

Studies on an improved Bokeh Mirror Alignment for the Medium Sized Telescopes for CTA

Humboldt-Universität zu Berlin
Faculty of Mathematics and Natural Sciences
Institute of Physics



Bachelor Thesis

Submitted as part of the requirements for
obtaining the academic degree B.Sc. in Physics

Author:

Domenik Ehlert

May 31, 2018

Advisor:

Dr. Ullrich Schwanke

HU Berlin, Institute of Physics

Referees:

Prof. Dr. Thomas Lohse

HU Berlin, Institute of Physics

Prof. Dr. David Berge

HU Berlin, Institute of Physics / DESY Zeuthen

Selbstständigkeitserklärung

Ich versichere, dass die vorliegende Bachelorarbeit von mir selbstständig, unter Beachtung der Regeln der Studien- und Prüfungsordnung von 2014 des Monobachelor Physik, angefertigt wurde. Alle verwendeten Quellen, in unveränderter oder abgewandelter Form, wurden an gegebener Stelle gekennzeichnet. Diese Arbeit wurde für keine andere Prüfung eingereicht und ich bin mir bewusst, dass bei Verstoß gegen die Grundsätze der Selbstständigkeit ein Verfahren wegen Täuschung eingeleitet werden kann.

Berlin, May 31, 2018

The Medium-Sized Telescopes are part of the planned Cherenkov Telescope Array (CTA) and primarily intended to observe Cherenkov radiation produced by 0.1 TeV to 10 TeV[1] gamma- and cosmic-rays in the upper atmosphere. Their main mirror is composed of up to 90[2] smaller hexagonal mirrors making an alignment procedure necessary for proper observations. A promising technique for a rough pre-alignment is the Bokeh method which was first introduced at FACT[3] and already implemented on the MST prototype in Berlin, Germany[4, 5]. The Bokeh method requires the detection of patterns which can be done by different techniques. Hough transform and Convolutional Neural Network (CNN) based methods for pattern recognition are evaluated for use in conjunction with the Bokeh method and compared to the currently used technique of fitting a weighted circle. The primary evaluation is done via simulated images, intended to approximate the real CCD camera images. Results show an improvement in accuracy by a factor of about four on average for the optimized CNN-based approach, compared to the current standard method, while the technique using Hough transform is worse by a factor of approximately three. As the exact translation to the accuracy on real data is not known yet, this does not necessarily prove the CNN's superiority but should rather be understood as an indicator for possible improvements.

Contents

1	The CTA Project	1
2	The Medium-Sized Telescopes	3
2.1	Mechanical Structure	3
2.2	Optical System	4
2.2.1	Mirror Types	5
2.2.2	Active Mirror Control	5
3	Mirror Alignment	7
3.1	Overview of different Alignment Methods	7
3.2	The Bokeh Method	8
3.2.1	Overview of the Bokeh Method	8
3.2.2	Application at the MST prototype in Berlin	9
4	Studies on improved Pattern Recognition for the Bokeh Method	10
4.1	Current Implementation and Motivation for further Studies	10
4.2	Hough-Transform	12
4.2.1	General Concept	12
4.2.2	Image Filtering	13
4.2.3	Canny Edge Detector	14
4.2.4	Implementation of the Hough-Transform	14
4.3	Convolutional Neural Networks	18
4.3.1	Introduction	18
4.3.2	General Concept	18
4.3.3	CNN Implementation	23
4.4	Comparison of the different methods	25
4.4.1	Simulated Data	25
4.4.2	Application to Real Observation Data	33
5	Conclusion	35

1 The CTA Project

The Cherenkov Telescope Array (CTA) is the latest large scale project in the field of ground-based very-high-energy (VHE) gamma-ray astronomy. CTA is developed by an international consortium of scientists from 210 institutions in 32 countries[1] and planned as the successor to current generation Imaging Atmospheric Cherenkov Telescope (IACT) arrays like H.E.S.S.[6], MAGIC[7] or VERITAS[8] which were already very successful in their own right and played a major role in opening up the area of VHE gamma-ray astrophysics.

In general there are three possible detector designs for detecting VHE gamma-rays. For one there are satellite-based gamma-ray observatories like COMPTEL[9], INTEGRAL[10], AGILE[11] and FERMI[12] which enable direct detection of incident gamma photons. However since the gamma-ray flux of typical astrophysical sources scales like $dN/dE \approx 1/E^2$ the number of incident gamma photons quickly drops for larger energies in the order of several 10 GeV[13]. Therefore observation of VHE photons requires detection areas significantly larger than the typical 1 m^2 of satellite-borne telescopes, making them unsuitable for the observation of VHE gamma sources[13]. Nonetheless they had, and still have, great success in observing the keV – GeV energy range and therefore it is the intention for CTA to work in conjunction with space-based telescopes[1].

Moreover there are also ground based gamma-ray observatories which can in turn be divided into particle detectors that directly observe the resulting particles of air showers upon reaching the ground and atmospheric Cherenkov telescopes, like the ones CTA will contain. Both methods use the fact that VHE gamma-rays impinging upon the atmosphere will interact with the nuclei of air constituents producing an electron-positron pair in the course due to pair-production. These will in turn produce atmospheric showers of further electrons, positrons and photons of lower energy by means of bremsstrahlung and pair-production and therefore generate an electromagnetic air shower. Normally the shower has a length of some kilometers and width of a few hundred meters. The resulting charged particles, traveling at superluminal speeds (relative to c_{air}), will generate a Cherenkov light cone with a footprint of around 120 m radius and duration of a few nanoseconds[13]. Representatives for the group of air shower detectors are e.g. Milagro[14], Tibet AS-gamma[15], ARGO-YBJ[16] or the recently built HAWK[17]. Unfortunately it turns out that, even with the advantage of a 24 h duty cycle and long observation times, their sensitivity is only sufficient for detection of steady gamma-ray sources with intensities similar to the Crab Nebula, the common standard candle in gamma-ray astrophysics[18].

For the mentioned reasons ground-based (imaging) Cherenkov telescope arrays are the best solution for observations of VHE gamma-rays (100 GeV and above) since they offer a significantly improved sensitivity compared to air shower detectors, on the order of 10 mCrab for 25 h observation time and major improvements in angular and energy resolution[18]. Moreover, due to deployment of many telescopes as part of large arrays, IACT arrays can easily achieve collector areas of several hundreds of square meters, and the effective detection area is even larger. Provided enough photons reach the real detector, a telescope located within the Cherenkov light pool will actually observe the air shower and thus its effective detection area is approximately given by the light cone's footprint of around 10^5 m^2 , making it vastly superior for the detection of TeV gamma-rays compared to space-based observatories[13]. The telescopes themselves usually consist of a large collection mirror made out of many smaller mirror facets that reflect the incident Cherenkov light into a camera installed in the focal plane.

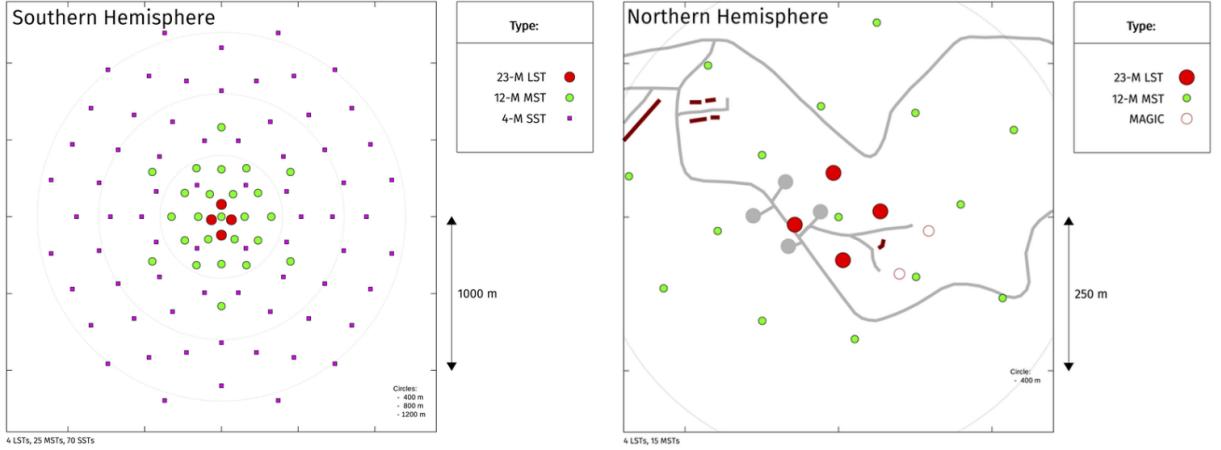


Figure 1.1: Possible layouts for the two telescope array locations. CTA will consist of three different telescope sizes for observations at different energies. The lower energies up to a few 100 GeV will be primarily covered by the Large-Sized Telescopes (LSTs) with a main mirror of 23 m diameter, the range 0.1 TeV to 10 TeV by the 12 m diameter Medium-Sized Telescopes (MSTs) and the highest energies at 10 TeV and above by the Small-Sized Telescopes (SSTs) with a 4 m diameter primary mirror. Left: The southern site (Chile) contains all three telescope sizes to cover the full observation range \sim 20 GeV to \sim 300 TeV. Right: Current plans for the northern site (La Palma) do not include any SSTs, since these high energies are mainly interesting for the study of cosmic accelerators in our galaxy, which will be done by the southern array. [13]

CTA aims to improve the sensitivity of current IACT arrays by about an order of magnitude in the central energy region from 100 GeV to several 10 TeV and to expand the available range to energies above 300 TeV and substantially below 100 GeV. It is expected that this enhancement will enable the detection of over 1000 sources of high energy gamma-rays[18].

To achieve these goals CTA will rely on proven technologies that were already extensively tested in the aforementioned previous IACT arrays but will significantly increase the number of deployed telescopes to obtain an effective collection area on the order of 10 km^2 [18]. This expanded area will also improve observations by increasing the number of events whose Cherenkov footprint is completely contained within the array, an improved stereoscopic view benefiting shower reconstruction and background rejection, and a lower energy threshold since the brighter central region of the shower should usually be contained within the array[1]. To enable a full sky coverage the telescopes will be deployed on two different sites, one in each hemisphere (South: Chile, North: La Palma; Fig. 1.1). For reasons given in the next paragraph the southern array will contain ninety-nine telescopes while the northern site will only consist of nineteen[13].

The most cost-effective approach for reaching the desired large energy range is the deployment of telescopes with different sizes (Fig. 1.2). The lowest energies up to some 100 GeV will be covered by 4 Large-Sized Telescopes (LSTs) per site, with mirror diameters of around 23 m[13]. Observations in the central energy region from 0.1 TeV to 10 TeV will be done by the Medium-Sized Telescopes (MSTs) with mirrors of approximately 12 m in diameter and a fairly large field-of-view (FoV) of 8° . The northern array will contain 15 MSTs while the southern will comprise 25 of them[13]. Finally Cherenkov light of gamma-rays with energies above 10 TeV up to around 300 TeV[19] will be detected by Small-Sized Telescopes (SSTs). Currently their design is not yet finalised and there exist 2 prototypes SST-2M ASTRI, SST-2M GCT based on the dual reflector Schwarzschild-

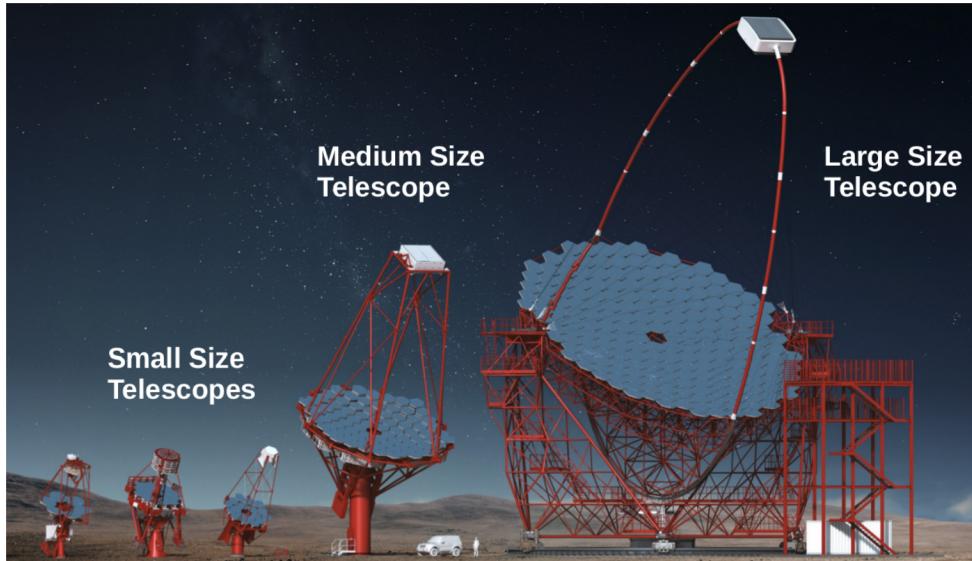


Figure 1.2: Different telescopes developed for the CTA project. From left to right: SST-2M GCT and SST-2M ASTRI based on Schwarzschild-Couder optics, the Davis-Cotton based SST-1M, all three with a primary mirror of 4 m diameter. Next an MST with Davis-Cotton design and 12 m dish diameter and on the right side an LST design with a parabolic dish and 23 m diameter main mirror. [13]

Couder optics and one with a single reflector Davis-Cotton set-up (SST-1M). They all offer a main mirror diameter of 4 m and, like the MSTs, a large FoV of around 9° to enable good performance for surveys and extended galactic sources[13]. Since these high energies are most important for galactic sources the SSTs will only be incorporated into the southern array (70 SSTs)[1].

2 The Medium-Sized Telescopes

The Medium-Sized Telescopes¹ are intended as the primary tool for observations in the central energy range from 0.1 TeV to 10 TeV within the CTA framework. They employ a single reflector modified Davies-Cotton (DC) design to obtain a good temporal resolution and push the time spread below an RMS (Root Mean Square) of 0.8 ns, sacrificing some angular resolution in the process compared to a pure DC design. Davies-Cotton optics allows good imaging over relatively wide FoVs (field of view), but introduces a significant time dispersion correlating with the reflector's diameter[20].

In 2012 a prototype of the modified Davies-Cotton MST design was built in Berlin to test the different components and interfaces between them, and to devise a construction procedure. Furthermore, the prototype is also used to develop various necessary software applications, e.g. for pointing and mirror alignment.

2.1 Mechanical Structure

An overview of the mechanical structure of the MSTs is presented in Fig. 2.1. The optical support structure (OSS) holding the up to ninety mirror facets is composed of a central

¹Unless indicated otherwise most information in this section is reproduced from the *Medium-Sized Telescope Technical Design Report*[2].

plate and twelve arms extending radially, made of 6 m long rectangular tubes interconnected with rib-like circular tubes, providing the necessary strength and stability. The arms are bent to a curvature radius of 19.4 m achieving the desired reflector curvature of 19.2 m together with the mirror support structure.

The camera support structure (CSS) is connected to the OSS at six points on the dish's outer perimeter, extending toward the curvatures center to hold the camera in the focal plane at a distance of 16 m. To achieve the required stiffness the camera is held via four sturdier beams in four of the points and additional thinner supporting tubes on the sides. The torque resulting from the 2 t camera and additional mass of the CSS is compensated by a counterweight installed on the opposite side of the elevation axis. To minimize stress on the dish resulting from this setup the counterweight is fixed to the OSS at the same points as the CSS.

All of the dish and supporting structure is held by a hollow cylindrical positioning tower of 1.8 m diameter. The interior houses three floors, which hold some of the electrical equipment used for controlling the telescope, and can be accessed by a door on the ground level as well as two hatches on the top of the tower. An air conditioning system is installed to maintain a stable climate enabling maintenance even during hot summer days and reducing the likelihood of equipment failure.

The top of the tower also functions as azimuth bearing containing the slewing drive for horizontal rotation of the telescope dish. The drive allows for an operational range of $\pm 270^\circ$ on the azimuthal axis and satisfies the CTA requirement to reach any pointing position above 30° elevation within 90 s. Above the azimuth bearing two yokes extend outwards from the sides of the tower in opposite directions, connecting it with the elevation drive system responsible for the vertical orientation of the dish. Each of the two sides represents a separate subassembly respectively equipped with two motors, like the azimuthal drive, providing redundancy in case of failure and improved slewing speeds. The elevation drive provides an operational range of 111° satisfying the CTA condition of observation capabilities between 25° and 91° on the elevation axis and allowing for a parking position at -20° . Both drive systems employ absolute position encoders with an accuracy of $\pm 20''$ and are connected to a central lubrication system.

2.2 Optical System

One of the most important parameters of a reflector is its Point Spread Function (PSF). Within the CTA context the PSF is defined as the diameter of a circle in the reflector's focal plane that contains 80 % of the reflected light of a point-like source. In general the smaller the PSF the better a reflector's performance, however, there are no real gains when improving the individual reflector's PSF below the size of one third of a pixel of the main camera for on-axis observations. From the Cherenkov camera's pixel size of $0.18''$ this leads to a required PSF for the mirrors of $<4.8'$ which corresponds to <22.3 mm in the focal plane, for mirror's with focal length 16 m. Since most shower images are recorded off-axis by the IACTs the PSF of the mirrors should be smaller than one camera pixel within 80 % of the camera's FoV.

The optical support structure is able to accommodate up to 90 hexagonal mirrors with a flat-to-flat distance of 1.2 m, but the adverse impact on the overall PSF from the most distant mirrors is too strong and therefore presumably only 86 mirrors will be installed in the end.

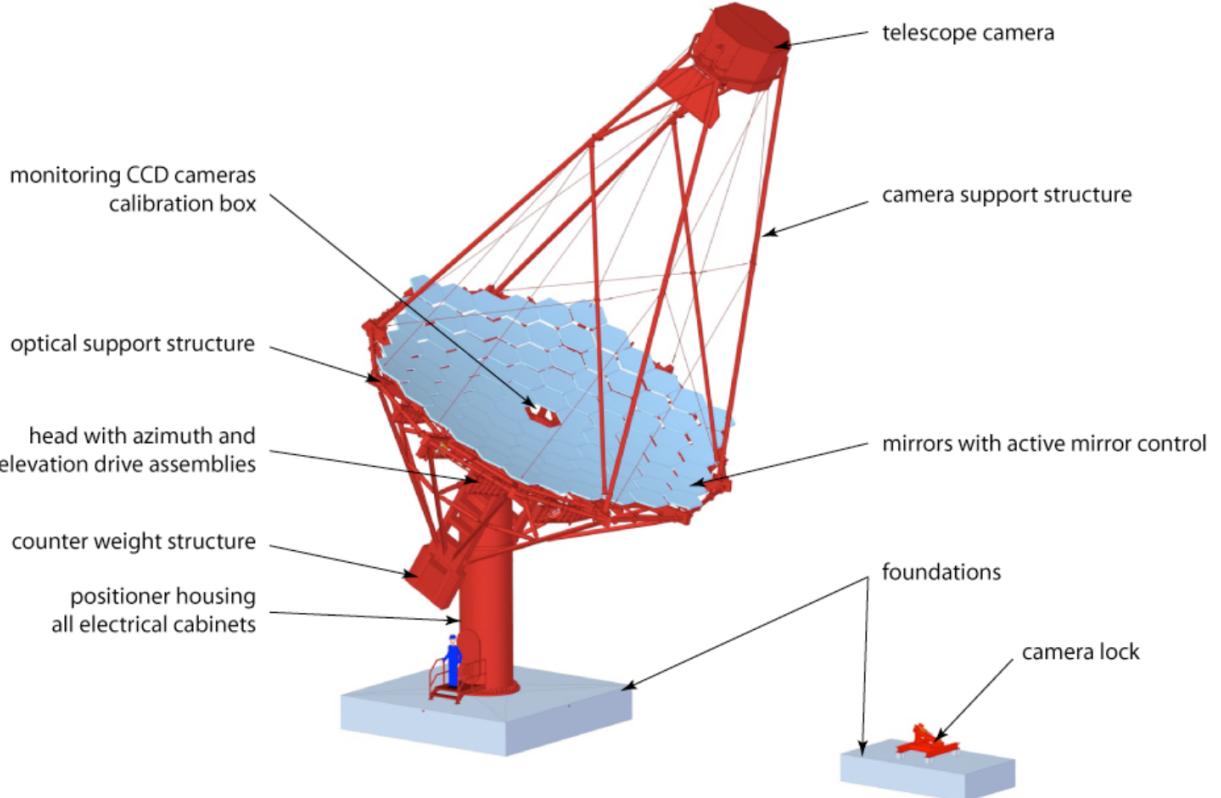


Figure 2.1: Schematic overview of the MST prototype's mechanical structure and main components. [2]

2.2.1 Mirror Types

Due to various problems with the mirrors of previous IACTs² the MST mirrors are created via a new process based on the cold slumping technique. Essentially a thin glass sheet is bent by vacuum suction on a precisely shaped mould giving it the necessary curvature. Afterwards additional layers are added to build the mirror panel's sandwich structure. After the glue has hardened the panel can be released and the mirror is coated with a reflective layer of aluminum and a layer of SiO_2 (or several layers) for protection. Currently there are 3 institutes involved in producing the mirrors each of which builds slightly different mirrors, although all of them use the same general process outlined above. The main characteristics of each version are described in Tab. 2.1. An outline of the production process for the INAF (Italy) mirrors is displayed in Fig. 2.2. All mirror types are designed to provide the same interface at the back and therefore allow for easy interchangeability. Measurements done in 2015 and 2016 at DESY Zeuthen determined the different mirror's PSF as $d_{80} = (11.6 \pm 2.2)$ mm for INAF, $d_{80} = (23.5 \pm 2.4)$ mm for CEA and $d_{80} = (17.1 \pm 2.1)$ mm for the IFJ-PAN mirrors[5]. The mirrors produced by INAF and IFJ-PAN therefore already satisfy the requirement of a PSF less than 22.3 mm.

2.2.2 Active Mirror Control

Although a re-adjustment of mirrors during normal operation is not necessary, the possibility for remote control of mirror positions is highly advantageous for initial alignment

²described in more detail in [2]

Institute	Description	Coating	Weight Width
INAF, Italy	Two glass sheets, Al honeycomb layer between them, plastic sidewalls	Al+SiO ₂	16 kg 25 mm
CEA, France	Two glass sheets, Al honeycomb layer between them, glass sidewalls	Al+SiO ₂ +HfO ₂ +SiO ₂	19 kg 40 mm
IFJ-PAN, Poland	Two glass sheets, Al tube spacers (sides open), coated before slumping	Al+SiO ₂ +HfO ₂ +SiO ₂	33 kg 60 mm

Table 2.1: Overview of the different MST mirror types. [2]

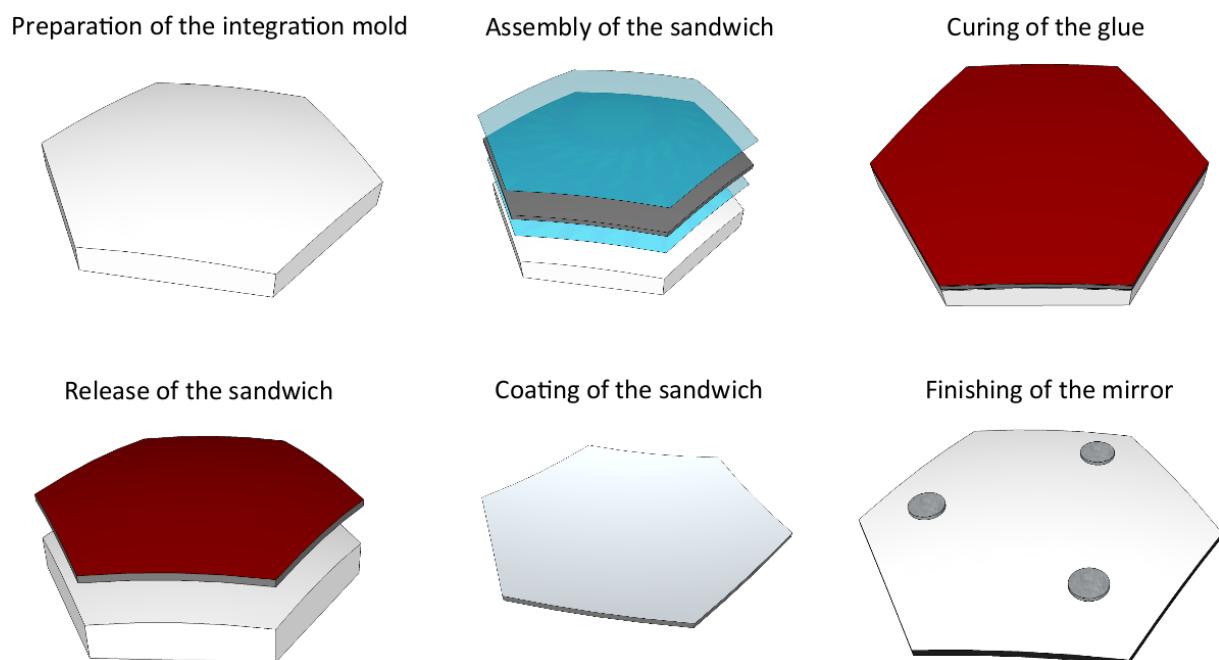


Figure 2.2: General outline of the production process for INAF mirrors. [2]

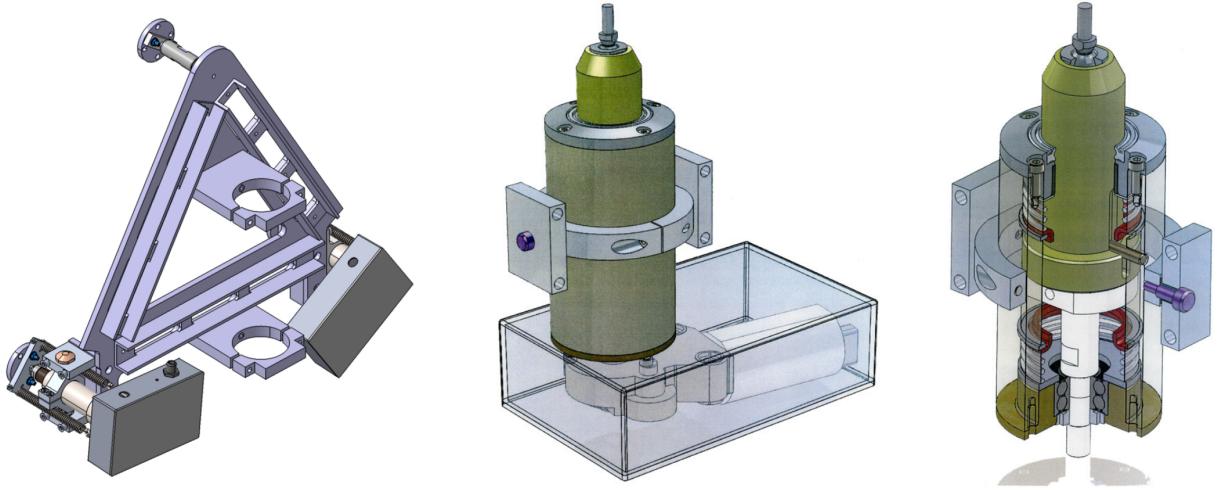


Figure 2.3: Left: Triangle element for mirror - OSS interface, Right: General schematic of the actuators. [2]

as well as re-orientation after replacement of a mirror and for a fast defocussing in case of emergency. Therefore the MSTs will be equipped with an Active Mirror Control (AMC) system. Every individual mirror is connected to the OSS at the three corners of an equilateral triangle support element attached to the back of the mirror (Fig. 2.3). Two of the connection points are equipped with custom, motor powered linear drives (actuators), that enable a remote tuning of the mirror's orientation. Each actuator is outfitted with a separate micro controller that connects to the central control and monitoring software via wifi, allowing for individual setting and reading of the actuator values. It is envisioned to automate the alignment process to a large extend.

3 Mirror Alignment

3.1 Overview of different Alignment Methods

As the MST's main mirror is made up from around 80 smaller mirror facets a high accuracy mirror alignment procedure becomes necessary. Various alignment methods were already developed for other IACTs with the most important of them being Laser-based 2f-alignment, which suffers from geometric restrictions and requires additional hardware, alignment via imaging of a sufficiently bright star on a reflective screen in the focal plane with the help of a dedicated CCD camera and alignment based on the SCCAN method where a star is tracked with the telescope and the reflections of the individual mirrors are observed with a camera installed in the telescope's focal plane[3]. The SCCAN method is complex to implement and also requires a bright star. The second technique, based on imaging a star, is the one that will most likely be used for final alignment of the MSTs at the CTA sites, however, there exist a couple of limitations. For one it requires a bright star which limits operation to clear nights without much background light pollution. It is also limited by the size of the actual screen. As the main camera is already located in the focal point of the telescope, no additional screen can be installed for the alignment process and therefore the closed lid of the camera has to be used as a screen. However, because the reflectivity of the lid is not sufficiently high, a spectralon target screen with a size of around 30 cm × 30 cm was placed on the lid at the MST prototype (Fig. 3.1, Left). The alignment process works by calculating the difference between two images taken before

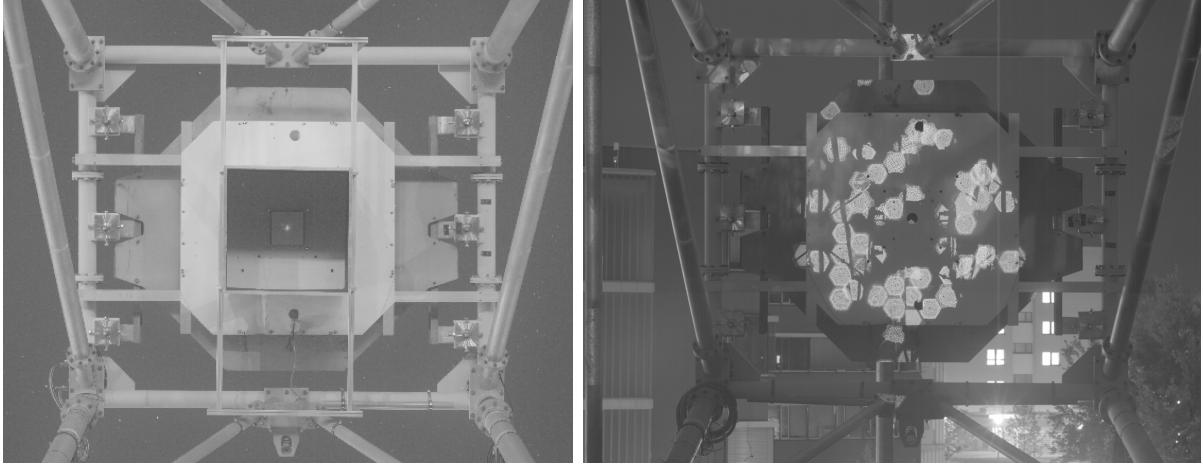


Figure 3.1: Left: Image of the main Cherenkov camera and part of the support structure taken by the SingleCCD camera on the MST prototype in Berlin while tracking the star Arcturus. Because of substantial background light from nearby streetlights and buildings the spectralon target in the center is surrounded by a baffle in this setup. The intensity is scaled logarithmically. Right: Image made by the SingleCCD camera during Bokeh alignment. The area of the spectralon target is smaller by a factor of twenty compared to the full Cherenkov camera lid, decreasing the available target area for the star-based mirror alignment process. [19]

and after changing the value of one of the actuators (see Sec. 3.2.2) thus it requires the mirror to already be sufficiently aligned so that both times the star's reflection falls on the target screen. For these reasons the Bokeh method was proposed as a rough automated pre-alignment method[4], with the advantage of not requiring a bright star, clear weather or even total darkness. In addition the whole surface of the camera's cover can be used as a screen (Fig. 3.1, Right), increasing the target area by a factor of about 20. The goal is a PSF of at most 0.5° , i.e. 140 mm in the focal plane[19], which can then be further improved by the star-based alignment.

3.2 The Bokeh Method

3.2.1 Overview of the Bokeh Method

The central idea of the Bokeh method is the out-of-focus imaging of a point-like light source that is placed on the optical axis at a distance $g > 2f$. Assuming a thin reflector for simplicity, the thin lens equation gives

$$\frac{1}{f} = \frac{1}{b} + \frac{1}{g} \quad (1)$$

where f is the focal length and b the distance where a sharp image can be observed[3] (Fig. 3.2). However, when observing at a distance $d \neq b$, the result is a de-focused reflection of each mirror segment in a hexagonal form, called its Bokeh function $B(r, \phi)$. (r, ϕ) are the coordinates of a polar coordinate system in the screen's plane centered on the point where the point-like image of the star would be, if the screen were at the distance b . For thin optics it is linearly dependent on the aperture function $A(r, \phi)$ [3] with a scaling factor r_B/r_A of

$$\frac{r_B}{r_A} = 1 - d \cdot \left(\frac{1}{f} - \frac{1}{g} \right). \quad (2)$$

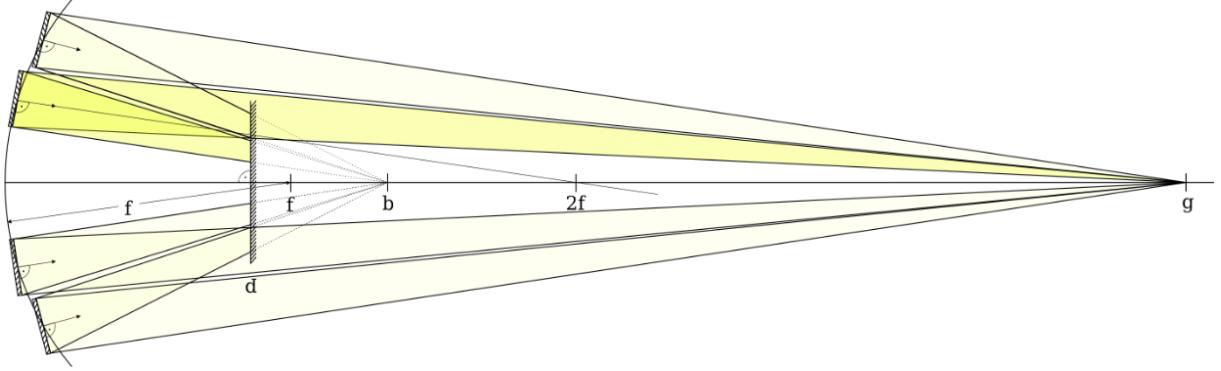


Figure 3.2: Schematic of a setup for Bokeh imaging. Light from the source at g is reflected by the mirrors and focused in the point b . The positioning of the screen at distance $d < b$ leads to a de-focused image, called the Bokeh. [3]

3.2.2 Application at the MST prototype in Berlin

A dedicated lamp, positioned at a distance of (104.7 ± 0.3) m[5] on the roof of a nearby building, is used as a light source for the Bokeh alignment. As described in section 2.2.2 the individual mirror facets are mounted on triangular supports with one fixed point and two mounts that are movable in the z-direction via actuators. This allows for remote changes of the mirror's orientation and in extension the position of the Bokeh pattern on the screen (the closed lid of the main camera). For the telescope pointing calibration a CCD camera ("SingleCCD", Tab. 3.1) was installed in the dish's center, facing towards the telescope's main camera. This CCD camera can also be used for the Bokeh-based mirror alignment by taking a picture of the Bokeh pattern, then changing the values of the actuators of a single mirror and taking another picture. Afterwards the difference between both images is calculated, ideally canceling out most background and unchanged mirrors, leaving only the Bokeh pattern of the adjusted mirror before and after the change, as illustrated in Fig. 3.3. In general the x- and y-position of the mirror's Bokeh hexagon are complicated functions of both actuators values a_0 and a_1 , which can be simplified using a linear approximation

$$\begin{aligned} x = f_1(a_0, a_1) &\approx f_1(\tilde{a}_0, \tilde{a}_1) + \frac{\partial f_1}{\partial a_0}\Big|_{\tilde{a}_0} (a_0 - \tilde{a}_0) + \frac{\partial f_1}{\partial a_1}\Big|_{\tilde{a}_1} (a_1 - \tilde{a}_1) \\ y = f_2(a_0, a_1) &\approx f_2(\tilde{a}_0, \tilde{a}_1) + \frac{\partial f_2}{\partial a_0}\Big|_{\tilde{a}_0} (a_0 - \tilde{a}_0) + \frac{\partial f_2}{\partial a_1}\Big|_{\tilde{a}_1} (a_1 - \tilde{a}_1) \end{aligned} \quad (3)$$

where $(\tilde{a}_0, \tilde{a}_1)$ are the actuators initial positions (Fig. 3.3, first image). By defining $f_1(\tilde{a}_0, \tilde{a}_1) = x_0$ and $f_2(\tilde{a}_0, \tilde{a}_1) = y_0$ Eq. 3 can be rewritten in a convenient matrix form as

$$\begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \approx T \cdot \begin{pmatrix} a_0 - \tilde{a}_0 \\ a_1 - \tilde{a}_1 \end{pmatrix}, \quad \text{with } T = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}, \quad t_{ij} = \frac{\partial f_i}{\partial a_j}\Big|_{\tilde{a}}. \quad (4)$$

By determining the movement $\Delta x = x - x_0$ and $\Delta y = y - y_0$ of the hexagon on the screen for a given change in actuator values $(\Delta a_0, \Delta a_1)$, it is possible to assign a response matrix T to every mirror that translates between actuator and Bokeh movement. The movement $(\Delta x, \Delta y)$ can be calculated by identifying the difference between the hexagon centers (x_0, y_0) and (x, y) before and after change in actuator value from the images taken by the CCD camera, which will be discussed in the next sections.

CCD camera	pixel width [μm]	pixel scale ω [arcsec]	pixels	image depth	FoV
SingleCCD	5.5	21.7	3296 \times 2472	16 bit	20.5° \times 15.5°

Table 3.1: Properties of the CCD camera (Aspen Apogee) used for mirror alignment on the MST prototype. (Source: U. Schwanke, private communication)

Since the ideal position for every mirror's Bokeh pattern can be theoretically calculated from models, the matrix' inverse T^{-1} can be used to determine the optimum actuator values \vec{a} for every mirror segment with

$$\vec{a} = \vec{a}_0 + T^{-1} \begin{pmatrix} x_0 - f_1(\vec{a}) \\ y_0 - f_2(\vec{a}) \end{pmatrix}. \quad (5)$$

Due to uncertainties in the alignment, introduced by the limited accuracy in determining the movement and by the hardware, this process needs to be executed iteratively for every mirror facet until the deviation between center of the mirror facet's reflection and the ideal position is below the required threshold. A Bokeh pattern template for the MST was produced using a Monte Carlo ray-tracing simulation[21].

4 Studies on improved Pattern Recognition for the Bokeh Method

Two new methods, based on Hough transform and Convolutional Neural Networks respectively, for finding the centers of the hexagons produced by the de-focused imaging of the mirror facets were implemented and compared to the currently employed technique, where a weighted circle is fitted to the data. A solution that simply calculates the center of gravity of all pixels was used for comparison as well. The real point of interest is the performance of the different methods on the difference images based on images taken by the SingleCCD camera installed on the MST prototype in Berlin. As the true centers of the hexagons in these images are unknown the comparison was primarily done by means of simulated images were the true hexagon center is known and therefore the accuracies of the individual methods can be evaluated. Afterwards the techniques were also applied to 54 images from the SingleCCD and the differences between their results were analysed.

4.1 Current Implementation and Motivation for further Studies

The currently employed method for finding the hexagon centers is based on fitting a weighted circle with fixed radius R to the Bokeh pattern[19], in the following called WCF method (Weighted Circle Fitting). The algorithm tries to maximize the quantity

$$A(\vec{x}) = \frac{\sum_k d_k^2 p_k}{\sum_k d_k^2} \quad (6)$$

where the sum goes over all pixels k with a distance d_k from \vec{x} less than R and magnitude p_k after image cleaning. The center of the circle with the highest value for A will roughly coincide with the center of the Bokeh hexagon pattern. The pixels are weighted with the squared distance d_k^2 from \vec{x} , meaning more distant pixel have a much stronger effect

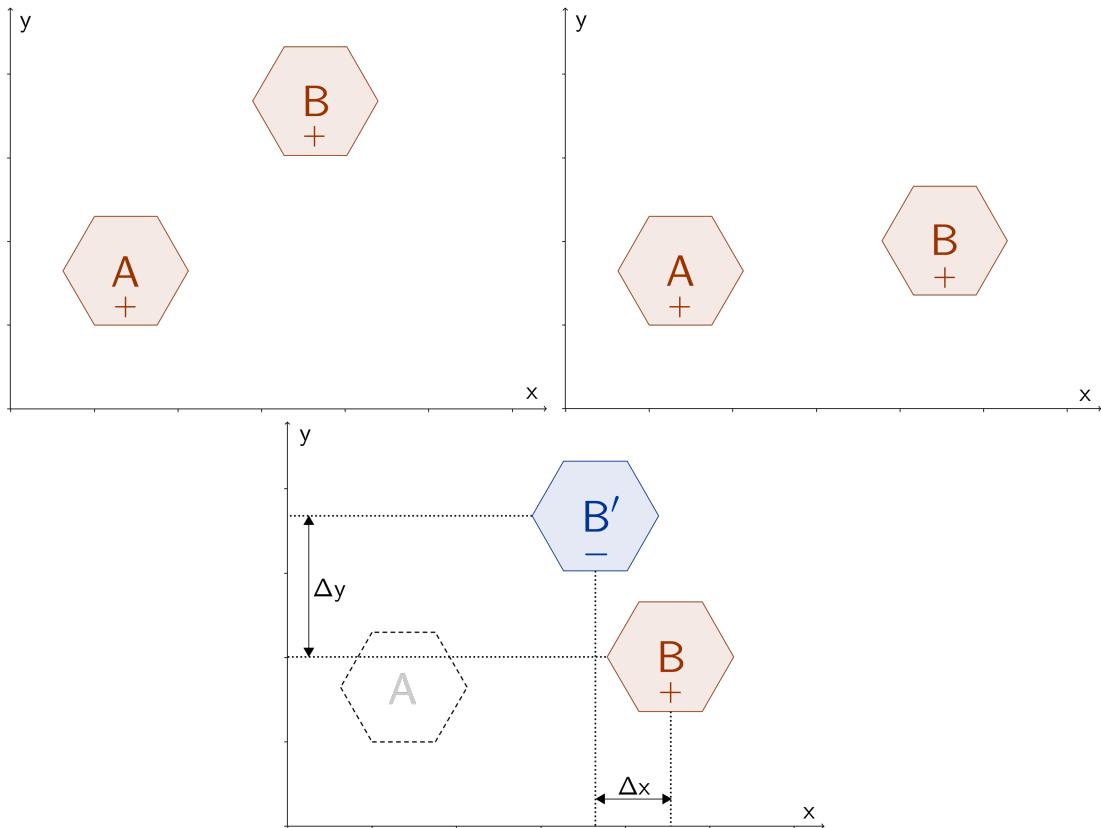


Figure 3.3: Procedure for Bokeh-based mirror alignment. First an image of the screen on the closed main camera's lid is taken, containing the Bokeh patterns of multiple mirrors (Top Left). Next the actuator-values of only one of the mirrors are changed, moving the mirror's Bokeh reflection on the screen as a result (Top Right). Finally the difference between both images is taken, largely canceling out the reflections of other mirrors and constant background light (Bottom). The result is a difference image containing both a positive and negative Bokeh hexagon of the same mirror (see Fig. 4.3, Top). From the difference in the centers between both hexagons and the known change in actuator values the response matrix T can be calculated.

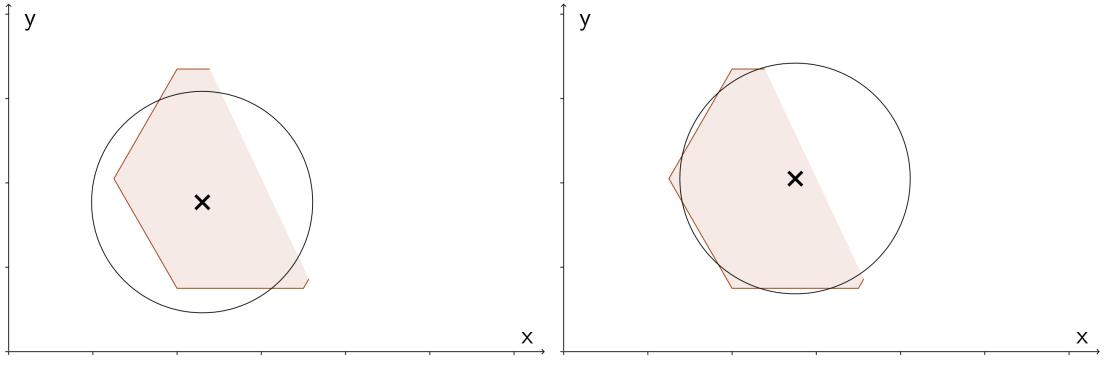


Figure 4.1: (Schematic) Effect of partial shadowing of the Bokeh hexagon on the Weighted Circle Fitting (WCF) method. Left: All pixels are weighted the same; a missing part of the hexagon substantially alters the center found by the algorithm. Right: More distant pixels are weighted stronger; therefore the effect of shadowing is decreased.

on the fitting algorithm. This is necessary because the Bokeh patterns in general do not follow perfect hexagonal shapes but can be substantially shadowed by the camera support structure, cutting out parts of the reflection. If all pixels were weighted equally then this would lead to a shift of the fitted circle's center away from the pattern's cut-off side (Fig. 4.1).

However, the modified weights only reduce the effect of shadowed regions and do not completely remove it. In addition, there are other problems with the pattern quality, like a rather low contrast between Bokeh pattern and background as well as relatively diffuse edges of the hexagons. The algorithm also compresses $8 \text{ px} \times 8 \text{ px}$ kernels into one pixel for processing, further reducing the possible accuracy. Therefore, additional methods for extracting the reflection's center will be detailed and investigated in the following sections.

4.2 Hough-Transform

4.2.1 General Concept

The general idea underlying Hough transform is the representation of shapes in images by a parametric description, e.g. as lines or curves. More specifically in this case we are interested in the boundaries of the Bokeh hexagons where every edge can be parametrized by a linear equation of the form

$$x \cos(\theta) + y \sin(\theta) = r \quad (7)$$

with the parameters (r, θ) being constant for all points (x, y) along the same edge[22] (see Fig. 4.2). This form is chosen over the more common linear representation of $ax + b = y$ since this would lead to infinite values of the parameter a for vertical edges. Every line in 2D space, and therefore every possible hexagon edge, can be described completely by the two arguments (r, θ) . These parameters are now determined by quantization into discrete bins in a sort of voting process that constitutes the actual transformation. Every non-zero pixel (x, y) in space is transformed into all possible values (r, θ) for which eq. 7 is fulfilled, essentially voting for all lines that go through this point in space. Each vote is registered by increment of the corresponding bin. Although there will be many votes for lines (r, θ) that do not constitute actual edges of the hexagon, peaks will develop for bins representing real edges since all points along the same border will vote for this pair

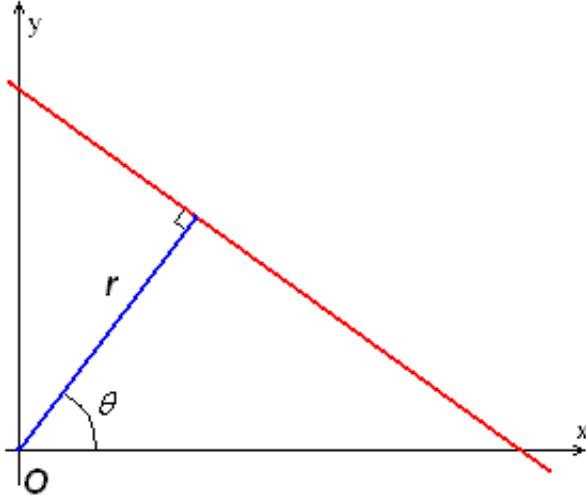


Figure 4.2: (r, θ) -parametrization of a line in (x, y) -space. [23]

of parameters among the multitude of other fictional lines with only very few votes. This way the hexagon's outer shape, and by extension its center, can be reconstructed from the image.

A significant advantage of the Hough transform is the ability to cope with gaps in the parametric features. However, one needs to be able to parametrize the wanted features in the first place plus the method's accuracy and robustness degrade quickly with a decaying sharpness of features.

4.2.2 Image Filtering

The set of real images used in the comparison contains 54 difference images (Fig. 4.3, Top), based on photos recorded by the SingleCCD camera. Before the center detection techniques are applied the images are subjected to a preprocessing procedure, which can be split into two different steps. In the plain non-difference 16-bit CCD images every pixel can have an amplitude $0 \leq p < 65,535$, with a mean signal strength of around 23% of the saturation level of the camera, i.e. $p_{\text{signal}} \approx 15,000$, and camera background noise $p_{\text{noise}} \approx 2000$. The pixel amplitudes are normalized by dividing the amplitude of every pixel by the mean signal strength, and the difference image is constructed by subtraction of one image from the other for every two paired images. For Hough transform the quality of the resulting images is not sufficiently high as especially the edges of the hexagons are not sharp enough and further image enhancement is required (Fig. 4.3, Bottom). Without this additional step the algorithm will completely fail or return centers outside the dimensions of the image for most of the images in the dataset.

For the enhancement process the difference image is separated into its positive and negative part, each containing one hexagon. The following steps are done separately for both hexagons, where for cut purposes the absolutes of all magnitudes are used. In the current configuration the background rejection is done by setting all pixels with amplitude $|p| < 0.05$ as well as the dimmest 15% and brightest 3% of remaining pixels to zero. These parameters were visually observed to provide the images with best hexagon edge crispness and acceptable remaining background noise.

Afterwards, contrast is further improved by setting the amplitude of all remaining non-zero pixels to unity, so that every pixel has either an amplitude of unity or zero. Lastly, leftover stray pixels and holes are eliminated by analysing every pixel's close neighbor-

hood. If a pixel's amplitude is unity but only few of its immediate neighbors (< 3) share this property then the pixel is set to zero and the other way around for pixels with $p = 0$ and at least three neighbors with amplitude of unity.

4.2.3 Canny Edge Detector

Before the actual Hough transform and edge parameter extraction can be applied the pixels along the hexagon's borders must be identified. This is done via a simple version of the Canny Edge Detector algorithm in conjunction with the Sobel operator[24]. It is based on the idea, that the gradient of pixel values orthogonal to an edge should be substantially larger than at other places in the image. Therefore the first derivative in the horizontal and vertical direction (g_x/g_y) is determined for each pixel by folding a 3x3 image-kernel centered around the current pixel with the Sobel-operator in x/y-direction. The choice of Sobel operators in this case was

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} .$$

This results in a separate gradient map g_x/g_y for both Cartesian directions which are combined into one single absolute gradient map by euclidean addition as

$$G(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2} . \quad (8)$$

The gradient strength of an edge in one of the directional maps depends on its orientation, however, the total gradient is independent of it. In addition the gradient's direction θ is calculated from the two gradient maps, indicating the direction of strongest change with

$$\theta = \arctan(g_y, g_x) . \quad (9)$$

This theta is equivalent to the one used for the (r, θ) line parametrization (Fig. 4.2) but is only used for background rejection and not for characterization of potential hexagon edges. For practical purposes the resulting values are rounded to 0° , 45° , 90° and 135° since the pixels are ordered in a discrete orthogonal grid and each of them only has 8 immediate neighbors.

Afterwards a cut is applied setting all pixels of the absolute gradient map whose value is less than one half of the largest gradient to zero, essentially removing most of the points not along one of the edges. Finally non-maximum suppression is applied to the remaining pixels as a measure to decrease line thickness. For every non-zero point the two neighbors along the gradient direction are evaluated and if one of them is larger then the point is set to zero. This way ideally an edge with thickness of only one pixel is left by removing every pixel except the ones on the "mountain ridge". The results of the Canny Edge Detector applied to Fig. 4.3 (Bottom) are shown in Fig. 4.4.

4.2.4 Implementation of the Hough-Transform

The algorithm, including the preceding image enhancement and Canny edge detector, was implemented in C++ utilizing the ROOT Data Analysis Framework[25].

As described in section 4.2.1 the parameters defining the Bokeh hexagon's edges are now determined by a voting process over all the remaining pixels. The peaks in the (r, θ) -space, and therefore possible edges, are acquired by taking the parameters corresponding to the

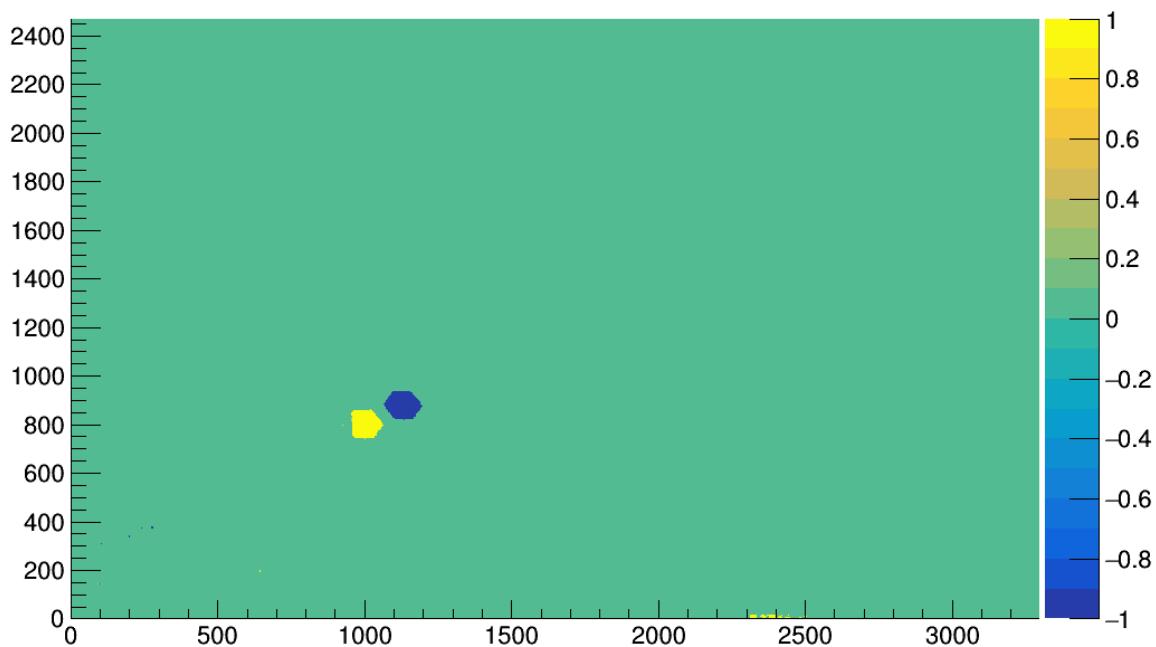
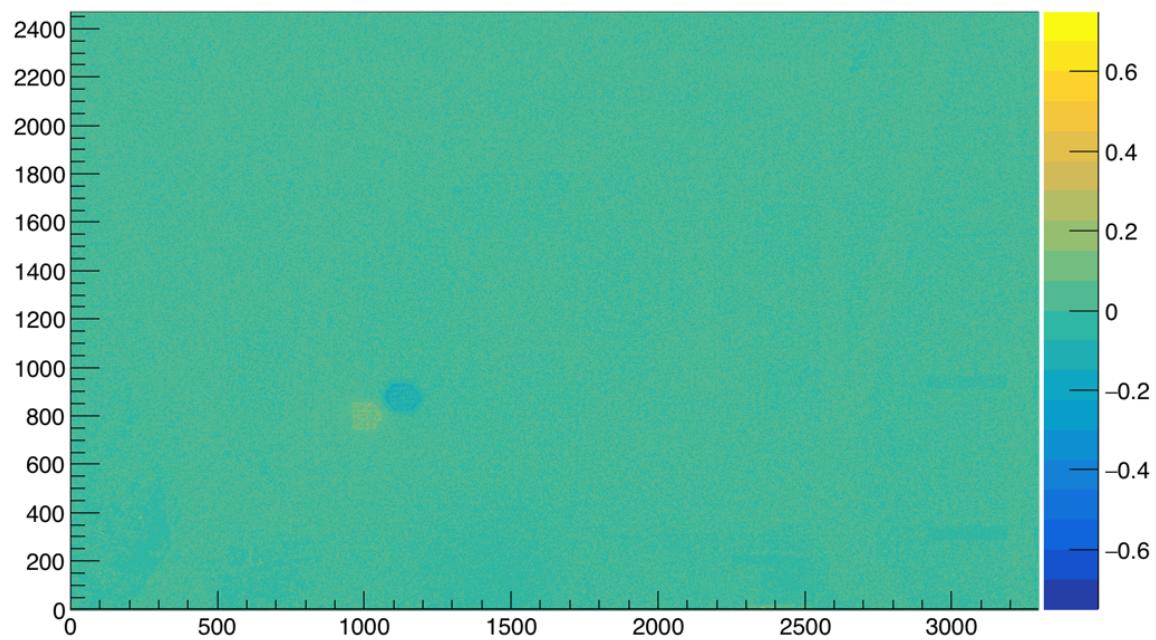


Figure 4.3: A difference image from 2 photos taken on Run 232 at the MST prototype, Berlin, by the SingleCCD camera. Top: after normalization, Bottom: after the additional enhancement for Hough. The axis are in CCD camera pixels where each pixel corresponds to about 1.7 mm in the focal plane[19]. The contrast is significantly sharper after enhancement and most irregularities were removed.

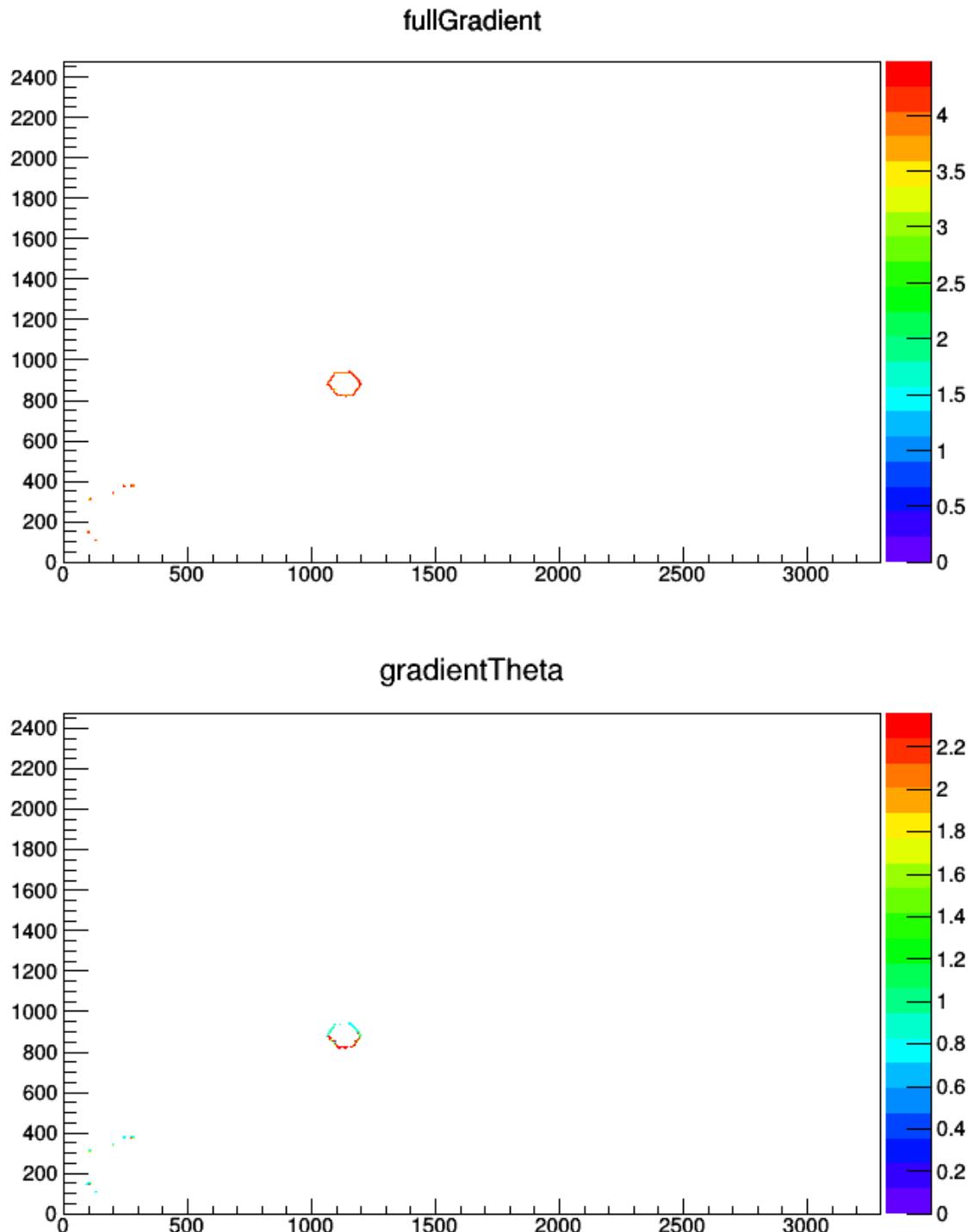


Figure 4.4: Results of the Canny Edge Detector algorithm on the negative part (blue hexagon) of the enhanced difference image from Fig. 4.3. Top: The absolute gradient $G(x, y)$ (Eq. 8), Bottom: Edge direction angle θ in radians (Eq. 9).

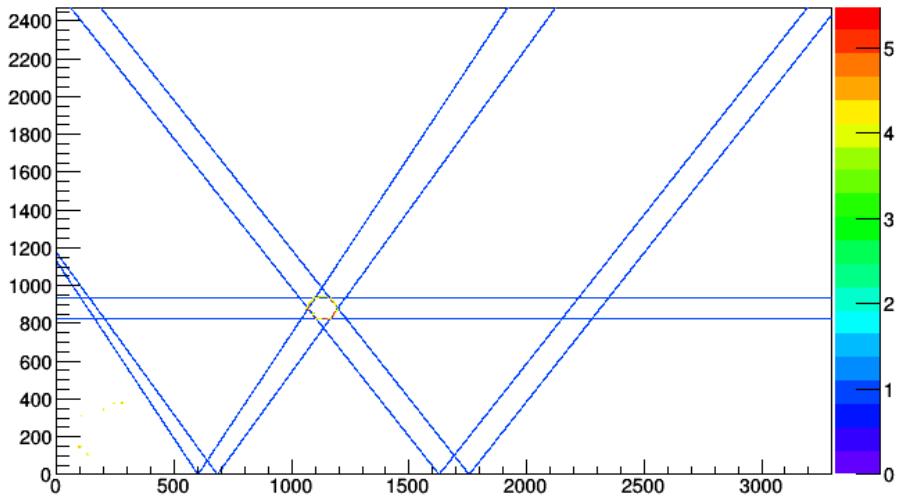


Figure 4.5: Good edges (blue) found by the Hough-algorithm for the image-data from Fig. 4.4 (Top).

bins with highest counts, but since the values for the angle θ and distances between edges are already relatively well known due to the geometry of hexagons, only specific values are considered for further analysis. Specifically all possible edges are left out were theta is not close to one of the values expected for normal hexagons, 30° , 90° or 150° . A range of four degrees around these values was used to include edges with a slight deviation from the ideal case. This assumes that the hexagons are not substantially rotated compared to the case where their upper and lower edge are parallel to the x-axis. Although this limits the scope of the implemented Hough-based technique it is required to avoid the algorithm mistakenly identifying non-existing edges over existing ones. As the quality of the images is relatively poor, even after enhancement, there usually exist several bins with high counts that do not correspond to any real hexagon edge. However, by enforcing a strict boundary on allowed angles they can largely be discarded.

In the next step the algorithm tries to combine suitable edges into pairs, where a pair is composed of the two parallel opposite edges of the hexagon. This is done by finding edges with an angle difference $\Delta\theta < 5^\circ$ and a distance $70 \text{ px} < d < 110 \text{ px}$. The expectation for the distance is fixed by the size of the mirror's Bokeh pattern and was estimated by empirical analysis of the given set of images. For possible pairs where the distance is within the given limit but the angle difference is above five degrees, the weaker edge, i.e. the one with less votes, is removed from further considerations. The good edges identified by the algorithm in the data from Fig. 4.4 are displayed in Fig. 4.5.

In the final step those edges are used to determine the hexagon's center and thus its position in (x, y) -space. This is done via two different methods which will be called "Geometric Shifting" and "Monte Carlo Accumulation" in the following.

- **Geometric Shifting** exploits the fact that the sides of hexagons are equal in length to the distance between a corner and the hexagon's center. If the previous algorithm was able to find 3 edges that form a continuous border together, e.g. Fig. 4.6 (Top), then this property can be used to determine the center by parallel shifting edge 1 so that it goes through the intersection of edge 2 and 3 and vice versa for edge 3 (see Fig. 4.6). The resulting intersection of shifted line 1 and 3 now marks the

hexagon's center. This method's advantage lies in its ability to function even if only 3 consecutive edges have been found, i.e. it does not need a complimentary pair of edges. If multiple consecutive "3-pairs"³ have been found then the method is applied to every one of them separately and the mean of all resulting center-coordinates is calculated.

- **Monte Carlo Accumulation** requires at least 2 pairs of edges to work which will automatically form a parallelogram (example see Fig. 4.7) whose center is identical to the hexagon's center. Due to the finite resolution and errors in the recorded CCD images this will not be the exact case, however, it is a reasonably good approximation. The parallelogram's center is now determined by randomly creating a large number of points with coordinates within the bounds of the image dimensions, but keeping only those within the parallelogram and calculating their mean (\bar{x}, \bar{y}). This returns an estimate for the parallelogram's center and thus the center of the hexagon. Although it requires more stringent operating conditions, with at least two pairs of edges necessary, it was observed to generally yield better results compared to Geometric Shifting even if multiple 3-pairs were available for averaging.

Because of the mentioned (dis-)advantages of both methods the implemented Hough algorithm returns the MCA result exclusively if it was available and doesn't average it with the Geometric Shifting output.

Due to its simplicity a small function calculating the center-of-gravity (CoG) of all pixels, after image enhancement, was also implemented alongside the Hough transform algorithm, providing another method for comparison in section 4.4. The CoG algorithm uses the same, further enhanced, images as the Hough technique.

4.3 Convolutional Neural Networks

4.3.1 Introduction

The second proposed solution for hexagon detection utilizes Convolutional Neural Networks (CNNs). Due to substantial advancements in computing power and understanding of Neural Networks (NNs) in recent years, they have become a commonly used technique in data analysis tasks, especially in the field of computer vision, offering substantial improvements compared to more classical approaches. This makes them a promising tool for detecting the centers of the Bokeh hexagons in this particular case.

4.3.2 General Concept

Convolutional Neural Networks form an expanding area of active research in their own right. Therefore the following section will only present a short overview of the most important aspects for this specific context. For a more exhaustive summary see [26] or for an in-depth introduction the book by Ian Goodfellow et al[27].

Neural networks in general can be understood as multidimensional decision functions, usually taking a larger vector of input values and returning a result based on a complex dependence on the input, with the form of the result depending on the specific network. The substructure of CNNs is typically composed of several key components as well as

³In this case "3-pair" refers to a group of 3 consecutive edges of the hexagon.

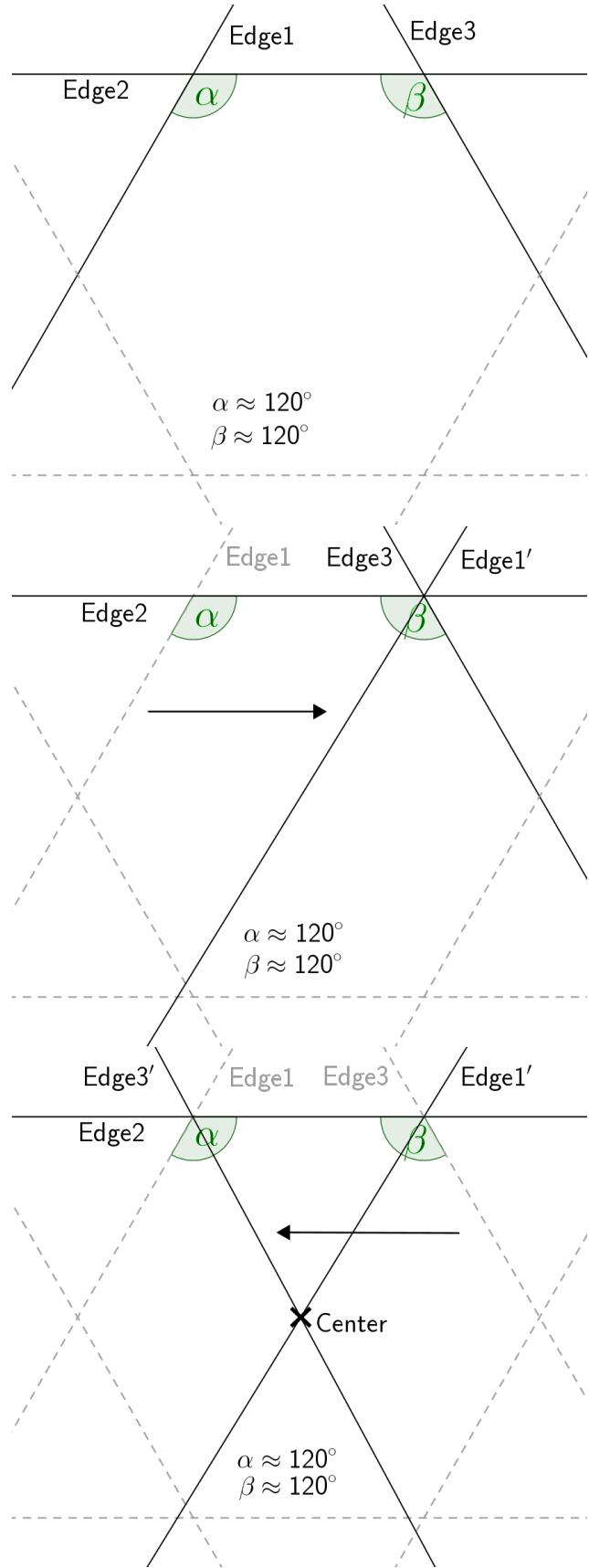


Figure 4.6: General process for geometric shifting method. Top: 3 connected edges have been found. Center: edge 1 is parallel-shifted to the intersection between edge 2 and 3. Bottom: analogous for edge 3. The intersection between the two shifted edges gives an approximation for the hexagon's center.

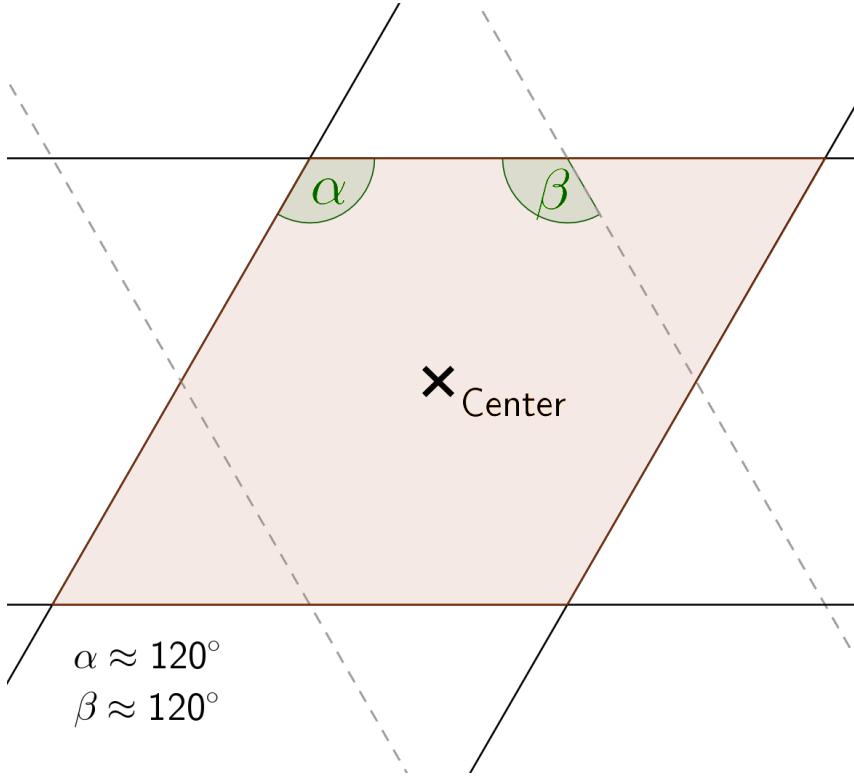


Figure 4.7: Schematic of Monte Carlo Accumulation process for determination of the hexagon's center. Two edge pairs are required for finding the center of the constructed parallelogram and by extension of the hexagon.

some optional, specialized modules. In their simplest form CNNs are made from a number of convolutional layers, hence the name, followed by fully connected layers (linear layers) and interspersed with pooling layers (Fig. 4.8). In addition a non-linearity generator/activation function is attached to every convolutional and linear layer. The non-linearity is required to allow for the network's result to be modeled by a non-linear dependency on the input and effectively divides the different layers into separate entities. Without the non-linearity all the different layers (linear in themselves) could be modeled as one single linear layer instead, providing no possibility to include non-linear dependencies.

The central difference between neural networks and classical methods is the training required for NNs. For classical approaches the developed algorithm will never do anything it was not explicitly programmed for, however, this is not the case for NNs. They must instead be trained to solve a specific problem before they can be used for actual analysis tasks. CNNs are a special form of the more general group of neural networks with their distinction being the use of the mentioned convolutional layers, which can be thought of as non-fully connected linear layers with partially shared weights. This reduces the scaling of processing power for problems with high-dimensional input vectors, e.g. images with several hundred or thousand pixels.

The training is necessary because essentially a CNN functions by subjecting the input data to weight-based filters (convolutional layers) and correlation selectors (linear layers), where the weights are not predefined but are learned by the network itself in a process called backpropagation. This will be explained in more detail later. It is important to note that during training the network's code itself is not changed but only the weights are adjusted.

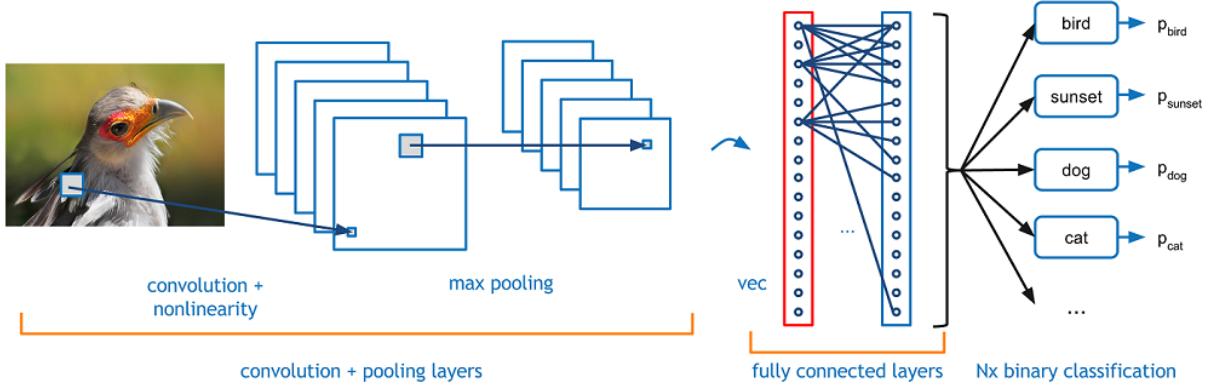


Figure 4.8: General structure of Convolutional Neural Networks.[26]

The general idea now is to have the computer employ the same method humans subconsciously use when viewing images. Start by extracting simple shapes from the image in the earlier convolutional layers and construct increasingly complex features from these in the later ones, finally connecting all the detected features in the linear layers, in the end allowing the classification and/or localization of complicated objects[26]. For example when an image of a human is used as input for the net, then the first layers will only focus on very simple geometric shapes like slanted lines in this or that direction, or curves of various radii. The following layers will then detect e.g. a hand if the previous layers found a sufficient number of fingers and in the end the fully connected layers will decide whether the input image represents a human or other object (classification) or were the object is located (location) based on the findings of preceding layers.

Convolutional layers work by convolving a filter (e.g. Fig. 4.9), with weights determined by training, with sections of the image. The result of the filter for some section/kernel from the image with same size as the filter is simply the sum S of the amplitudes p_{ij} of all relevant image pixels ij weighted with the value w_{ij} of the filter at the corresponding positions, i.e.

$$S = \sum_{ij} w_{ij} p_{ij} . \quad (10)$$

If the currently selected image kernel contains the feature this particular filter is trained on, then the operation will output a larger value, and a lower one if not. The result of the filter's convolution over the whole image is called an activation map or feature map and is fed into the subsequent layer after being passed through a non-linearity generator. Between layers the map may also be injected into a pooling layer whose purpose is shrinking the map's size by compressing neighboring pixels by means of different possible methods, e.g. simple averaging. This increases the perceptive field of subsequent filters which might now be able to "see" more adjacent results of the previous filter. Every convolutional layer usually consists of multiple filters working in parallel on the input data and each supplying its activation map to the following layers.

After passing through the convolutional layers the resulting activation maps are now used as input for the fully connected/linear layers. These consists of a large number of neurons where every neuron is connected with every neuron of the previous and following layer. All connections between neurons have an individual weight expressing their correlation strength. The network is now ideally able to identify cross correlations and output a high value for the most likely answer, e.g. if there are 2 hands, 2 feet and a human head on the image chances are high it is a human and rather low it might be an elephant

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

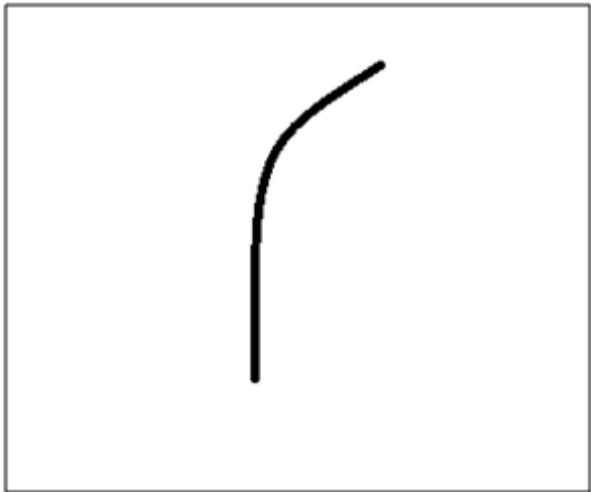


Figure 4.9: Example for a possible filter used in a convolutional layer. Left: Weight matrix of the filter; every element represents the weight applied to the image pixel at this place. Right: The type of curve this filter was trained on and for which it will return a high value. [26]

(classification).

However, networks can be used for much more than just classification of objects. One might also have them output the area of a roof given a picture of it, or the time of day a photo was likely made at. It depends entirely on the training of the network.

This leads to the, probably, most crucial point of working with CNNs, the training, i.e. the process of finding the best possible weights for every filter and neuron for achieving the specified analysis. Typically training is done by using the method of backpropagation, which is essentially based on supplying the network with a large number ($\mathcal{O} \geq 10,000$) sample images⁴ and the desired output of the network for every image. The optimal values for all filters, based on the given desired output, are now reverse-engineered starting from the back of the network's processing chain and propagating the changes through the network structure in the reverse direction compared to the normal analysis process. There exist multiple techniques for backpropagation, differing in the details of how the propagation is done.

While proper CNN construction is an art itself, an increase in network depth (number of layers) generally leads to a higher accuracy, however, one needs a large amount of training data to avoid overfitting, i.e. overoptimizing on patterns in the training data that are not present in general. The effect of overfitting can be reduced by adding special dropout layers between the existing linear layers. They randomly turn off the connections from a defined fraction of neurons of the preceding layer to the next layer. Since the disabled neurons are chosen at random and are changed for every training image this reduces the creation of overly strong connections between individual neurons and spreads the detection more evenly over the correlation strength of all neurons instead of just a select few contributing the majority to the final result. After training is done the dropout is turned off and all neurons are used for image analysis. However, the downside of using dropout layers is the reduction of available neurons, as a result typically reducing the CNNs accuracy. Therefore the reduction in overfitting has to be compared to the overall loss in accuracy to evaluate the effectiveness of using dropout layers on a case to case basis.

⁴For other analysis tasks, e.g. audio analysis, the sample data is in the respectively required format.

4.3.3 CNN Implementation

Here the desired output of the network is the center of the Bokeh hexagon, given a difference image as input. For processing the image is split into its positive and negative part, which are evaluated independently by the CNN. The network was implemented in Python with PyTorch[28] based on the open source machine learning library Torch. It utilizes code written for the analysis of images from H.E.S.S. by T. Holch (HU Berlin). Many different network structures with different settings for the various hyper-parameters (learning rate, weight decay, etc.) were tested and in the end a structure of 6 convolutional, 5 pooling and 5 linear and no dropout layers (Fig. 4.10) was chosen for its good compromise between accuracy and training duration. The nonlinearity is provided by the rectified linear unit function $\text{ReLU}(x) = \max(0, x)$ applied to the outputs of all conv. and linear layers.

The training (learning of weights and filters) was done via Stochastic Gradient Descent (SGD) on a dataset of 80,000 training images for 100 epochs. After every epoch the network is automatically evaluated with an independent set of 16,000 validation images, to assess how good it performs for images it was not trained on. Since no real CCD images with a known true center (ground truth) are available both the training and validation data is generated by a generator-script written in Python which tries to approximate the hexagons found in real images (e.g. Fig. 4.11). The loss l , i.e. difference between network answer a and supplied ground truth t , was calculated with a simple L1Loss-function, where $N = 16$ is the batch size, as

$$l(a, t) = \text{mean}(L) = \text{mean}(\{l_1, l_2, \dots, l_N\}^T), \quad l_n = |a_n - t_n|. \quad (11)$$

The batch size determines the number of images evaluated simultaneously for the training as a batch/group, which speeds up the training significantly due to parallel-processing capabilities of modern GPUs. For every batch the network's weights are only adjusted once based on the results of all images in the batch combined, instead of individually for every image.

A RLROP (Reduce Learning Rate On Plateau) scheduler was used to decrease the rate by which the weights are adjusted every iteration after there was no significant improvement over the last few epochs. This allows to achieve higher accuracies by avoiding overcompensating with the weights. The learning curve of the above net is shown in Fig. 4.12.

Due to hardware limitations the largest images the network can be trained on in a timely manner are of size $\mathcal{O}(100 \text{ px} \times 100 \text{ px})$ which is substantially smaller than the full SingleCCD images with size $3296 \text{ px} \times 2472 \text{ px}$. Therefore, only the immediate region of the CCD image containing the actual Bokeh hexagon can be fed into the network. The region is chosen as a $400 \text{ px} \times 400 \text{ px}$ image section of $\pm 200 \text{ px}$ in x- and y-direction around the center returned by the WCF mentioned in section 4.1. This approach includes the relevant hexagon in all analyzed 54×2 (positive/negative) SingleCCD images. In the final step of preparation the resulting image is down-sampled, by averaging over $4 \text{ px} \times 4 \text{ px}$ image kernels, to a size of $100 \text{ px} \times 100 \text{ px}$ and then used as input for the network's first layer. As an example, the selected, down-sampled region for the negative part (blue hexagon) of Fig. 4.3 (Top) is shown in Fig. 4.13. For all CCD images the network never mistakes the hole, left by the removed hexagon with opposite sign, as the desired hexagon.

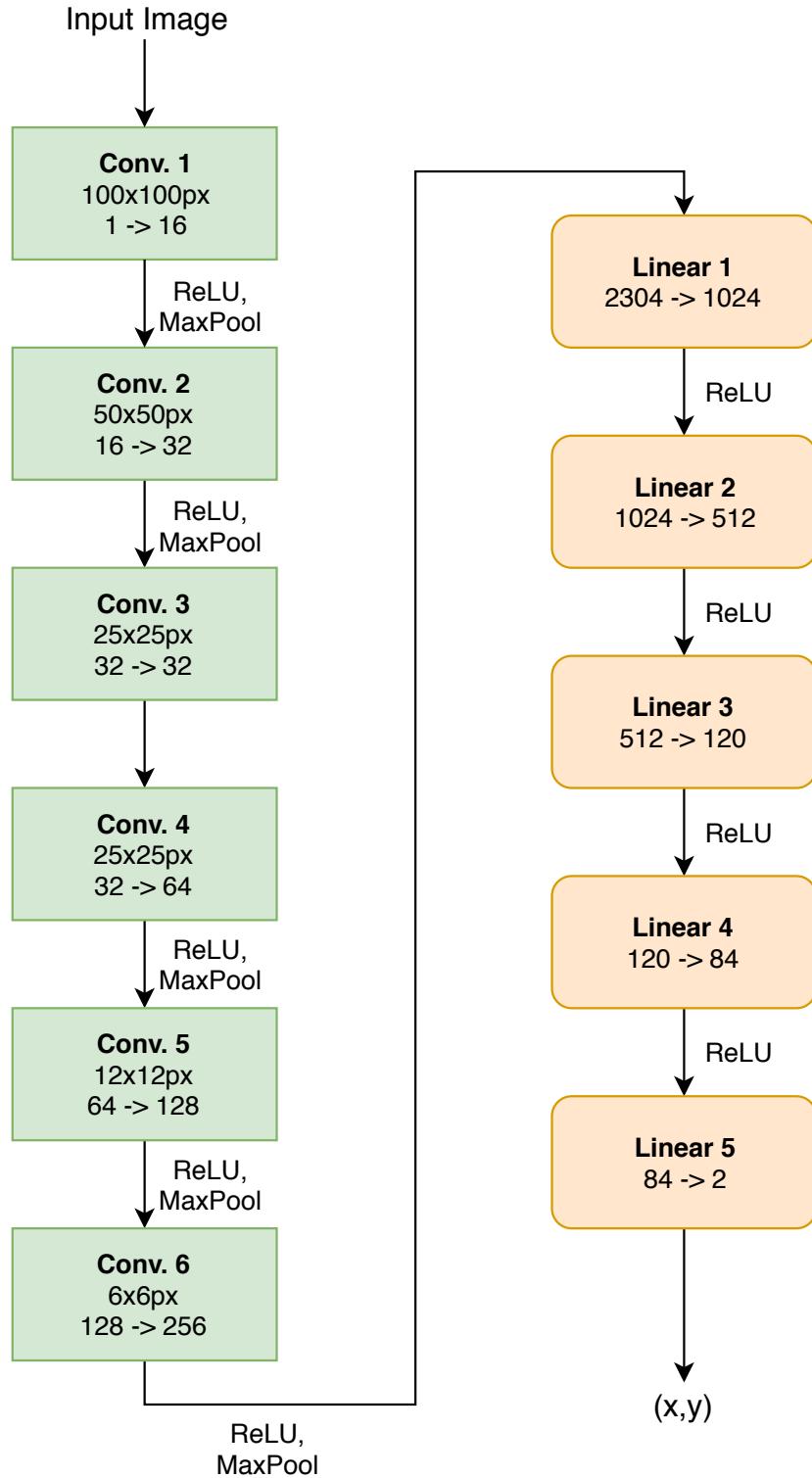


Figure 4.10: Structure of the implemented CNN for detection of hexagon centers from $100 \text{ px} \times 100 \text{ px}$ images. 'Conv.' are the convolutional layers and 'Linear' the linear/fully connected layers. 'ReLU' is the rectified linear unit $\text{ReLU}(x) = \max(0, x)$ that is used to introduce a non-linearity into the network, and 'MaxPool' shrinks the image/map with a 2×2 kernel, i.e. by a factor of two, by replacing every non-overlapping 2×2 image region with the maximum of the section. For the convolutional layers the first line gives the size of the map that is used as input to the layer, which is equal to the size of the layer's output map (only for this setup; not generally). The second line of the conv. layers/first line of the lin. layers shows the number of input and output channels. After the last convolutional layer with added ReLU and MaxPool the result are 256 different $3 \text{ px} \times 3 \text{ px}$ activation maps, the input size to the first linear layer is therefore $256 \cdot 3^2 = 2304$.

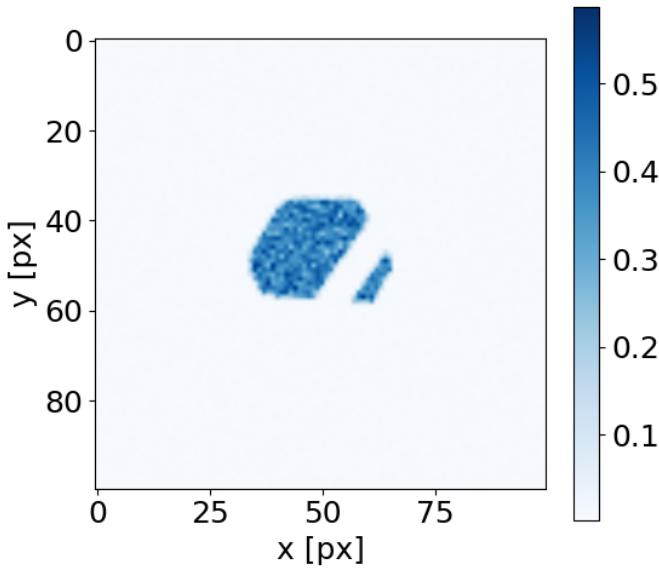


Figure 4.11: Example for a simulated image generated by the script. Goal was the best possible imitation of real images, e.g. Fig. 4.13, including shadowing by the CSS.

4.4 Comparison of the different methods

4.4.1 Simulated Data

The generator-script mentioned in the previous section was used to generate another batch of 300 simulated $3296 \text{ px} \times 2472 \text{ px}$ -sized images, each containing a single hexagon. These were then processed in the same way as the real CCD images.

For all evaluated methods the difference between answer and ground truth (loss) in both x- and y-direction was calculated and sorted in a histogram. The resulting distributions all exhibit a spread with several outliers substantially above the central part of the distribution. The histograms were fitted with a Gaussian, however, to obtain sensible results, a range of reasonable values for fitting had to be determined and all points outside rejected. This was done by calculating the median for each distribution respectively (Tab. 4.1) and including only the points within an interval with size of one standard deviation to either side in the actual Gaussian fit.

For an implemented algorithm to be considered as unbiased in this context the true mean loss of zero was required to be included by the 99.7% confidence interval of the mean difference μ for both the x- and y-coordinate. As seen from the results (Fig. 4.14, Tab. 4.2), this is only the case for the technique based on Hough transform⁵ as well as the Weighted Circle Fitting method where the value of zero bias is already included by the 68% confidence interval. Comparing both of these methods, WCF is substantially better than Hough in terms of accuracy since the bias and uncertainty are smaller by at least a factor of two in both x- and y-direction. Due to the large spread compared to the other methods, it is much harder to prove a bias for the Hough method and the result should therefore not be taken as confirmation for the algorithm's unbiasedness. For the Hough-based algorithm the bias in x is roughly three times the size of the bias in y, but

⁵All images where the algorithm failed to find a center within the bounds of the image were excluded from the analysis.

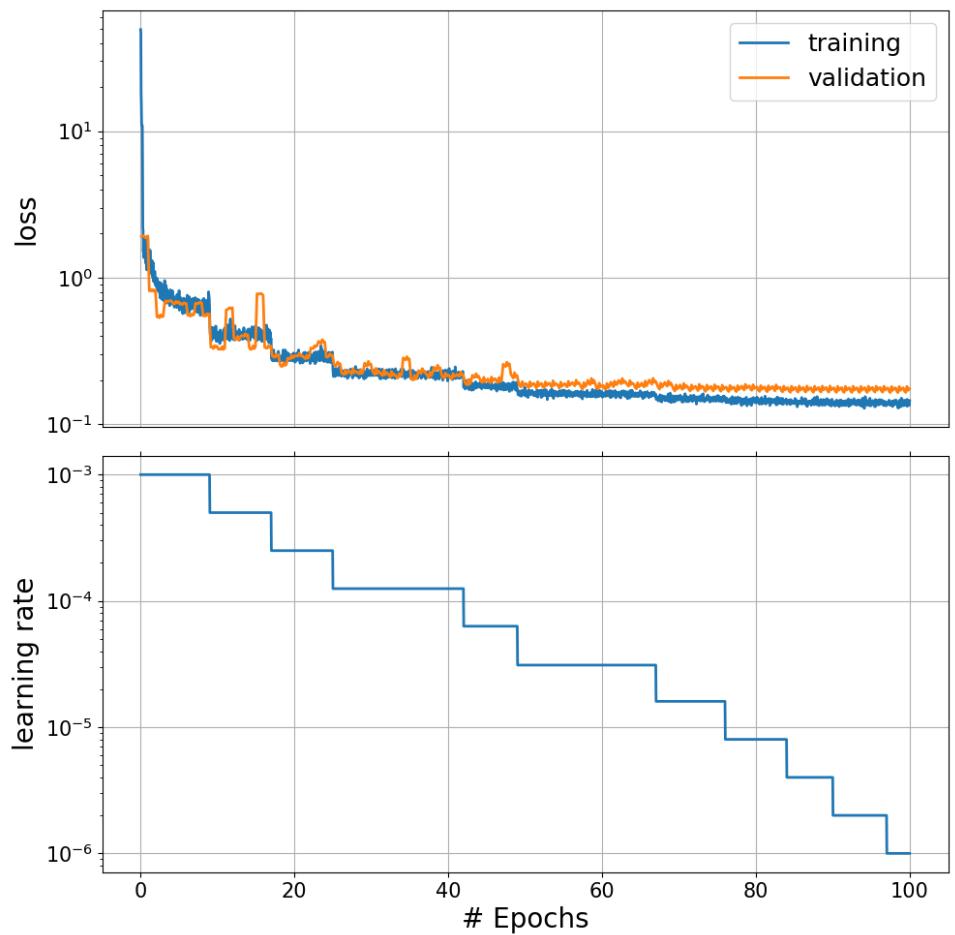


Figure 4.12: Learning curve for the CNN described in section 4.3.3. Top: difference between network result and image label ("loss") for training and validation data. Bottom: Decrease in the learning/adjustment rate of the net's weights. Only minor improvements in accuracy are made past epoch ≈ 80 .

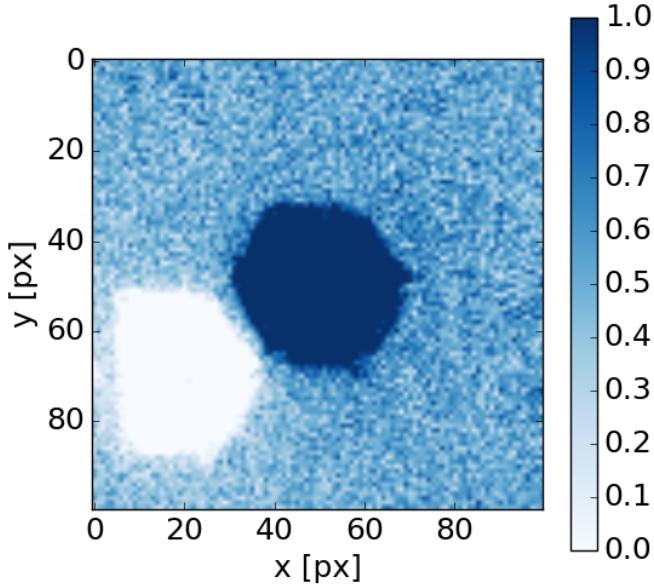


Figure 4.13: The center ± 200 px area around the center returned by the WCF method for the negative (blue) hexagon from Fig. 4.3, down-sampled by a factor of 4 to a size of 100 px \times 100 px. The relevant Bokeh hexagon (dark blue) is well within the selected region and the CNN never mistakes the hole of the removed with opposite sign as the relevant hexagon.

this is very likely just a statistical fluctuation as the uncertainty of the ratio is large and the neutral value $x/y = 1$ is easily included within one confidence interval

$$\frac{\Delta x_{\text{Hough}}}{\Delta y_{\text{Hough}}} = 3 \pm 4 \quad . \quad (12)$$

Both the Center-of-Gravity and the CNN approach exhibit a bias that can not be explained by statistical fluctuations alone since zero is not included in the 99.7% confidence interval, neither in x- nor y-direction. For the CoG method the bias itself is on the same order as for WCF, however, the confidence intervals for the loss in x and y are smaller than their WCF counterparts and therefore lead to a statistically significant bias of the CoG method.

The distribution of losses for the convolutional neural network exhibits the second lowest spread, after CoG, but with the largest bias of all methods roughly one order of magnitude larger than for WCF. The intended value of zero bias is not included in the 99.7% confidence interval of the mean loss (in x and y) and the distribution can therefore not be considered unbiased. Interestingly this substantial bias is not caused by the CNN itself but rather introduced by the process of transforming the 3296 px \times 2472 px-sized images into the 100 px \times 100 px format processable by the network. The exact origin of this bias is currently unknown, but its influence can be proven by feeding the CNN with images directly generated in size 100 px \times 100 px, as done for the training. For a sample of two thousand 100 px images the same analysis as above for the large images gives Gaussian distributions (Fig. 4.15) with parameters

$$\begin{aligned} \overline{\Delta x} &= (-0.009 \pm 0.002) \text{ px} \\ \overline{\Delta y} &= (0.014 \pm 0.004) \text{ px} , \end{aligned} \quad (13)$$

Method	Median loss in x [px]	Std. Dev. of Distrib. in x [px]	Median loss in y [px]	Std. Dev. of Distrib. in y [px]
Hough	0.6	10.3	0.3	10.8
CoG	0.05	6.0	0.2	4.2
CNN	-2.5	0.6	-1.4	1.6
modified CNN	-0.4	1.0	-0.2	0.7
WCF	-0.13	4.31	-0.05	4.08

Table 4.1: Median and standard deviation for the loss-distributions of the evaluated image analysis methods, based on 300 simulated images. 'Loss' is the difference between the center-coordinates returned by the respective method and the known ground truth. For the 'modified CNN' the cut-out region was shifted by $+2/-1$ px in x/y in the preceding image transformation, based on empirical observations.

Method	Mean loss in x [px]	Mean loss in y [px]	N_{Fails}/N
Hough	1.2 ± 0.4	0.4 ± 0.5	46 / 300
CoG	0.097 ± 0.013	0.197 ± 0.011	0 / 300
CNN	-2.460 ± 0.025	-1.376 ± 0.025	0 / 300
modified CNN	-0.318 ± 0.027	-0.188 ± 0.024	0 / 300
WCF	-0.14 ± 0.21	0.19 ± 0.20	2 / 300

Table 4.2: Results for the Gaussian fit of the different methods used for image evaluation based on 300 simulated images. 'Loss' refers to the difference between the method's answer and the reference truth, i.e. $\mu \pm t \cdot \frac{\sigma}{\sqrt{N}}$ with $N = 300 - N_{Fails}$ the number of available datapoints. ' N'_{Fails} ' is the amount of events where the particular algorithm failed to find a center within the bounds of the image. For a sample size of $N \approx 300$ the Student t-factor t can be reasonably approximated as $t = 1$.

which is still not quite unbiased given the above definition, but much less than for the down-sampled images. Nonetheless, there seems to exist a small, statistically significant, inherent bias of the convolutional neural network.

Although the bias' reason is unknown the accuracy can be substantially improved by shifting the cut-out section around the mirror facet by plus two / minus one pixel(s) in the x-/y-coordinate. This reduces the bias in x and y by roughly an order of magnitude while keeping the spread largely constant (Tab. 4.2 "modified CNN", Fig. 4.16).

One effect that increases both bias and uncertainty is the down-sampling necessary in the process of converting the 400 px image sections around the mirror facet, that are cut out from the main image, into their 100 px counterparts.

An advantage of the CNN is its processing speed, aside from training the network itself which can take from several hours to a couple of days (approx. 10 h in this case), depending on the exact network-structure and available hardware. However, once training is complete and a suitable network model has been constructed the actual evaluation takes only 0.1 s per image (on this hardware) while the Hough script requires around 5 s per image, approximately a factor of 500 longer. Although it should be mentioned that the Hough transform implementation is not really optimized and could certainly be significantly improved, but it is unlikely this will be enough to outperform CNN-based methods. In addition, because of its heavy reliance on clean contours of the mirror facets, the Hough transform is not very stable with a failure⁶ rate of about 15% on the batch of 300 simulated sample images, compared to no failures or a negligible amount for the other three methods (Tab. 4.2).

Analysing the absolute (Euclidean) difference

$$\Delta l = \sqrt{(x - x_{\text{true}})^2 + (y - y_{\text{true}})^2} \quad (14)$$

with respect to the truth also shows a substantial advantage of the CNN-based technique over the one using Hough transform, in terms of accuracy and stability. As seen in Tab. 4.3 (Fig. 4.17) the CNN's precision is better by a factor of about four on average compared to the Hough technique and achieves an accuracy of less than five pixels for 95% of the images together with a comparatively much smaller spread, while for the Hough method 82% of all images have an Euclidean loss above five pixels⁷.

Notably the CNN-based method also delivers a small improvement over the Weighted Circle Fitting method, the current de facto standard and is therefore a promising candidate for future Bokeh alignments. The CNN with modified image transform shows an improved accuracy of around one pixel on average and a loss below one camera pixel for 74% of all tested images.

Nevertheless, it must be stressed that this comparison, based on simulated images, should only be considered as an indicator instead of a definitive statement regarding the different method's accuracies. Although some effort was put into the image generator, there is still a noticeable difference between simulated and real images. Most strikingly, the very simple center-of-gravity algorithm performs roughly as well as the WCF technique on simulated images, while it was already discarded for use on the real images due to a relatively poor performance. This can be attributed to the fact that the simulated images still largely conserve the symmetries required, which is generally not the case for real mirror reflections.

⁶Failure in this case means events where the center returned by the algorithm is out of bounds of the image or none was found at all.

⁷Excluding images where the determined center is out of bounds of the image.

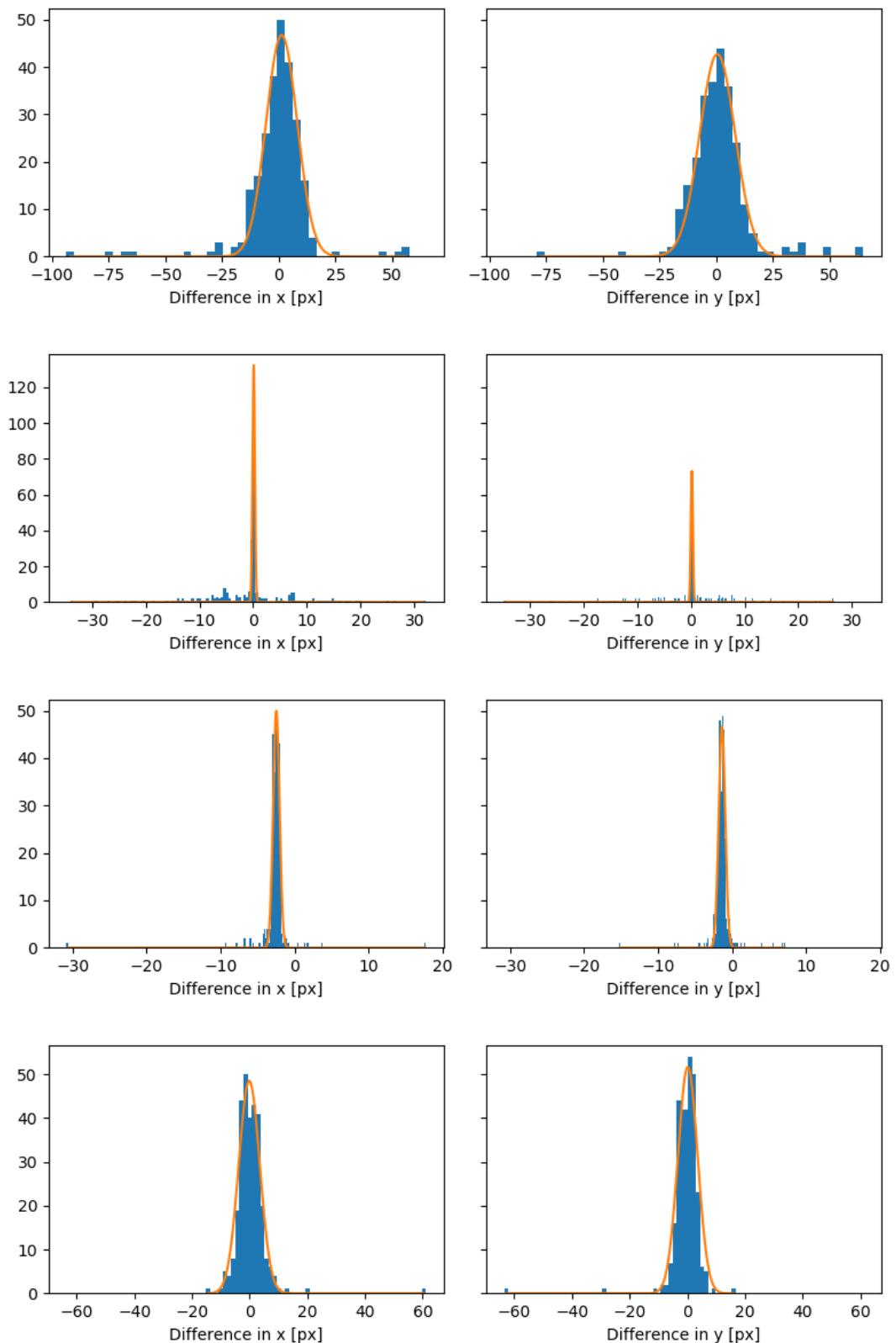


Figure 4.14: Distribution of differences compared to the reference value (truth) for the implemented Hough-based (1. row), Center-of-Gravity (2. row), CNN- (3. row) and WCF-based (4. row) methods using 300 simulated images. Displayed is the loss/difference relative to the ground truth in both x- and y-direction. The parameters of the distributions are shown in Tab. 4.2. Note the different x-axis scaling.

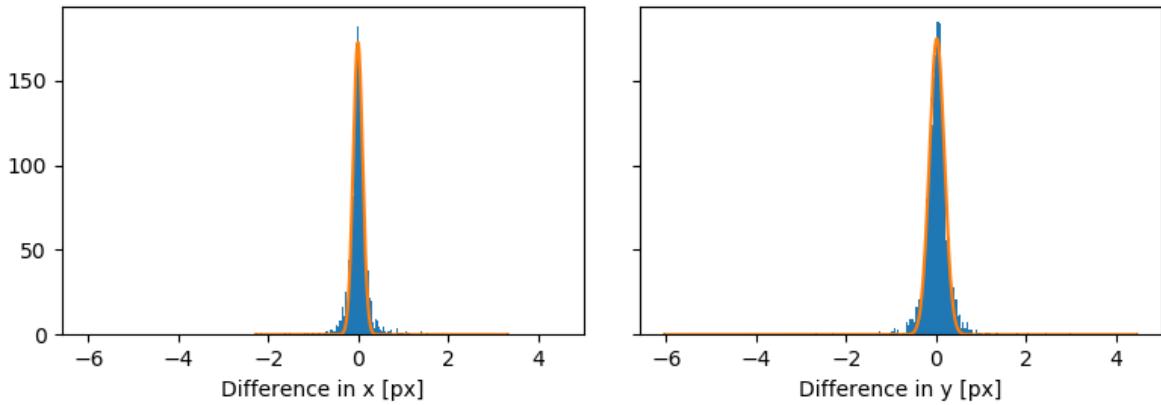


Figure 4.15: Loss/difference relative to the ground truth for the CNN results based on 2000 simulated $100 \text{ px} \times 100 \text{ px}$ images. The bias is substantially smaller than for the down-sampled images (Fig. 4.14, row 3).

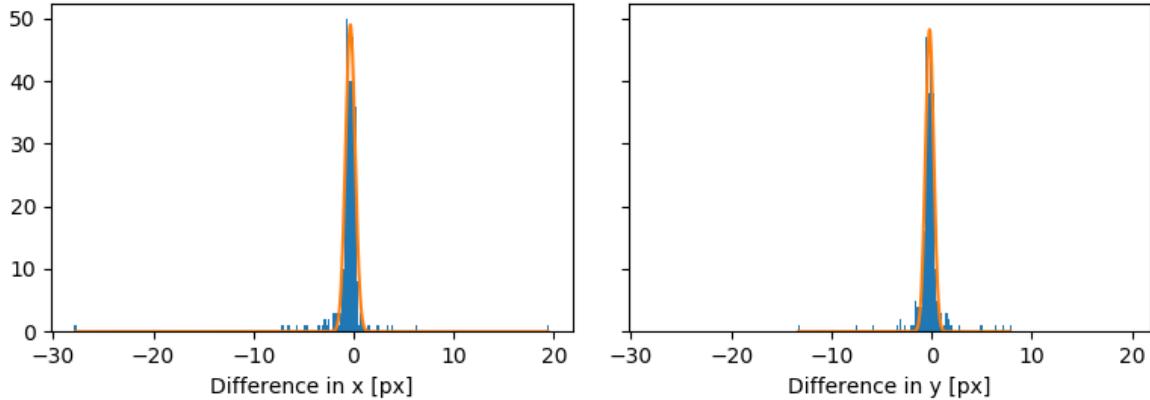


Figure 4.16: Loss/difference relative to the ground truth for the modified CNN results based on 300 simulated images. The region around the presumed hexagon that is cut out from the large image in the preceding image transform is shifted by $+2/-1 \text{ px}$ in x/y, based on empirical observations. This results in a distribution with bias reduced by about one order of magnitude but similar spread compared to the unmodified CNN implementation (Tab. 4.2).

Method	Mean Loss [px]	Std. dev. [px]	Median Loss [px]	MAD [px]
Hough	12.9	14.7	9.0	5.2
CoG	6.6	8.7	1.3	1.8
CNN	3.3	2.3	2.9	0.5
modified CNN	1.2	2.4	0.7	0.4
WCF	4.5	5.7	3.7	1.7

Table 4.3: Characteristics of the distribution of the Euclidean difference/loss $\Delta l = \sqrt{(x - x_{\text{true}})^2 + (y - y_{\text{true}})^2}$ with respect to the true value of the hexagon's center, based on 300 simulated images. MAD is the median absolute deviation. From the difference between mean and median loss the impact of outliers on the mean can be observed. Especially for Center-of-Gravity the mean loss is substantially increased by a few strong outliers, while for the other methods mean and median are much closer together.

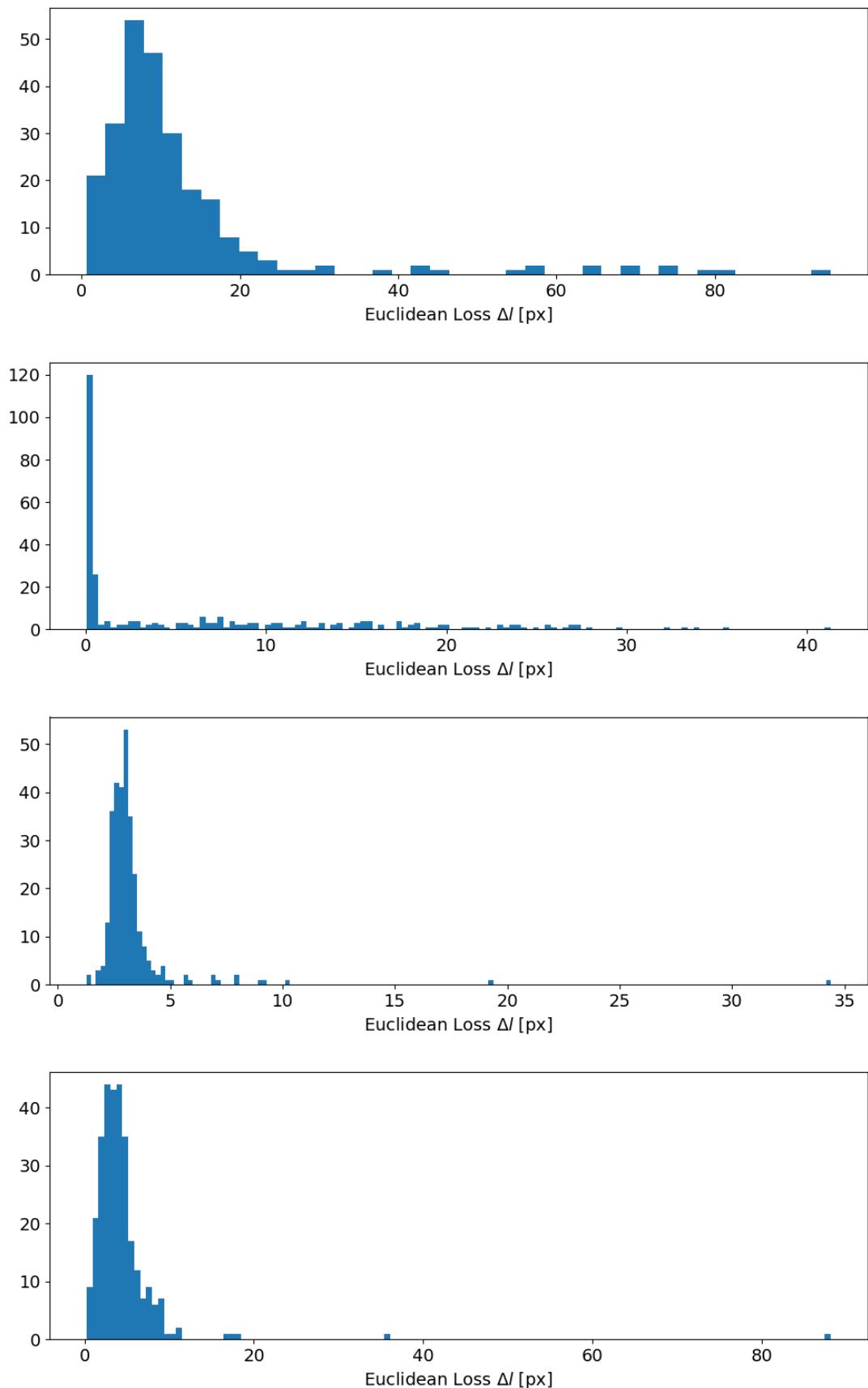


Figure 4.17: Comparison of Hough- (1. row), Center-of-Gravity- (2. row), CNN- (3. row) and Weighted Circle Fitting (4. row) methods based on 300 simulated images. Displayed is the loss/difference $\Delta l = \sqrt{(x - x_{\text{true}})^2 + (y - y_{\text{true}})^2}$ relative to the ground truth. Note the different x-axis scaling.

	CoG	CNN	WCF
Hough	15.1 ± 5.4	14.4 ± 5.5	13.4 ± 5.5
CoG	0	12.4 ± 4.1	12.5 ± 4.6
CNN		0	4.4 ± 1.1

Table 4.4: Comparison of the different methods used for image evaluation based on the MST prototype's dataset of 54 difference images. Shown is the mean of the differences $d_{m1,m2} = \sqrt{(x_{m1} - x_{m2})^2 + (y_{m1} - y_{m2})^2}$ and std. dev. of the found centers over all 2×54 hexagons between the respective methods. The 16 hexagons were the Hough method completely failed to find a center are excluded from the analysis. Further, because of the strong influence on the mean for large differences, only events are included where the difference is less than fifty pixels. For both Hough vs. CNN and Hough vs. WCF the same two events were excluded where the the difference is around one- and two thousand respectively. For comparisons of CoG vs. CNN/WCF 22 images were excluded, 16 for Hough vs. CoG and none for CNN vs. WCF.

4.4.2 Application to Real Observation Data

To get an idea of the different algorithm's performances on real data a set of 54 difference images based on images taken by the SingleCCD (see Sec. 3.2.2) installed on the MST prototype in Berlin, Germany was used. Although no ground truth is available, the differences between the methods can still provide important insights (Tab. 4.4).

Noticeably the difference between Neural Network and Weighted Circle Fitting is relatively small, compared to the other methods, at only around 4 pixels with a smaller spread as well. As WCF is known to perform well on real images this indicates that the CNN seems to perform relatively decent as well. Meanwhile the Hough-based method's result is substantially further away from both their answers and in extensions presumably also from the real answer. Assuming WCF and CNN as close to the truth assigns the Hough algorithm an accuracy on real images approximately equal to the one on the simulated data. The Hough method's stability is similar for the real data as well, with a failure rate of $14/108 = 15\%$ compared to $46/300 = 15\%$ for the test on simulated data. Both accuracy and stability seem to indicate a roughly equal overall performance of the Hough-based technique for the simulated and real data.

However, most striking are the much worse results of the center-of-gravity approach for the real dataset, with a mean difference towards both the CNN- and WCF-based method approximately twice as large as the loss for the simulated images, even after excluding 22/54 images where the difference is above fifty pixels. This coincides with the previous knowledge about the relatively poor performance of simple center-of-gravity approaches on real data.

The test on real images therefore supports the insights gained from the simulated images as the method based on a Convolutional Neural Network seems to be the most promising alternative to the currently used Weighted Circle Fitting.

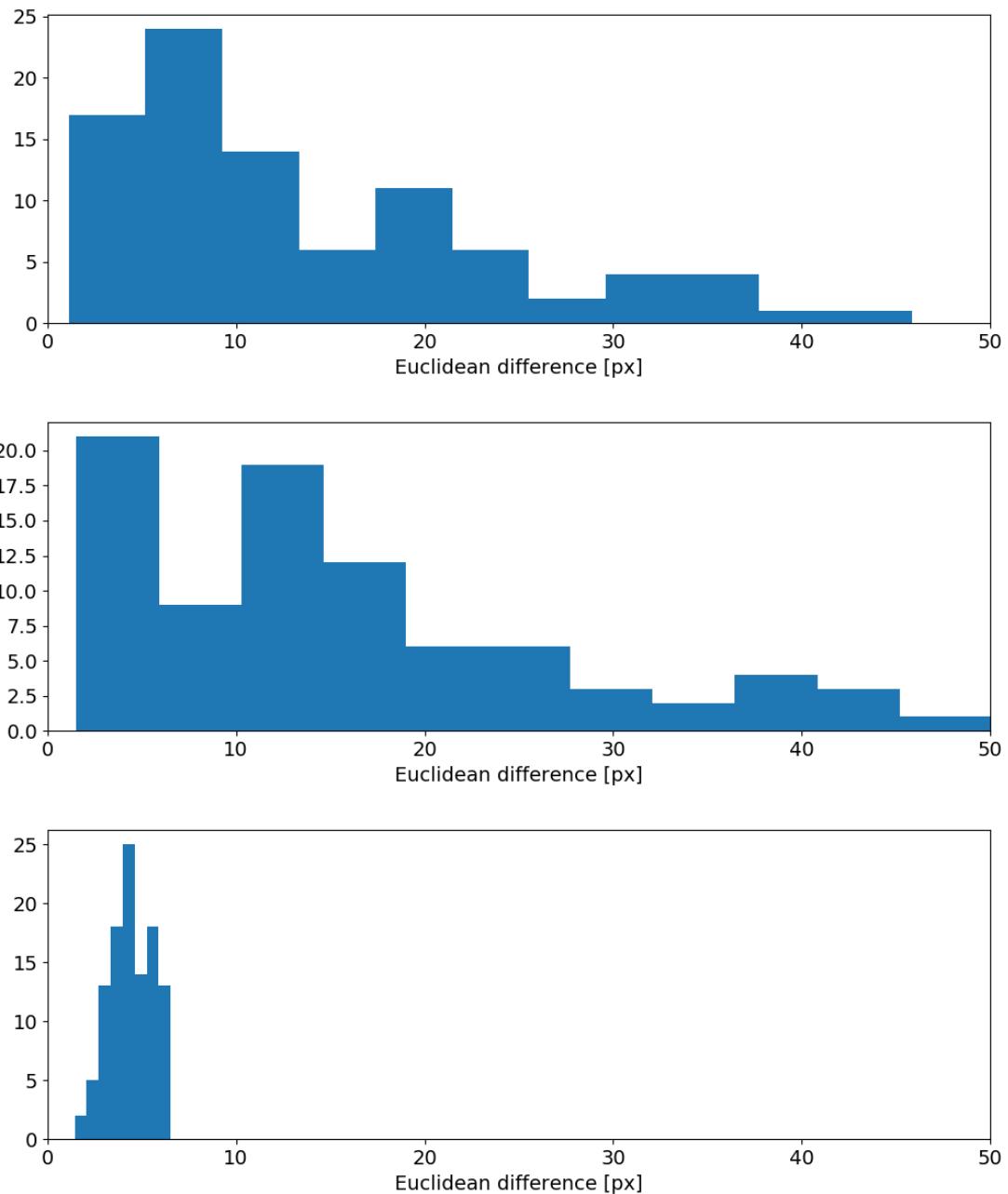


Figure 4.18: Euclidean difference in the coordinates of the determined centers between WCF method and Hough (Top), CoG (Mid) and CNN method (Bottom). All events with differences above fifty pixels are excluded (see Tab 4.4). The CNN-based approach is, on average, substantially closer to the answer of WCF than Hough and CoG.

5 Conclusion

The viability of two new image analysis techniques, based on Convolutional Neural Networks and Hough transform, for use in conjunction with the Bokeh method for mirror alignment on the Medium-Sized Telescopes was evaluated and compared to the currently employed technique of Weighted Circle Fitting.

On a batch of 300 simulated evaluation images the CNN-based approach outperformed all other methods (see Tab. 4.3), achieving a loss towards the true hexagon center of less than five pixels for 95% of the images and only 3% above ten pixels. Meanwhile the Hough transform-based technique only reached a mean accuracy of around 13 px, the worst of all the tested methods, with a loss above five pixels for 82% of the images. However, the CNN approach exhibits a statistically significant bias on the order of one to two pixels in both coordinates that is largely introduced by the required transformation of the 3296 px \times 2472 px-sized images to the network-processable format of 100 px \times 100 px. This transformation is not necessary for Hough transform and Weighted Circle Fitting, although WCF employs a down-sampling algorithm as well. The bias can be reduced by one order of magnitude by slightly modifying the image transformation based on empirical observations. This modified setup achieves a loss of less than one pixel for 74% of the simulated mirror facets. The remaining loss of the network can be reduced via two, non mutually exclusive, approaches. For one, the uncertainty introduced by the down-sampling can be decreased by training a network that is able to process images with a higher resolution. This is primarily a question of available hardware with both required processing power and memory consumption scaling with the total number of image pixels. Further, a CNN with higher accuracy can certainly be trained by using a more elaborate network structure. The training of a better network may also eliminate the remaining inherent bias, that was discovered for this particular network.

Ultimately Convolutional Neural Networks seem to be a promising candidate for Bokeh-based mirror alignment. However, there is an inherent accuracy limit of the Bokeh setup itself, which renders attempts at maximum optimization futile. In the current configuration the Bokeh lamp is placed at a zenith angle of about 5°[19] and likely even less if placed on a neighboring telescope in the final array. For these positions the mechanical stresses on the telescope's structure are not the same as for real observations which are usually performed at elevation angles above 45°, and a good Bokeh alignment will therefore not necessarily translate to a good final alignment.

References

- [1] Rene A. Ong. “Cherenkov Telescope Array: The Next Generation Gamma-ray Observatory”. In: 35th Int. Cosmic Ray Conf - ICRC2017. (July 2017). The CTA Consortium. arXiv: 1709.05434 [astro-ph.HE].
- [2] Markus Garczarczyk. *Medium-Sized Telescope Technical Design Report*. MST-TDR/140415. Version 1.0. CTA Construction Project, May 3, 2015.
- [3] Sebastian A. Mueller et al. “Bokeh Mirror Alignment for Cherenkov Telescopes”. In: *Astroparticle Physics* 82 (Sept. 2016), pp. 1–9. DOI: 10.1016/j.astropartphys.2016.05.003. arXiv: 1605.06277 [astro-ph.IM].
- [4] Louise Oakes et al. “Techniques And Results For The Calibration Of The MST Prototype For The Cherenkov Telescope Array”. In: 6th International Symposium on High Energy Gamma-Ray Astronomy (Gamma2016). (July 2016). The CTA Consortium. DOI: 10.1063/1.4969032. arXiv: 1610.03347 [astro-ph.IM].
- [5] Stephan Kaphle. “Mirror Alignment on the Medium-Sized Telescope Prototype for the Cherenkov Telescope Array”. Master’s Thesis. Humboldt-Universität zu Berlin, Apr. 4, 2017.
- [6] *H.E.S.S. - The High Energy Stereoscopic System*. URL: <https://www.mpi-hd.mpg.de/hfm/HESS/>.
- [7] *MAGIC*. URL: <https://magic.mpp.mpg.de/>.
- [8] *VERITAS*. URL: <https://veritas.sao.arizona.edu/>.
- [9] *COMPTEL*. URL: <https://heasarc.gsfc.nasa.gov/docs/cgro/cgro/comptel.html>.
- [10] *INTEGRAL*. URL: <http://sci.esa.int/integral/>.
- [11] *AGILE*. URL: <http://agile.rm.iasf.cnr.it/>.
- [12] *FERMI*. URL: <https://fermi.gsfc.nasa.gov/>.
- [13] Salvatore Mangano. *Cherenkov Telescope Array Status Report*. The CTA Consortium, May 22, 2017. arXiv: 1705.07805 [astro-ph.HE].
- [14] *Milagro*. URL: <http://umdggrb.umd.edu/cosmic/milagro.html>.
- [15] *Tibet AS-gamma*. URL: <http://www.icrr.u-tokyo.ac.jp/em/index.html>.
- [16] *ARGO-YBJ*. URL: <http://argo.na.infn.it/>.
- [17] *HAWK - The High-Altitude Water Cherenkov Gamma-Ray Observatory*. URL: <https://www.hawc-observatory.org/>.
- [18] B. S. Acharya et al. “Introducing the CTA concept”. In: *Astroparticle Physics* 43 (Mar. 2013), pp. 3–18. DOI: 10.1016/j.astropartphys.2013.01.007.
- [19] Thomas Murach et al. “Automatic Mirror Alignment for the Medium-Sized Telescopes of the Cherenkov Telescope Array Using the Bokeh Method”.
- [20] The CTA Consortium. “Design Concepts for the Cherenkov Telescope Array”. In: *Experimental Astronomy* 32 (3 Dec. 2011), pp. 193–316. DOI: 10.1007/s10686-011-9247-0. arXiv: 1008.3703 [astro-ph.IM].
- [21] Gareth Hughes. *Ray-Tracing the ANL-CEA-DESY Medium Size Telescope*. Tech. rep. DOC-RTMST 2.0.0. DESY Zeuthen, Sept. 19, 2012.

- [22] *Image Tranforms - Hough Transform*. URL: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>.
- [23] *r,theta line parametrization*. URL: https://commons.wikimedia.org/wiki/File:R_theta_line.GIF.
- [24] *Canny-Algorithmus (German)*. URL: <https://de.wikipedia.org/wiki/Canny-Algorithmus>.
- [25] *ROOT a Data Analysis Framework*. URL: <https://root.cern.ch/>.
- [26] *A Beginner's Guide To Understanding Convolutional Neural Networks*. URL: <https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>.
- [27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: <http://www.deeplearningbook.org>.
- [28] *PyTorch*. URL: pytorch.org.