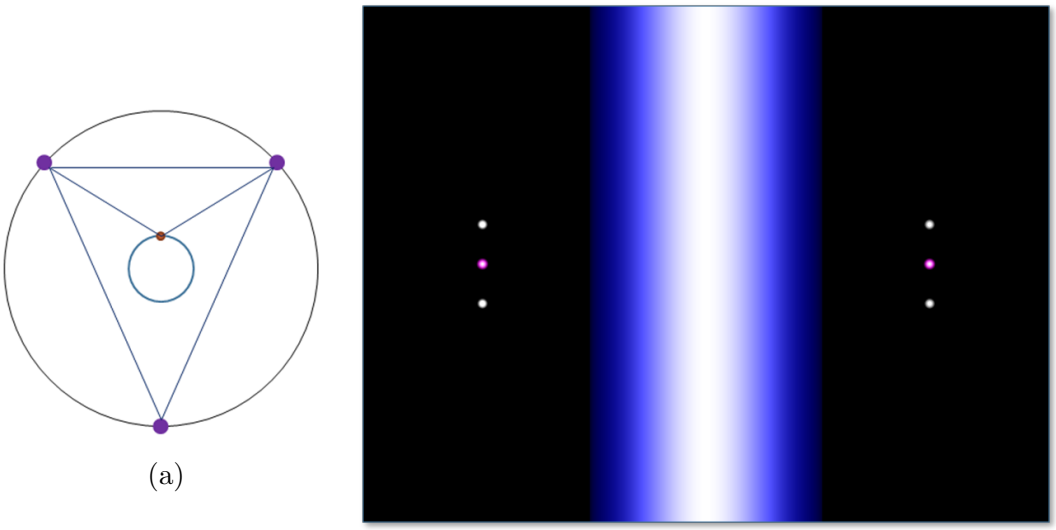
The fascinating fact about the ray-tracing technique is that it does not require much more than a few basic Python packages and some knowledge of linear algebra. The only non-intuitive segment, which will be explained in more detail, is the coloring of the objects in the scene (pixels on the screen). The model which was used to color the pixels is called the *Blinn-Phong model*. According to it, all materials/objects have four properties - ambient color, diffuse color, specular color, and shininess - which are represented with red, green, and blue values ranging from 0 to 1. Given the properties of individual objects, the color which is ascribed to a pixel representing a part of an object is calculated in the following way

$$I_p = k_a\, i_a + k_d\, i_d\, L\, N + k_s\, i_s \left(N \cdot \frac{L+V}{\|L+V\|} \right)^{\frac{\alpha}{4}}, \tag{5.1}$$

where a, d and s stand for *ambient*, *diffuse* and *specular*, respectively. Constants denoted with k are the properties of the object, the ones denoted with i are the properties of the light source, and α is the shininess of the object. **N**, **L** and **V** are the unit normal vector to the surface of the object at the intersection point, direction unit vector from the intersection point to the light source, and direction unit vector from the intersection point towards the camera, respectively.[1]

Once the code is interpreted and the ray scans the screen looking for objects,

[1]The ideas for and the execution of the simulation came from https://medium.com/swlh/ray-tracing-from-scratch-in-python-41670e6a96f9.



(a)

(b)

Figure 5.2: Rendered image of the detector and the camera set-up.

the image shown in Fig. 5.2b is obtained. With the help of Fig. 5.2a, one can see that the algorithm rendered the image from the point of view of the bottom camera pointing upwards. The detector is depicted as a cylinder, as expected, and there are two camera modules, one on each side, positioned in such a way that the IR-LEDs (white) are placed at the top and the bottom of each camera (purple).

In Fig. 5.3, we can see a slightly different view of the set-up, since the image was rendered from the camera in the upper-right corner according to Fig. 5.2a. This means that the LEDs placed on the detector (red) are now visible, together with a camera module in the upper-left corner of the set-up scheme (right side of the image). These two views represent the true power of a ray-tracing algorithm by letting us pan around a 3D space and make projections of the same scene from different angles.

Once the image is rendered, the procedure of calculating the distance between the camera modules using geometric optics can be executed. This, in turn, yields one iteration of a simulation and produces a result that tells us how far away is the calculation of the distance from a set distance. The procedure is then repeated for multiple distances and hundreds of tiny movements in transverse and longitudinal directions. Such a routine gives an insight into the overall behavior of the system. Its results can be seen in the following section.

5.2 Simulation results

The main goal of the simulation was to determine the expectations of camera resolution both in the longitudinal direction and in the transverse plane, as well as the cumulative precision in 3D space, i.e., what kind of sensitivity towards movements could potentially be detected.

At the beginning of each simulation run, a camera module which was the object of investigation would be placed at a specified distance from the camera which was

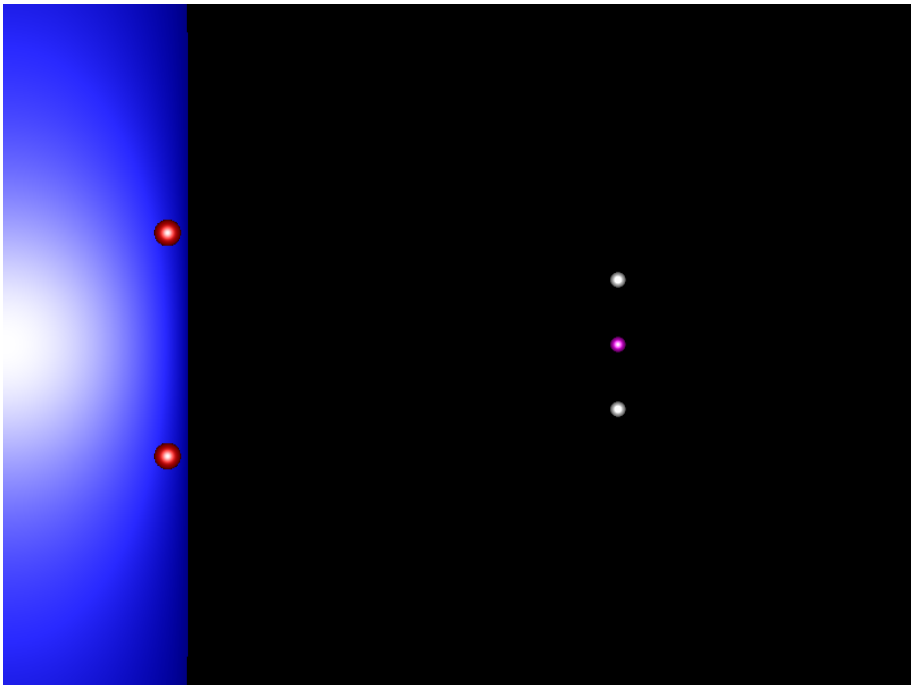


Figure 5.3: Rendered image of the detector and the camera set-up from a different angle.

taking the photo[2] and then moved backwards in 5 µm steps. After each rendering, an algorithm responsible for finding the centroid of the LEDs would be run and the formula derived from geometric optics (Section 4.1, Eq. (4.1)) would be used to calculate the distance from one camera module to another. The simulation regarding the test of the longitudinal resolution was run four times, at four different distances which are expected to be present in the real experiment - 400 mm, 500 mm, 600 mm and 700 mm. The same was done in the transverse plane - more specifically, only in one of the transverse directions, namely the x-axis, for distances of 50 mm, 100 mm, 150 mm and 200 mm from the center of the field of view of the picture-taking camera. This was done because both the resolution in the x- (left-right) and in the y-direction (up-down) should be comparatively similar inasmuch as the pixels of the screen have a square shape and the focal length in both directions is the same. It should also be noted that the transverse simulations were conducted with the camera module positioned at the maximal longitudinal distance of 700 mm. By doing this the worst-case scenario was taken into account, since, as the modules move towards each other, the transverse resolution can only increase - it is easier to resolve movement.

Finally, the general precision of the system in three dimensions was tested, however, in a slightly different fashion. The camera module, which was the object of investigation, was randomly positioned in space after each "picture" was taken, thereby allowing for the whole 3D space to be covered and giving slightly more intuition on the general possibilities of the camera modules.

The results of the longitudinal test at 400 mm can be seen in Fig. 5.4, whereas the rest of the results are listed in Table 5.1 together with their corresponding mean values and standard deviations.

[2]Taking a photo means going through the procedure of once scanning the space of objects through the pixels of the screen and rendering an image.

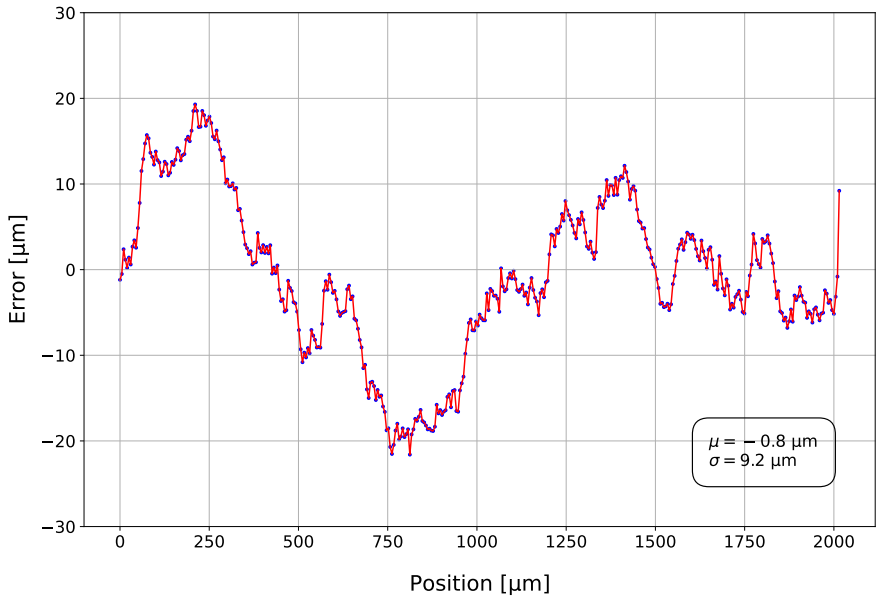


Figure 5.4: Simulation results for a test of the longitudinal resolution with an initial position at the distance of 400 mm and 5 µm step movements.

z	Initial distance [mm]	Mean [µm]	Standard deviation [µm]
1	400	-0.8	9.2
2	500	4.0	13.4
3	600	4.8	7.7
4	700	-3.8	18.6

Table 5.1: Results of the simulation for a test of the longitudinal resolution with different initial positions.

As far as the results of the test of transverse movements go, they are displayed in Fig. 5.5, whereas the rest of the results, similar as before, are listed in Table 5.2.

As was previously mentioned, another way of analyzing the data coming from the simulations was randomly picking positions of the camera module in 3D space and calculating the error of the distance calculation between the camera taking the picture and the one which is being randomly positioned. For each snapshot, projection to the z-axis, together with the distances in the x- and the y-axis from the center of the screen, was determined. After that, the total distance D was

x	Initial distance [mm]	Mean [µm]	Standard deviation [µm]
1	50	0.05	1.7
2	100	0.06	1.8
3	150	0.09	1.7
4	200	0.22	1.8

Table 5.2: Results of the simulation for a test of the transverse resolution with different initial positions.

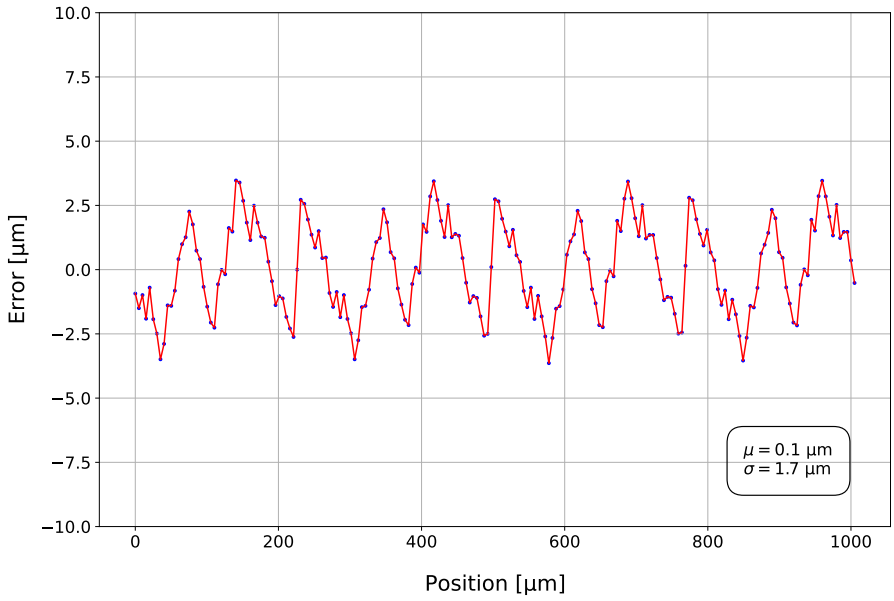


Figure 5.5: Simulation results for a test of the transverse resolution with an initial position in the z-axis at the distance of 700 mm, in the x-axis at 150 mm, and 5 µm step movements towards the right side of the screen.

calculated using the following equation

$$D = \sqrt{x^2 + y^2 + z^2} \ . \tag{5.2}$$

Values in the direction of the x-axis were ranging from 0 to 250 mm, of the y-axis from 0 to 10 mm, and of the z-axis from 400 to 700 mm. Following the previous step, an error between the true distance and the calculated one would be determined. This method resulted in plots presented in Fig. 5.6 and Fig. 5.7. The projection onto the x-axis (Fig. 5.7) points to an increase of the distance error as the values of the projection onto the z-axis increase.

The results of the simulation give a clear indication that if a camera alignment system is to be constructed in a fashion similar to the one described and investigated in this chapter, it can be expected that it performs in reality as it was intended to, at the resolution similar to, or below the required one (80 µm). Analyses of the mean and the standard deviation of residuals in each simulation configuration also indicate that no internal biases or systematic errors contribute to the results; instead, the visible fluctuations are purely statistical and depend on the picture (screen) resolution.

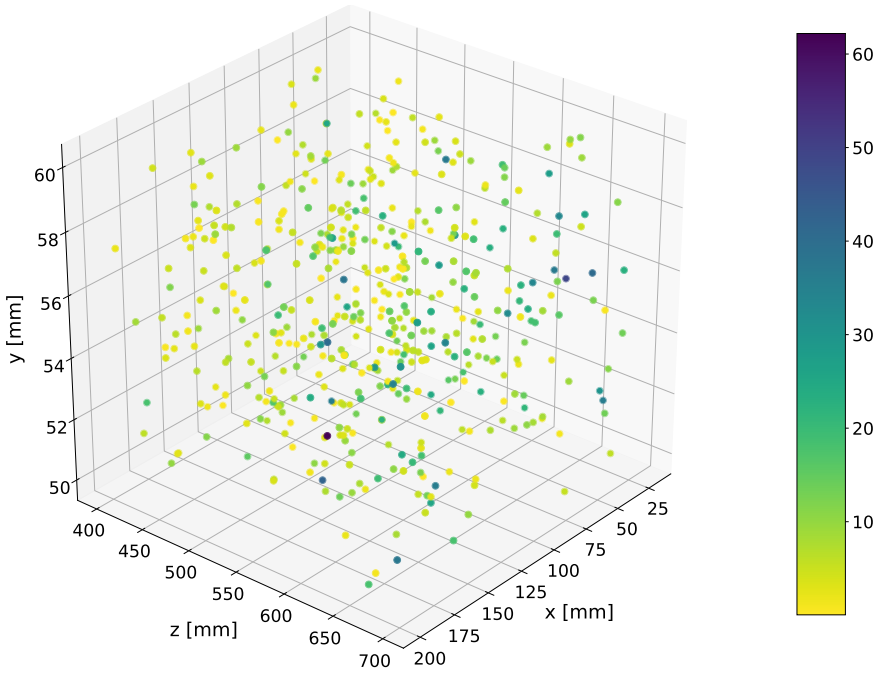


Figure 5.6: Simulation depicting a random scan of the 3D space. The color map refers to the absolute value of the error of the calculated distance D compared to the true one in µm.

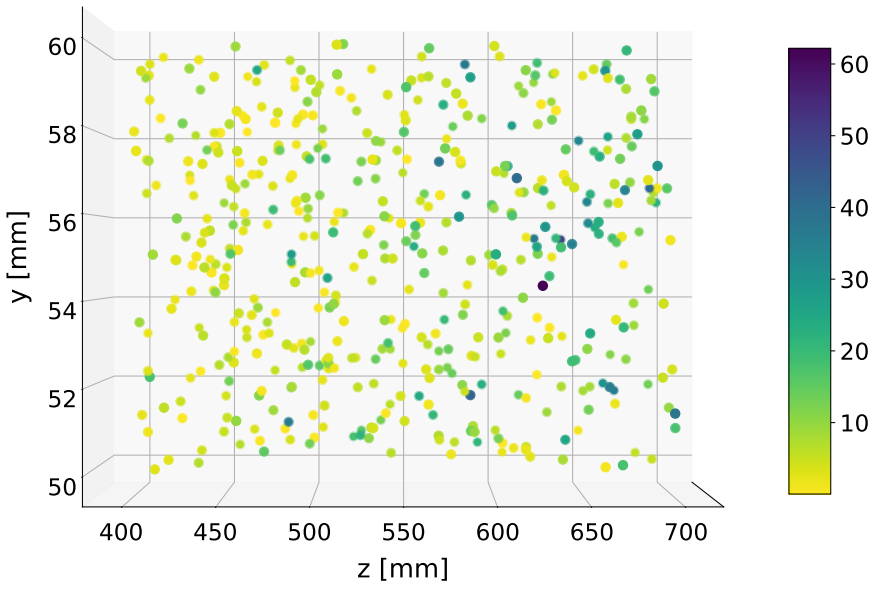


Figure 5.7: Simulation depicting a random scan of the 3D space, projected onto the x-axis. The color map is the same as in Fig. 5.6.

Chapter 6

Experimental Set-up

The following chapter will give an overview of different experimental set-ups, and the ideas behind them will be introduced. The primary purpose of the conducted experiments was to systematically approach the problem of camera alignment and address the difficulties that might arise along the way in constructing such a system. In the final phase, a set-up resembling the optimal solution for the actual implementation in Mu3e was developed and tested.

6.1 Camera resolution

For the purpose of comparing the results acquired from the simulations with the real-world system, a set-up was constructed using the MakerBeam kit[3] representing the detector cage, a Raspberry Pi NoIR v2 camera, LEDs, and a holder which could be moved in micrometer steps using a micrometer screw. The photos of the set-up for testing the longitudinal and transverse resolution can be seen in Fig. 6.1 and Fig. 6.2.

As can be seen in the photos, the LEDs were emitting red color, which would, in essence, go against the demands of the Mu3e experiment. Nevertheless, such a decision was made so that the experiments could be run without completely darkening the room the set-up was in. Testing of the camera resolution does depend on the color of the LED; however, the wavelength of the red and the infrared light differ by a fraction of a micrometer and should be well below the resolution of the camera that is being investigated. Therefore, the choice did not render any significant hindrance to the results. Additionally, when testing the transverse resolution, one LED was sufficient, whereas when testing the longitudinal resolution, two were necessary to be able to infer the change in the distance between the camera and the LED-holder.

6.2 Camera module rotation

A significant difficulty in accurately calculating distances between cameras and objects (LEDs) stems from the fact that a camera module, which has two LEDs attached to each side of it, is susceptible to rotation around its attachment point to

[3]MakerBeam B.V. https://www.makerbeam.com

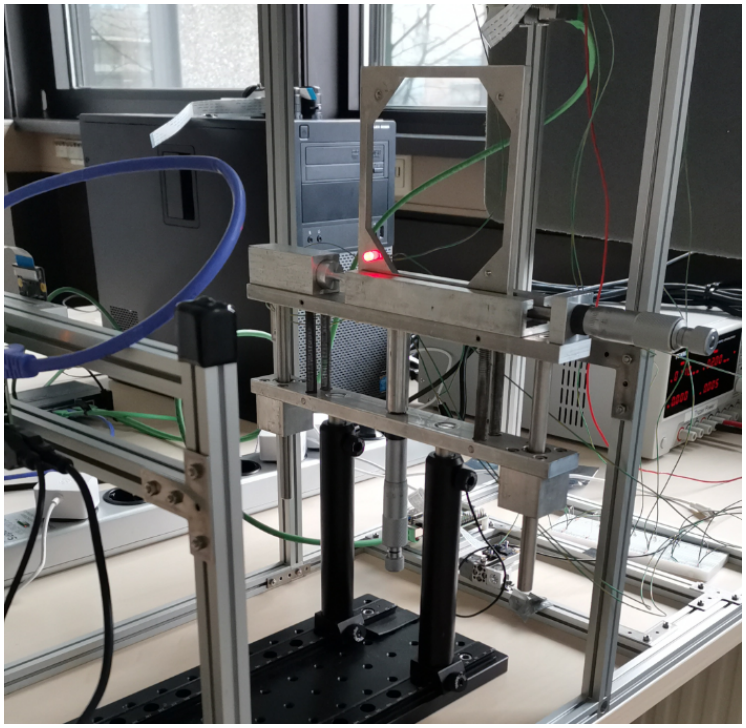


Figure 6.1: Experimental set-up for testing the transverse resolution of the camera. The camera can be seen in the middle on the very left side of the image. The holder with the LED attached to it and the micrometer screw on its right side are also visible.

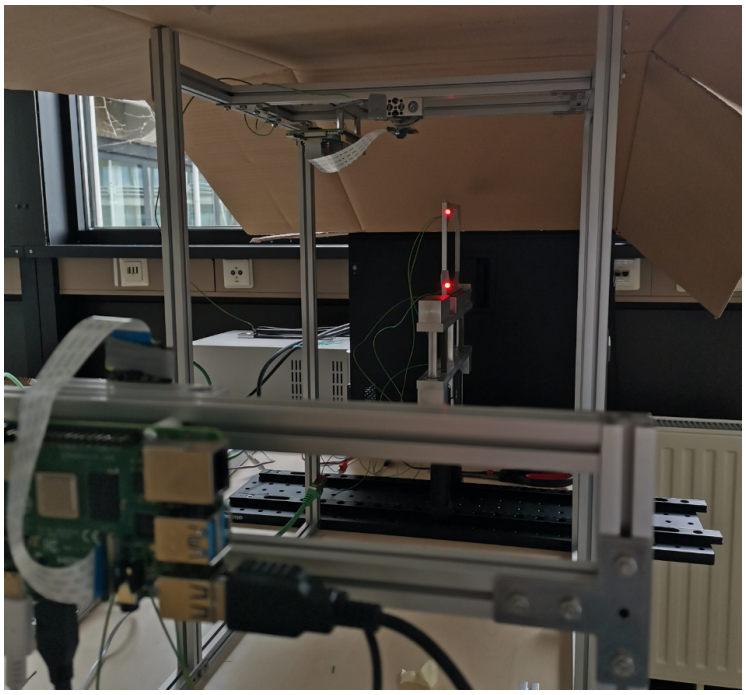


Figure 6.2: Experimental set-up for testing the longitudinal resolution of the camera. The set-up is the same as before with the exception of two LEDs being attached to the holder instead of just one.

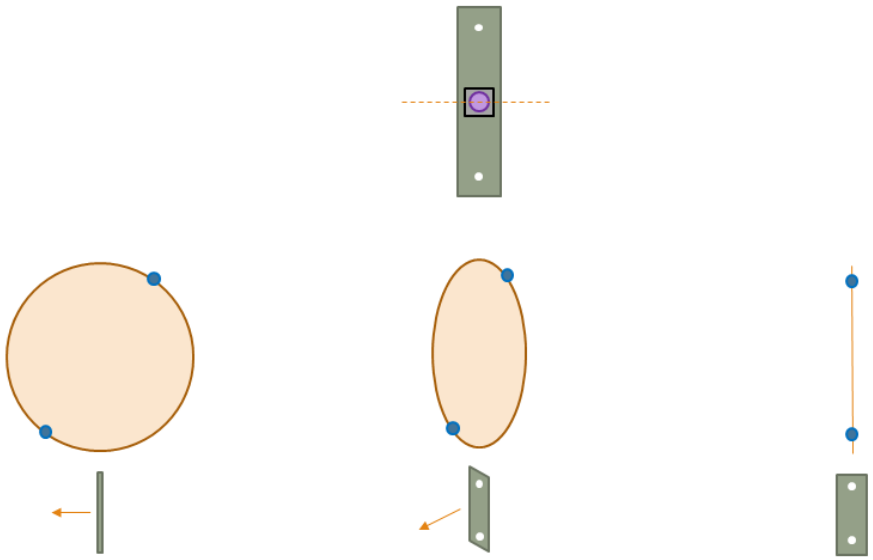


Figure 6.3: Depiction of the rotation problem. Top: A camera module and the axis of rotation. Bottom: Different situations in which the camera module is facing the picture-taking camera at different angles. Realistic scenario is the middle one where the camera module is slightly slanted and the LEDs describe an ellipse.

the aluminum struts of the detector cage (see Fig. 2.6) - the attachment point usually being at the very middle of the camera module. Such a movement distorts the distance between the LEDs, making it appear smaller and thus producing a false result for the distance between the camera taking the photo and the rotating camera module on the detector cage.

Luckily, the line the LEDs describe as they rotate around the attachment point is not a straight one; that is to say, camera modules are attached at an angle with respect to all the other ones, allowing one to determine the actual distance between the LEDs even in the situation where the apparent distance might be deceiving. The previously discussed scenario is portrayed in Fig. 6.3 and explained in more detail in the caption. In reality, the camera module will be at an angle between 0° and 90°. This fact allows us to transform the apparent ellipse the LEDs describe (second scenario in Fig. 6.3), into a circle (first scenario in Fig. 6.3). The circle can then be taken advantage of, as the circle's diameter represents the actual distance between the LEDs. The formula for the transformation is derived from the sketch in Fig. 6.4 in the following way:

$$x_c = \frac{x_{ell}}{\sin\alpha}\,, \quad y_c = y_{ell} \quad \text{and} \quad d_{LED} = 2\sqrt{x_c^2 + y_c^2} \tag{6.1}$$

leading to

$$d_{LED} = 2\sqrt{\left(\frac{x_{ell}}{\sin\alpha}\right)^2 + y_{ell}^2}\,. \tag{6.2}$$

Here, x_{ell} and y_{ell} represent the coordinates of one of the LEDs with respect to the center of rotation, calculated as the center of gravity between the two LEDs,

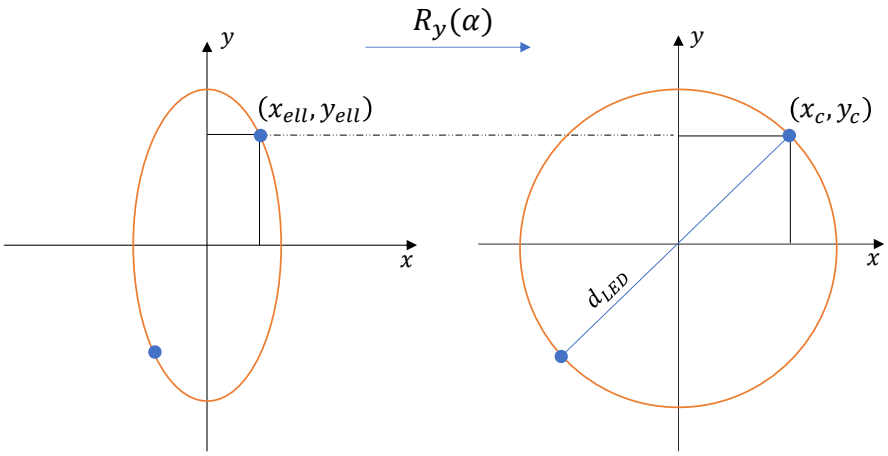


Figure 6.4: Sketch of the rotation problem used for the derivation of Eq. (6.2)

whereas α stands for the angle at which the module is slanted when observed from the camera that is trying to determine the distance (bottom row in Fig. 6.3).

To test the problem of rotation, a part of the set-up was rearranged so that the angle α at which the LEDs face the camera that is taking the picture is well-known, as this will also be the case when the modules get attached to the detector cage. As can be seen in Fig. 6.5, the sides of the angle α (yellow) are formed in such a way that the one side is parallel to the horizontal aluminum profile, whereas the other side is perpendicular to the shorter aluminum piece (blue) - the piece around which the LED-holder rotates.

The course of this part of the experiment was conducted in a way that the holder with the LEDs was photographed multiple times in a variety of inclinations - straight, with the top LED moving towards the camera, and with the top LED moving away from the camera. The different scenarios are depicted in Fig. 6.6.

6.3 The complete system

Once the resolution of the cameras has been tested, and the rotation problem addressed, the next step was to try and construct a version of the complete system that could describe its advantages and drawbacks.

For this purpose, an experiment resembling the one in Fig. 4.2 was set up. With three camera modules and multiple pairs of LEDs, with their holders placed on the central aluminum profile which substituted the Mu3e tracking detector, it aimed to reconstruct the conditions one might expect in the real experiment. This time, the experimental procedure was conducted in a completely darkened room; however, for practical reasons, the LEDs were once again emitting visible light. The experimental set-up can be seen in Fig. 6.7 and Fig. 6.8.

As was discussed in Chapter 3, the real-world detector can undergo elliptical deformations, and the elliptical shape of an object can be reconstructed with a minimum of five points. For this reason, two additional pairs of LEDs were added to the existing three on the aluminum profile. The LEDs were slightly moved around between different sets of shots with the aim of providing a proof of concept that the deformations of the detector truly can be precisely measured.

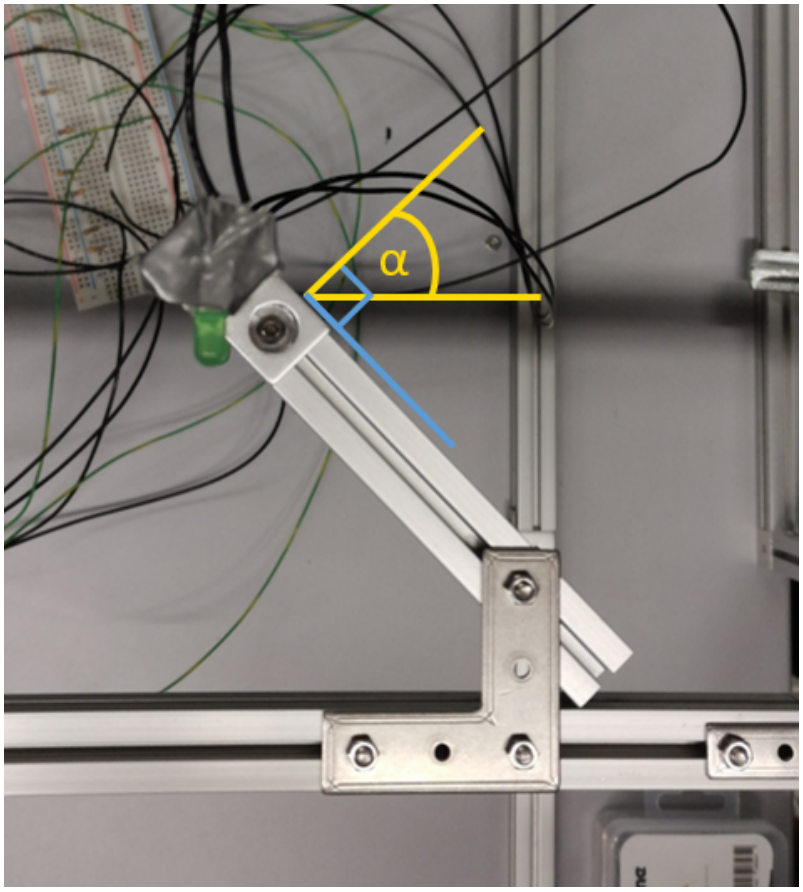


Figure 6.5: Experimental set-up for testing the rotation of the camera module together with a visual representation of the angle α.

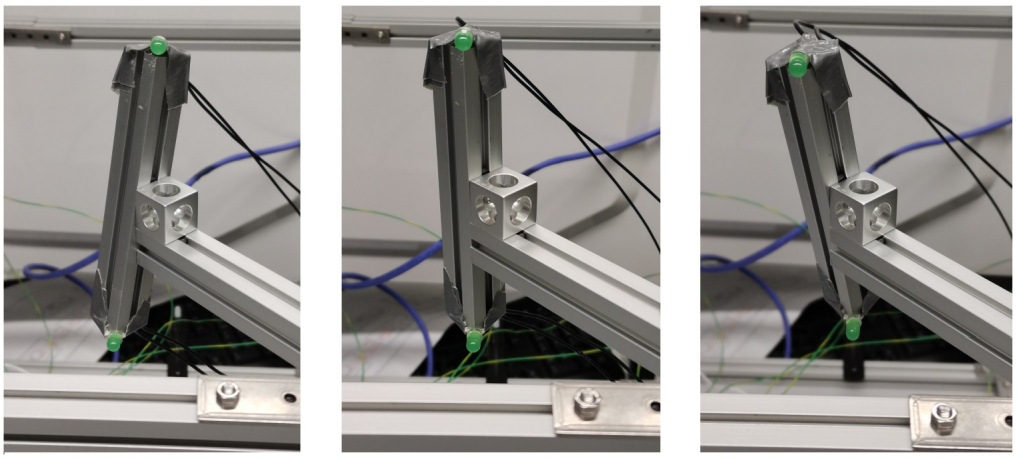


Figure 6.6: Three representative rotational positions of the module (holder with the LEDs).

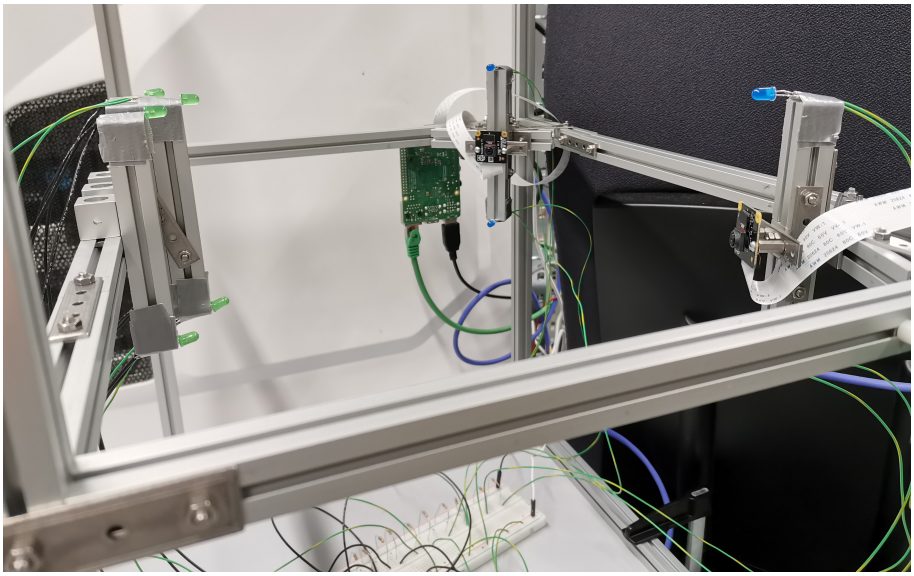


Figure 6.7: Experimental set-up for testing the complete system as seen from the side.

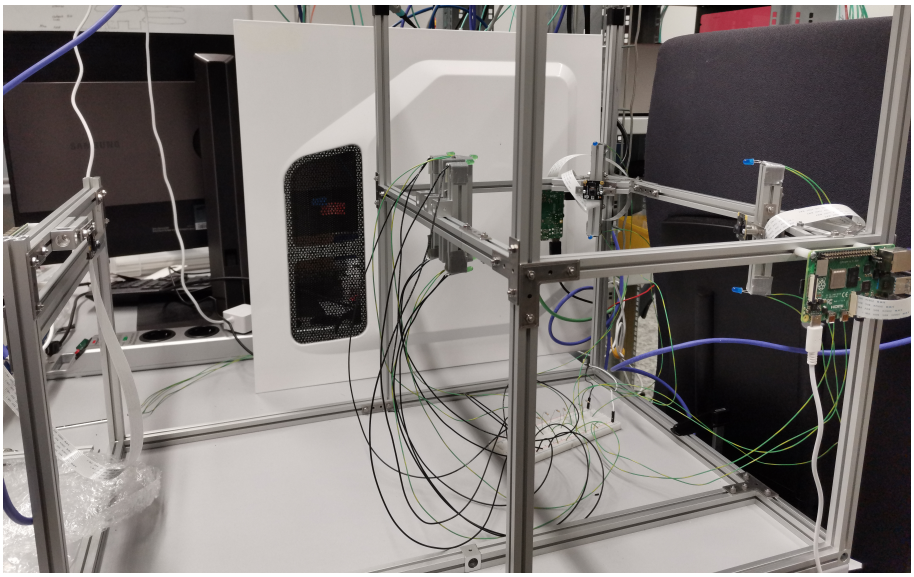


Figure 6.8: Expanded view of the experimental set-up with all of the camera modules and LEDs in view.

The following chapter will be dealing with the particulars of experimental set-ups, the obtained results, and their interpretations for all of the above-discussed frameworks in the order of mention.

Chapter 7

Results and Discussion

7.1 Camera resolution

In this part of the experiment, both the longitudinal and the transverse resolution of the camera were tested with the LED holder placed at around 400 mm from the camera. This distance was sufficient as it was not important where exactly the camera resolution was measured. The results which were obtained in pixels could later be translated into more intuitive units such as micrometers, and give insight into the precision of the camera at multiple different distances from it.

7.1.1 Transverse resolution

The measurements for testing the transverse resolution were as following. The movements between different holder positions were made in 20 µm steps, the precision of the micrometer screw was 10 µm, and each measurement of the LED position was repeated 10 times. The errors attributed to the measurements were calculated as standard errors of the mean

$$SEM = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \overline{y})^2}{N\,(N-1)}} = \frac{\sigma}{\sqrt{N}}\,, \tag{7.1}$$

where y_i represents individual measurements of the coordinate of the LED, $\overline{y}$ their mean value, and N the number of measurements. SEM was taken as a metric as it gives a better understanding of the relationship between different means. The results can be seen in Fig. 7.1, and it is important to notice that the horizontal errors represent the precision of the micrometer screw, whereas the vertical ones stand for the SEM which was previously discussed. The linear fit (red line) was made with the help of the Python package SciPy and the ODR method - short for Orthogonal Distance Regression. The sign of the slope of the fit is not important in this context as it only represents the general movement of the holder (LED) in the image - in this case from right to left. Once the data were fitted, the residuals were calculated as

$$r_i = \overline{y}_i - f(x_i)\,, \tag{7.2}$$

where $\overline{y}_i$ stands for the mean value of the coordinate of the LED, and $f(x_i)$ represents the value of the fitted linear function at x_i. Thereafter their mean and standard

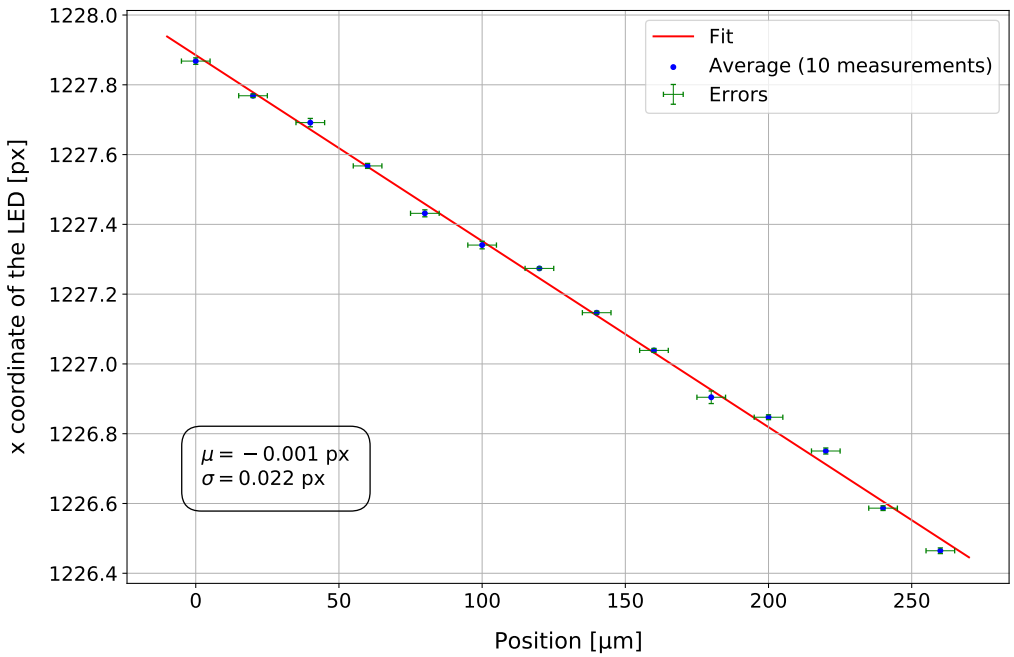


Figure 7.1: Test results of the transverse resolution of the camera.

deviation could be determined. This procedure resulted in a standard deviation of $\sigma = 0.022\,\text{px}$ and a mean of $\mu = -0.001\,\text{px}$.

7.1.2 Longitudinal resolution

A similar procedure was followed in the case of the longitudinal resolution, with the exception of the orientation and incrementation of the LED-holder, which was rotated by 90° and moved in 50 µm steps.

The results can be seen in Fig. 7.2 and the interpretation of the plot follows the same reasoning as in the previous case. This time however, the linear fit has a positive slope which means that, in this experiment, the holder was moved towards the camera (the apparent distance between the two LEDs increased). The standard deviation was calculated to be $\sigma = 0.033\,\text{px}$ and the mean $\mu = -0.012\,\text{px}$.

Moreover, the points exhibit a wave-like behavior around the fit. The behavior is also visible in the survey of the transverse resolution, but it is not as pronounced. The phenomenon can be attributed to the discreteness of the camera sensor. Namely, as the objects move across the image plane, they do not do so smoothly but rather in minute discrete jumps from one pixel to the next. The fact that the center of gravity of the LEDs is calculated reduces the effect, but it does not eliminate it completely. Similar behavior can be seen in the investigation of the target monitoring system by the group from the MEG II experiment, however, with no detailed explanation or review of the occurrence [14].

7.1.3 Discussion

The results can be interpreted by using the values of the standard deviation that came from the experiments, for the calculations of the error propagation (see Appendix A for the complete derivation). This then lets us estimate the expected error

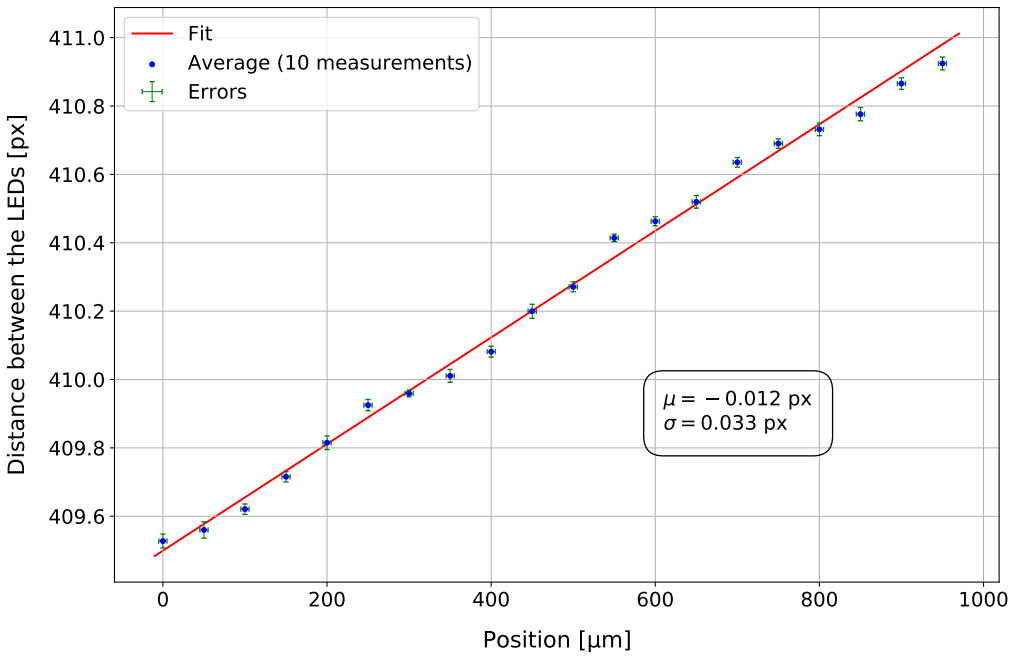


Figure 7.2: Test results of the longitudinal resolution of the camera. The SME can not be seen because it is too small compared to the precision of the micrometer screw.

at multiple different distances from the camera in a three-dimensional manner. Standard deviations for x- and y-direction were chosen to be the same (see Section 5.2), the error of the focal length had a value of 0.14 µm (see Section 4.2) and the distance between the LEDs was chosen to be 100 mm - a reasonable size in terms of detector cage demands. 1000 points were selected at random in a 3D-space with bounds corresponding to those in a real-world scenario ($x \in [-300\,\text{mm}, 300\,\text{mm}]$, $y \in [-100\,\text{mm}, 100\,\text{mm}]$, $z \in [200\,\text{mm}, 800\,\text{mm}]$). The resulting scatter plot of the expected errors when determining the distance can be seen in Fig. 7.3, where the color range represents the expected range of errors in the experiment.

As one could assume, the resulting errors are slightly larger then those presented in Section 5.2. This discrepancy is most probably owed to the fact that the standard errors in simulations are essentially non-existent - each measurement can be repeated multiple times with exactly the same results. Moreover, the dissimilarity of the result uniformity coming from the experiment and the simulation is visible and it can be attributed to a slightly different course of action that led to them. In the simulation, coordinates of each point were calculated and then compared to a true value, whereas the plot in Fig. 7.3 was generated with the help of the method of error propagation.

On a final note, even though the dominant factor in determining the 3D distance error comes down to the error of the projection of the distance in the z-direction (longitudinal error), the other two errors, namely the one in the x- and the one in the y-direction (transverse errors), should not be neglected. The reason becomes clear when the error of the calculated three-dimensional distance is plotted against the projection of the distance in the z-direction. The plot in Fig. 7.4 has a two-fold interpretation. Not only does it give insight into the relationship between the errors in the z-direction and the total errors, but it also shows that there is a range

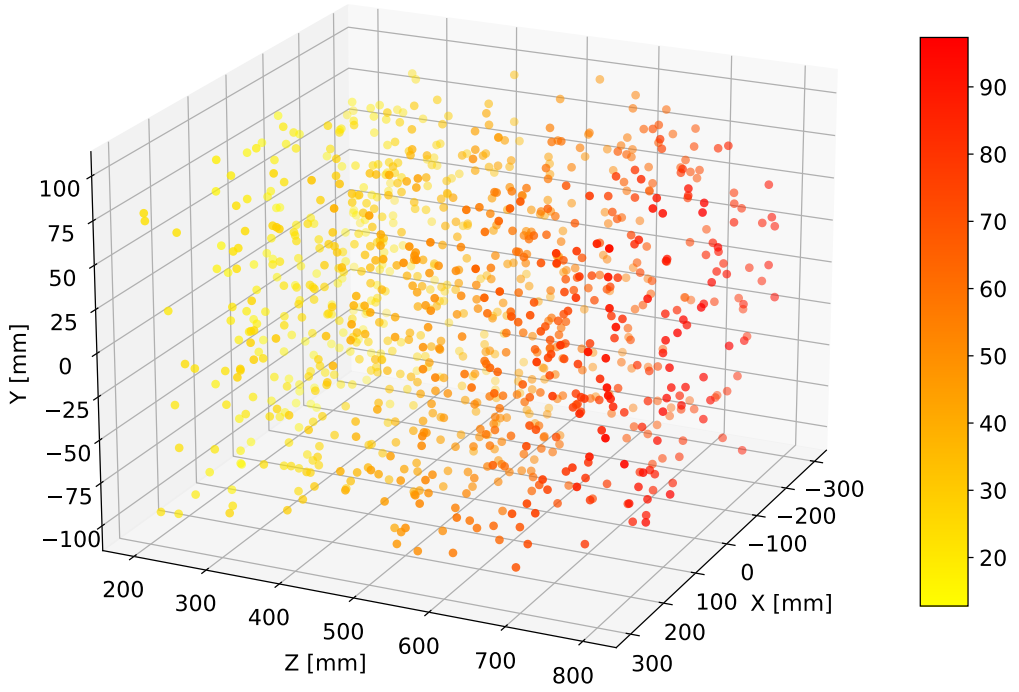


Figure 7.3: Results of the 3D error analysis. The color map represents the expected error for distance measurements in µm.

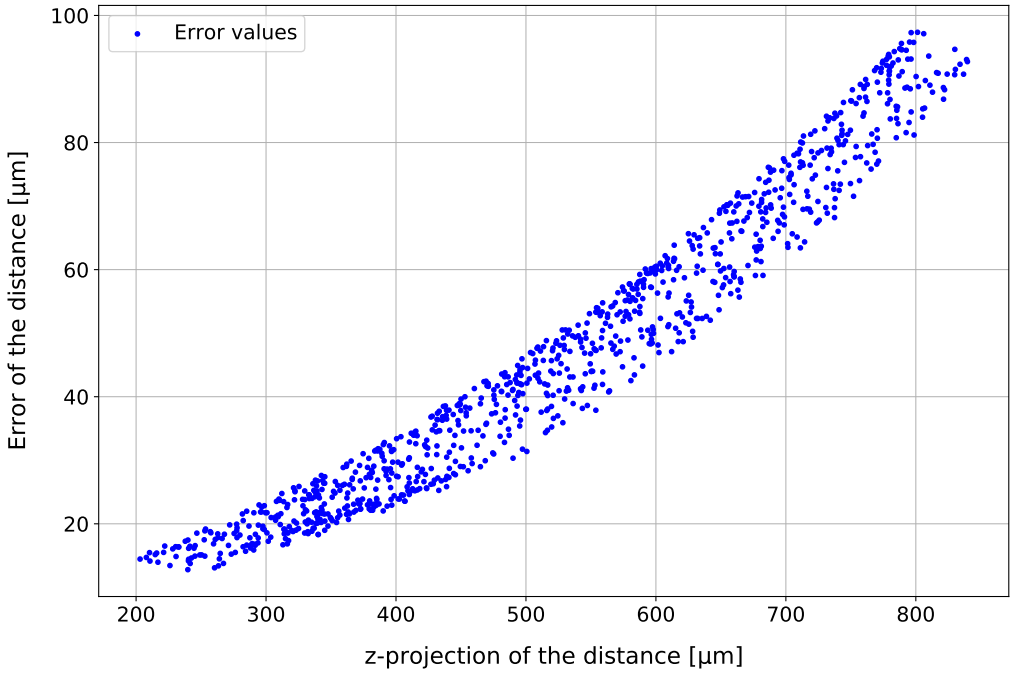


Figure 7.4: Results of the 3D error analysis in the z-projection. The band which the errors describe indicates that the exact position of all of the coordinates is important for precise error determination.

of error values, as large as 15 µm, for each z-distance from the camera, which in fact describes the influence the errors in the x- and y-direction have. Therefore, when the measurements are done and the positions of the camera modules or the LEDs are being determined, it is important to take into consideration not only the total calculated distance in three dimensions, but also how big of a role each of the coordinates played in the calculation.

7.2 Camera module rotation

The problem of rotation was investigated in order to provide a proof of principle for the conceptual solution discussed in Section 6.2. The experiment was conducted for a single angle of attack (α from Section 6.2) and multiple inclination angles of the camera module. Fig. 7.5 shows the path the LEDs trace as they rotate around the attachment point. Each point, once again, represents an average of 10 measurements. Additionally, the coordinates of the center of rotation depicted in the plot are also calculated as the average value of all the centers of rotation, which moved about slightly as the experiment was conducted. Error bars have been omitted because of the impossibility to represent them visually. In comparison to the movements that were tested, they were negligible.

The second approach to the analysis was to show if the formulas and ideas developed in Section 6.2 actually pan out. In order to do so, for all the inclination angles in Fig. 7.5, an apparent and a true distance between the LEDs was calculated and consequently, using those values, apparent and true distances between the camera taking the photo and the module with the LEDs were determined. Instead

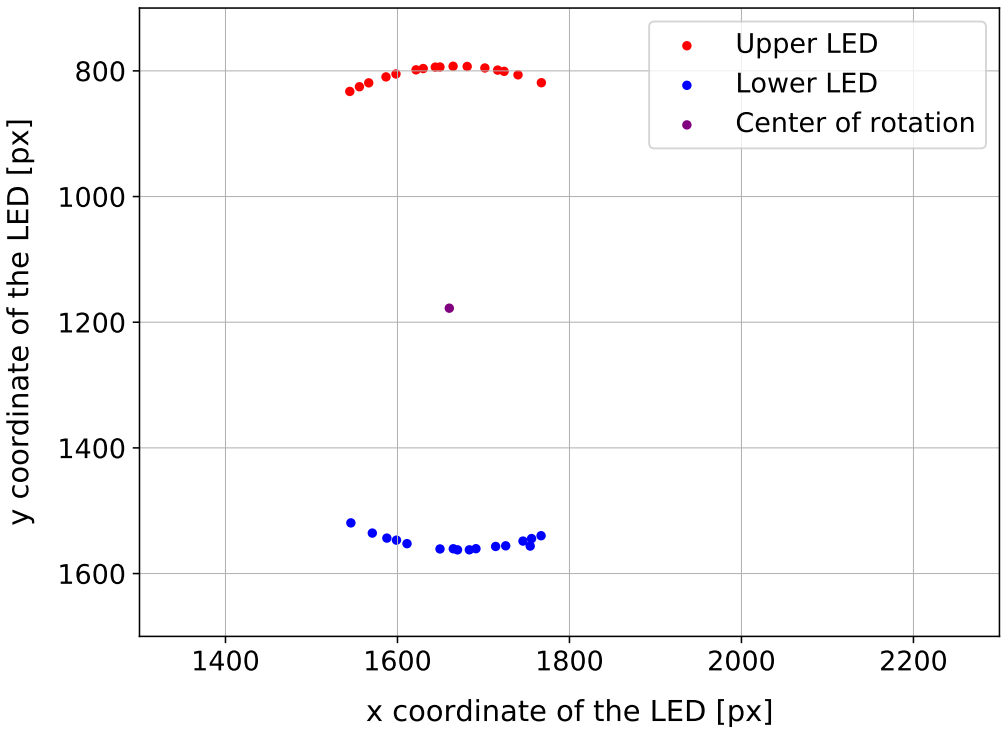


Figure 7.5: Results of the experiment regarding the camera module rotation. The plot shows the positions of LEDs at different inclinations.

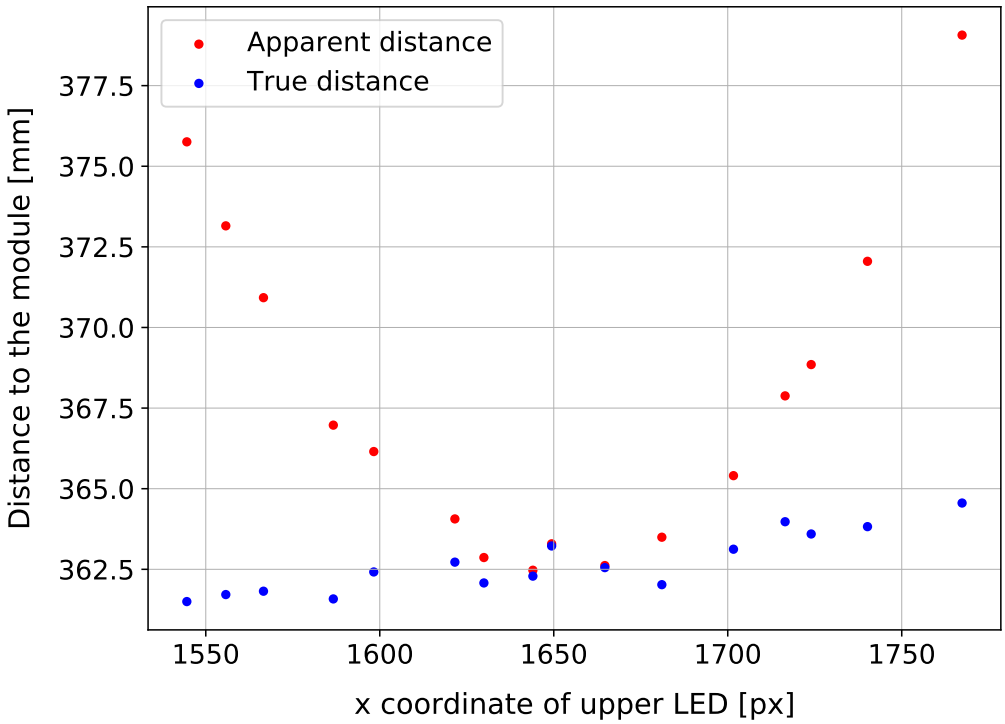


Figure 7.6: Results of the experiment regarding the camera module rotation. The plot depicts the proof of principle for the method developed in Section 6.2.

of plotting the distances against the inclination angle, a decision was made to place the x-coordinates of the upper LED on the abscissa. This procedure led to Fig. 7.6, where it can be seen that the correction algorithm works and manages to bring the apparent distances back to their true values. The curve that the red points describe has a correct sign - the more the module rotates, the smaller the apparent distance between the LEDs becomes, which makes one believe the module itself is further away than it really is. The reason for a slight slope of the line the blue points describe, and the general non-smoothness of points can be ascribed to the fact that the LEDs that were used have a certain depth to them. This makes them appear differently on the image as the module rotates, which consequently changes the center of gravity of the LED that is being calculated for each shot. Once the development of the system progresses, the LEDs will get much smaller and be firmly attached to a PCB, which should smooth out and level out the trends.

7.3 The complete system

Last but not least, the final part of the experiment will be discussed. Similarly to the rotation problem, it was conducted as a proof of principle - an indication that a continuation of the development of the camera alignment system could be a fruitful endeavour.

As was discussed in Section 6.3, at first, 3 pairs of LEDs were placed in the middle of the frame so as to simulate LEDs placed on the detector. The procedure for calculating the position of the LEDs and the shape of the tracker was as follows:

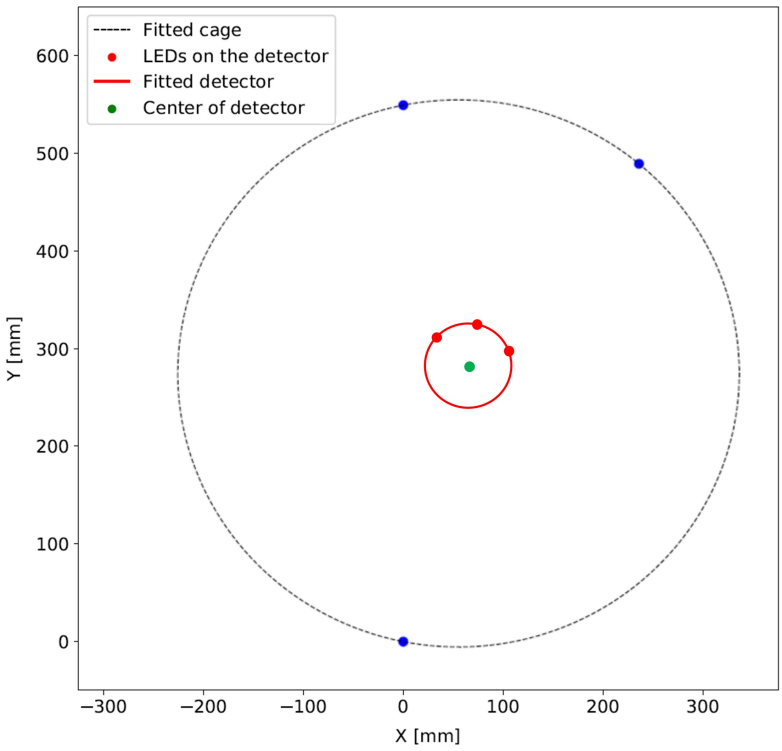


Figure 7.7: Results of the experiment with 3 pairs of LEDs standing in for the detector.

- Focal length of the cameras was determined as described in Section 4.2.
- Camera modules and LEDs were placed on the MakerBeam frame in a similar fashion as in Fig. 4.2.
- All of the variables needed for Eq. (4.3) and Eq. (4.4) were determined with the help of Eq. (4.1) and by arranging the camera modules as depicted in Fig. 4.3. The known variables were the distances between the LEDs on the camera module, and the distance between the LEDs arranged in pairs on the detector.
- Once the coordinates of the pairs of LEDs that represent the detector were determined, a fit for the detector cage and the tracker could be made using Least Squares Method - LSM in Python.

This led to a first confirmation that the ideas discussed in previous chapters could be realized. The resulting plot of the coordinates of different set-up elements, together with the fitted cage and detector, is shown in Fig. 7.7, and the live scenario can be seen in Fig. 7.8. It should be noted that the rate at which the measurements and calculations for a single arrangement of cameras and LEDs can be conducted predominantly depends on the rate at which the camera shots can be taken. If the procedure discussed in Section 4.2 is implemented, and each shot is taken 10 times, one can expect to have the coordinates of the tracker LEDs every $\sim 330\,\mathrm{s} = 5.5\,\mathrm{min}$.[4]

Since the step-by-step derivation of the error propagation for the points on the tracker was too cumbersome and error-prone, a more reasonable path towards the

[4] 3 s warm-up and 30 s cool-down, 10 times each

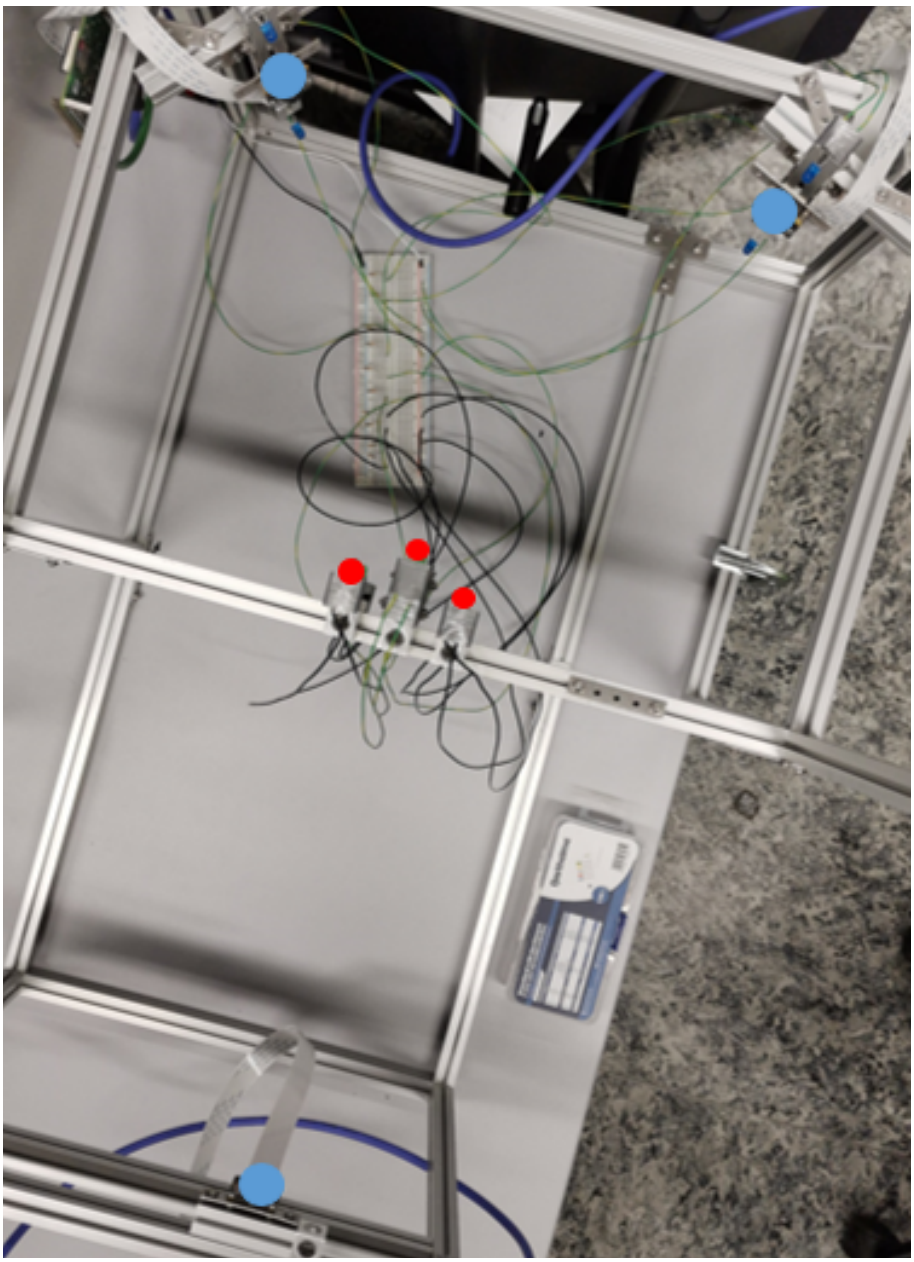


Figure 7.8: The experimental set-up with 3 pairs of LEDs attached to the middle aluminum frame (red dots) and three camera modules (blue dots).

Position	σ_x [µm]	σ_y [µm]
Leftmost	151	51
Middle	129	69
Rightmost	120	55

Table 7.1: Error values for the 3 points on the tracker (red) in Fig. 7.7.

solution was found by using WOLFRAM MATHEMATICA[5]. Standard deviations for different variables used in coordinate determination were calculated by propagating the uncertainty of variables that played a role in their calculation. This was followed by MATHEMATICA calculating the errors of the x- and the y-coordinate for all three points on the tracker. The error values for three points describing the detector are given in Table 7.1. These indicate that the reconstruction of the shape of the detector with an 80 µm precision (size of the MUPIX sensor) could be achieved with slight improvements to the set-up (discussed in Chapter 8). It should be noted that not only is there room for refinement in the design and execution of the experiment, but that the results that were achieved were done so with the first version of the frame, and with LEDs crudely attached to an aluminum profile instead of a PCB.

This however was not the end as it was noticed that 3 points are generally not enough to fit an ellipse to them, which is something that is needed if the elliptical deformations of the tracker are to be investigated. Consequently, two additional pairs of LEDs were attached to the central aluminium profile, which gave a total of 5 points for ellipse fitting. By once again reconstructing the scenario depicted in Fig. 4.3, the following plots (Fig. 7.9 and Fig. 7.10) were generated for two slightly different camera module positions, as well as a slight rearrangement of central pairs of LEDs. The ellipse fitting was once again achieved with the LSM.

The experiment at hand served a slightly different purpose than the previous one. It was not designed in the interest of estimating the errors but rather to capture disturbances and movements in the set-up. As it might not be obvious to the naked eye that the system managed to capture all of the changes and fluctuations between the two set-ups, an additional plot depicting their overlap was made (Fig. 7.11).

As a final note, it should be mentioned that the results presented in this section indicate a potential path forward with the camera alignment system. The resolution survey, as the most important part of the investigation, gives a clear indication that achieving a precision of the distance measurement of around 80 µm, which is the size of an individual MUPIX pixel, can be achieved. This however does not directly translate into the precision of determining the coordinates of the detector (see Table 7.1), which is why future improvements in the precision are needed. The summary of this work, along with an envisioned future course, is presented in the next and final section.

[5]For the purpose of this thesis, MATHEMATICA 9.0 by Wolfram Research, Inc. was used.

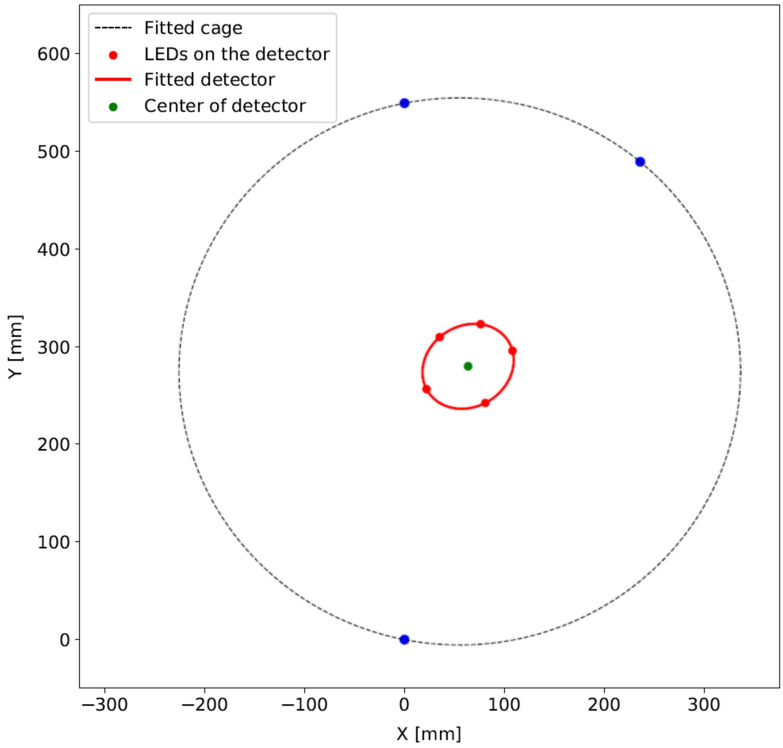


Figure 7.9: Results of the experiment with 5 pairs of LEDs standing in for the detector, allowing for the fitting of the ellipse, which helps in determining the elliptical deformation of the detector.

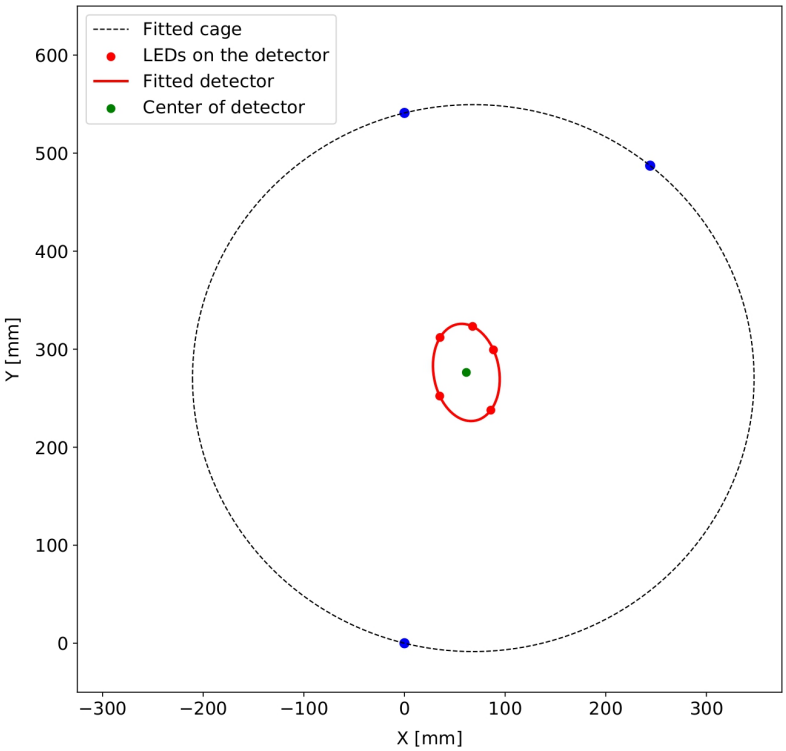


Figure 7.10: Results of the experiment with 5 pairs of LEDs standing in for the detector. All of the elements were slightly moved around compared to Fig. 7.9.

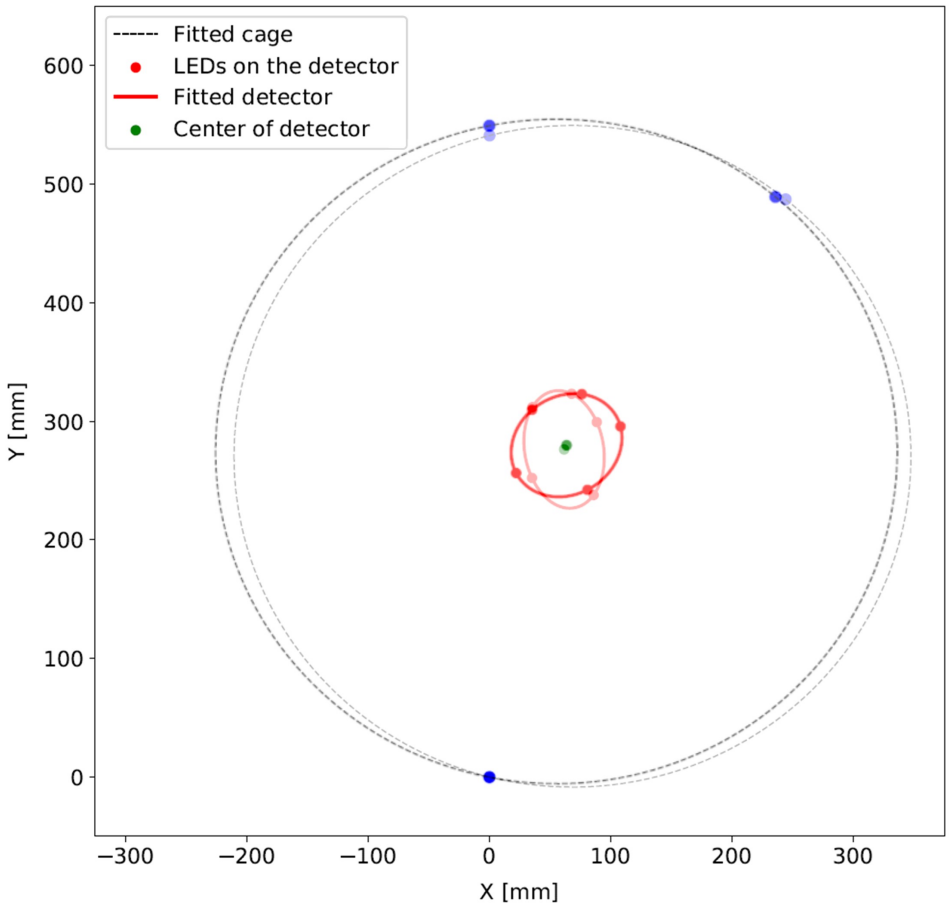


Figure 7.11: Overlap of the plots shown in Fig. 7.9 and Fig. 7.10.

Chapter 8

Conclusion and Outlook

Conclusion The Mu3e experiment is mainly a tracking detector and, as such, requires a high level of precision and accuracy, especially when it comes to spatial resolution. Reconstructing particle tracks together with their energies and momenta is of utmost importance when investigating physics beyond the Standard Model - the charged lepton flavor violation. This can, however, only be achieved with an excellent alignment strategy since numerous factors, which need to be corrected for, contribute to detector misalignment. A somewhat standard procedure, the track-based alignment, does not cover the so-called *weak modes*, so it needs to be amended with an external survey system. The development of a camera alignment system that aids in detector alignment and consequently in particle reconstruction was the main scope of this thesis.

It was established that the Raspberry Pi NoIR v2 cameras, together with infrared LEDs, can be used in a pitch-black environment to reconstruct the shape of the detector cage and the tracking detector. The resolution of the cameras is sufficient to achieve a precision of 80 µm (size of the individual MuPix sensor) at distances that are expected for Mu3e (Fig. 7.3). However, the combination of multiple results in order to infer the shape and position of the tracking detector requires further improvement (Table 7.1).

The problem of camera module rotation was also addressed, showing that the correction method developed in Section 6.2 can be implemented. This way, the errors that can crop up from the camera module's rotation around the struts of the detector cage can be alleviated.

Finally, a set-up that resembles the experiment's configuration has been developed by using the MakerBeam kit for the frame and the mock detector. It has revealed that the calculations discussed in Section 4.1 can, in fact, be applied and yield satisfactory results. The final section of the thesis, Section 7.3, has essentially confirmed the proof of concept developed in chapters prior to it.

Outlook As far as the future path of development for this project is concerned, several steps could lead to the improvement of results and stability of the system as a whole:

- The camera modules require the IR cameras and the LEDs to be attached to a single PCB. This way, the distance between the LEDs can be known to very high precision, and they themselves could be much smaller. This would then reduce the overall error of distance and position calculation.

- The cameras ought to be tested once again for radiation hardness in one of the upcoming beamtimes at the PSI; however, this time closer to the beam - preferentially inside the magnet.
- Communication between Raspberry Pis inside the detector and the outside world needs to be addressed. Ethernet connection has to be excluded because of the magnetic properties of the connector, which leaves optical fibers as a possible solution.
- A bigger version of the frame, closer to the size of the actual detector cage, ought to be constructed to reproduce the actual experimental conditions as faithfully as possible.
- Finally, a cylinder the radius and shape of the detector could be constructed, which would also improve the set-up's stability and reproducibility of the results. This would allow for the measurements that were conducted in Section 7.3 not only to be compared relative to each other but also to have the results compared to a predetermined and well-measured position of the LEDs on the cylinder, together with the position of the camera modules on the frame.

Appendix A

Error propagation

In the following, the derivation that led to the results depicted in Fig. 7.3 will be presented.

The process starts by choosing a reasonable distance between the two LEDs on a camera module in pixels, $d_{Iz} \in [300\,\text{px}, 1300\,\text{px}]$. We assume that the focal length of the camera f has an error of $\sigma_f = 0.14\,\mu\text{m}$, and that the true distance between the LEDs d_{real} has an error of $\sigma_{d_{real}} = 5\,\mu\text{m}$. This gives us all that we need to calculate the z-projection of the distance from the camera to the LEDs using Eq. (4.2)

$$L_z = f\left(\frac{d_{real}}{d_{Iz} \cdot m} + 1\right), \tag{A.1}$$

where m stands for a constant representing the size of an individual pixel on the camera sensor, which in this case amounts to $1.12\,\mu\text{m}$ (see Table 4.1 for reference). Next, the error of the z-projection is calculated as

$$\sigma_{L_z} = \sqrt{\left(\frac{\partial L_z}{\partial f} \cdot \sigma_f\right)^2 + \left(\frac{\partial L_z}{\partial d_{real}} \cdot \sigma_{d_{real}}\right)^2 + \left(\frac{\partial L_z}{\partial d_{Iz}} \cdot \sigma_{d_{Iz}}\right)^2}, \tag{A.2}$$

where the error of the distance between the LEDs was taken to be $\sigma = 0.033\,\text{px}$ (see Section 7.1.2). In this, as well as in all the other equations that contain partial derivatives, the DIFF function of the PYTHON package called SYMPY was used.

Coming after, the x- and the y-projection can be calculated. This is done by rearranging Eq. (4.2) to get

$$L_x = \frac{(L_z - f) \cdot m \cdot d_{Ix}}{f} \tag{A.3}$$

in case of the x-projection, and

$$L_y = \frac{(L_z - f) \cdot m \cdot d_{Iy}}{f} \tag{A.4}$$

in case of the y-projection. The same procedure as before followed for deriving the errors, which yielded

$$\sigma_{L_x} = \sqrt{\left(\frac{\partial L_x}{\partial f} \cdot \sigma_f\right)^2 + \left(\frac{\partial L_x}{\partial L_z} \cdot \sigma_{L_z}\right)^2 + \left(\frac{\partial L_x}{\partial d_{Ix}} \cdot \sigma_{d_{Ix}}\right)^2} \tag{A.5}$$

for the x-projection, and

$$\sigma_{L_y} = \sqrt{\left(\frac{\partial L_y}{\partial f} \cdot \sigma_f\right)^2 + \left(\frac{\partial L_y}{\partial L_z} \cdot \sigma_{L_z}\right)^2 + \left(\frac{\partial L_y}{\partial d_{Iy}} \cdot \sigma_{d_{Iy}}\right)^2} \tag{A.6}$$

for the y-projection. This time, the value chosen for the errors of the distance between the LEDs was $\sigma_x = \sigma_y = 0.022\,\text{px}$.

Finally, the distance from the camera to the LEDs can be calculated with

$$D = \sqrt{L_x^2 + L_y^2 + L_z^2}\,, \tag{A.7}$$

and the error can be derived by

$$\sigma_D = \sqrt{\left(\frac{\partial D}{\partial L_x} \cdot \sigma_{L_x}\right)^2 + \left(\frac{\partial D}{\partial L_y} \cdot \sigma_{L_y}\right)^2 + \left(\frac{\partial D}{\partial L_z} \cdot \sigma_{L_z}\right)^2} \tag{A.8}$$

where the values of errors are used from the formulas derived above. The error σ_D is the one that determines the color of a point in the plot in Fig. 7.3.

Bibliography

[1] Y. Fukuda et al., [Super-Kamiokande Collaboration], *"Evidence for oscillation of atmospheric neutrinos"*, Phys. Rev. Lett., **81** 1562–1567, 1998, (arXiv:hep-ex/9807003).

[2] Q. R. Ahmad et al., [SNO Collaboration], *"Measurement of the charged current interactions produced by B-8 solar neutrinos at the Sudbury Neutrino Observatory"*, Phys. Rev. Lett., **87** 071301, 2001, (arXiv:nucl-ex/0106015).

[3] K. Eguchi et al., [KamLAND Collaboration], *"First results from KamLAND: Evidence for reactor anti-neutrino disappearance"*, Phys. Rev. Lett., **90** 021802, 2003, (arXiv:hep-ex/0212021).

[4] Hernández-Tomé, G., López Castro, G. and Roig, P., *"Erratum to: Flavor violating leptonic decays of τ and μ leptons in the Standard Model with massive neutrinos"*, Eur. Phys. J. C, **80**(5) 438, 2020.

[5] Blackstone, Patrick, Fael, Matteo and Passemar, Emilie, *"$\tau \to \mu\mu\mu$ at rate of one out of 10^{14} tau decays?"*, Eur. Phys. J. C, **80**(6) 506, 2020.

[6] Mu3e Collaboration, *"Technical design of the phase I Mu3e experiment"*, NIM A, page 165679, 2021.

[7] Y. Okada, K. Okumura and Y. Shimizu, *"$\mu \to e\gamma$ and $\mu \to 3e$ processes with polarized muons and supersymmetric grand unified theories"*, Phys.Rev., **D61** 094001, 2000, (arXiv:hep-ph/9906446).

[8] M. Kakizaki, Y. Ogura and F. Shima, *"Lepton flavor violation in the triplet Higgs model"*, Phys.Lett., **B566** 210–216, 2003, (arXiv:hep-ph/0304254).

[9] J. Bernabeu, E. Nardi and D. Tommasini, *"$\mu - e$ conversion in nuclei and Z' physics"*, Nucl. Phys., **B409** 69–86, 1993, (arXiv:hep-ph/9306251).

[10] Bastian Diaz, Martin Schmaltz and Yi-Ming Zhong, *"The leptoquark hunter's guide: pair production"*, Journal of High Energy Physics, **2017**(10), Oct 2017.

[11] W. J. Marciano, T. Mori and J. M. Roney, *"Charged Lepton Flavor Violation Experiments"*, Ann.Rev.Nucl.Part.Sci., **58** 315–341, 2008.

[12] U. Hartenstein, *Track Based Alignment for the Mu3e Pixel Detector*, PhD thesis, Mainz University, 2019.

[13] V. Blobel, *"Software alignment for tracking detectors"*, Nucl. Instrum. Meth., **A566** 5–13, 2006.

[14] G. Cavoto, G. Chiarello, M. Hildebrandt, A. Hofer, K. Ieki, M. Meucci, S. Milana, V. Pettinacci, F. Renga and C. Voena, *"A photogrammetric method for target monitoring inside the MEG II detector"*, Review of Scientific Instruments, **92**(4) 043707, 2021, (arXiv:https://doi.org/10.1063/5.0034842).

Acknowledgments

I would like to thank my supervisor Prof. Dr. Niklaus Berger for the conversations we had, for the constructive advice he gave me, for the lightning-fast responses and proofreading, and above all, for the guidance through the project and all the ups and downs. My thanks also go to the whole Mu3e collaboration, especially the people in Mainz who were always ready to help. Next, I would like to thank my examiner Jun.-Prof. Dr. Florian Hug for his time and willingness to partake in my journey.

My studies would not be possible if it were not for the DAAD (Deutscher Akademischer Austauschdienst) who granted me a 2-year scholarship to study and live in Mainz, Germany. Therefore, my gratitude extends to all the people at the DAAD who work tirelessly to make non-German students feel at home.

Finally, I would like to thank my girlfriend Ajla Nurkanović for being with me through thick and thin, for all the creative suggestions, and for her inexhaustible support and love.