Image Based Heliostat-Calibration in Solar Tower Power Plants with Differentiable Kinematics and Artificial Intelligence

Bildbasierte Heliostat-Kalibrierung in Solarturmkraftwerken mit Differenzierbarer Kinematik und Künstlicher Intelligenz

Der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen vorgelegte Abschlussarbeit zur Erlangung des akademischen Grades "Master Of Science"

vorgelegt von
Tristan Alexander Wortberg
440258

Name of 1st Examiner: Univ.-Prof. Dr.-Ing. Robert Pitz-Paal
Name of 2nd Examiner: Univ.-Prof. Dr.-Ing. Bernhard Hoffschmidt

Name of 1st Scientific Supervisor: Dr.-Ing. Max Pargmann
Name of 2nd Scientific Supervisor: M.Sc. Jan Lewen

Aachen, August 8, 2025

Kurzdarstellung

Die Transformation des Energiesektors hin zu einer klimaneutralen Energiebereitstellung bedingt erhebliche technische und wirtschaftliche Herausforderungen. Solarthermische Turmkraftwerke (ST) gelten als vielversprechende Technologieoption, da sie nicht nur Strom erzeugen, sondern auch industrielle Prozesswärme bereitstellen können. Im Gegensatz zu Photovoltaik oder Windkraft verfügen sie über die Möglichkeit, thermische Speicher zu integrieren, wodurch eine bedarfsgerechte Strombereitstellung ermöglicht wird. Eine zentrale technische Hürde stellt jedoch die präzise Nachführung der Heliostate dar, da selbst geringe Abweichungen zu signifikanten Energieverlusten und Kostennachteilen führen können.

Die vorliegende Arbeit befasst sich mit der Entwicklung eines neuartigen Verfahrens zur Kalibrierung von Heliostaten, das auf der Analyse von Flussbildern basiert. Ziel ist es, strukturelle Merkmale in den gemessenen Flussverteilungen zu identifizieren und zur Optimierung der Spiegelstellung zu nutzen. Dabei wird insbesondere die obere Kontur des Lichtflecks als stabiler Referenzwert herangezogen, da sie im Vergleich zum gesamten Flussprofil weniger durch Verschattung (durch Blockierung) benachbarter Heliostate beeinflusst werden kann.

Im Zentrum der Arbeit steht die Konzeption eines differenzierbaren Optimierungsverfahrens, das auf Bildmerkmalen basiert. Die Methode wird auf synthetisch erzeugten Datensätzen getestet, die gezielt Block- und Verschattungseffekte benachbarter Heliostate simulieren. Diese kontrollierten Bedingungen ermöglichen eine objektive Bewertung der Kalibrierungsgenauigkeit unter realitätsnahen Bedingungen. Die entwickelte Loss-Funktion kombiniert verschiedene Ansätze der Bildauswertung und wird mithilfe einer Bayesian Optimierung auf den gegebenen Anwendungsfall optimiert.

Die Ergebnisse zeigen, dass das vorgeschlagene Verfahren auch unter extremer Verschattung eine robuste und präzise Kalibrierung ermöglicht. Im Vergleich zu klassischen Verfahren, die auf dem Schwerpunkt (Centroid) des Lichtflecks basieren, erzielt die bildbasierte Methode in allen getesteten Szenarien eine höhere Genauigkeit. Insgesamt liefert diese Arbeit einen Beitrag zur Entwicklung zukünftiger Kalibrierungssysteme, die die automatisierte Erkennung höherwertiger, strukturell signifikanter Flussmerkmale integrieren — und damit den Weg für robuste und skalierbare Lösungen in industriellen ST-Anwendungen ebnen.

Abstract

The transformation toward a climate-neutral energy system presents significant technical challenges. Solar tower (ST) power plants represent a promising solution, offering not only electricity generation but also industrial process heat. Unlike wind or photovoltaic systems, they can be equipped with thermal energy storage, enabling dispatchable power delivery. However, one of the key technological limitations is the accurate tracking of heliostats, since even small misalignments can lead to considerable optical energy losses.

This thesis investigates a novel approach to heliostat calibration based on computer-automated analysis of flux measurements. The central idea is to extract structural features from measured flux images and use them to guide alignment optimization. In particular, the upper contour of the focal spot is identified as a robust alignment feature, as it is less affected by shading and blocking by neighboring heliostats than the typically used flux centroid.

The core contribution of this work is the design of a differentiable optimization framework that operates directly on image-based features. The method is tested using synthetic datasets that include controlled scenarios of partial flux obstruction, allowing for systematic validation under realistic distortions. A custom loss function is developed, integrating multiple alignment objectives and tuned through Bayesian optimization.

Results demonstrate that the image-based method yields reliable and accurate calibration performance – even in the presence of severe occlusions. Compared to conventional centroid-based strategies, the proposed approach consistently achieves higher alignment precision. Overall, this thesis lays a conceptual foundation for future calibration systems that incorporate the automated detection of higher-order, structurally meaningful flux features—paving the way for robust and scalable solutions in industrial ST-applications.

Contents

Abstract **IV**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Objective . . . 2

2 Theoretical Foundation and State-of-the-Art **3**
2.1 Solar Tower Power Plants . . . 3
2.2 Heliostat Fields . . . 3
2.2.1 Surface and Optical Properties . . . 4
2.2.2 Coordinate System . . . 4
2.2.3 Heliostat Tracking . . . 5
2.2.4 Evaluation of the Tracking Error . . . 6
2.3 Heliostat Calibration . . . 7
2.3.1 Review of Methods . . . 8
2.3.2 Limitations of Present Calibration Methods . . . 9
2.3.3 Fundamentals of Machine Learning . . . 10
2.3.4 Machine Learning in Calibration . . . 11
2.4 Heliostat Simulation . . . 12
2.4.1 Key Parameters . . . 12
2.4.2 Foundations of the Kinematic Model . . . 13
2.4.3 Differentiable Kinematics Theory . . . 15
2.4.4 Differentiable Ray Tracing . . . 17
2.4.5 Image-Based Learning . . . 19
2.4.6 Summary and Research Question . . . 19

3 Calibration Data **20**
3.1 Motivation for Simulated Datasets . . . 21
3.2 Heliostat Selection . . . 22
3.3 Sun Position Splits . . . 24
3.4 Image Preparation . . . 25

4 Methodology **26**
4.1 Blocking and Shading . . . 26
4.1.1 Implementation through Image Augmentation . . . 26
4.1.2 Geometric Model for Realistic Evaluation . . . 29

4.2 Machine Learning Framework 33
4.2.1 Model Structure 34
4.2.2 The Alignment-Loss 34
4.2.3 Contour Extraction 35
4.2.4 The Contour-Loss 37
4.2.5 Combination of Loss Terms 40

5 Experimental Setup **42**
5.1 Pipeline for Generating Simulated Data with Blocking 42
5.2 Pipeline for Training and Calibration 45

6 Results and Discussion **51**
6.1 Evaluation of the Simulated Scenario-Datasets 51
6.2 Observations on Loss Curves 54
6.3 Accuracy of Final Model Predictions 56

7 Conclusion and Outlook **64**
7.1 Summary of Results 64
7.2 Recommendations for Future Research 65

Acronyms **67**

References **72**

A. Kinematic Parameters Table **73**

1 Introduction

In the year 2024 global energy demand increased by 2.2.%, with emerging economies accounting for over 80% of the observed growth. In the same period, 1.100 TWh were added to global electricity production, constituting an even higher growth rate of 4.3%. [1] Simultaneously, many nations have confirmed their commitments to ambitious emissions reduction targets under frameworks such as the Paris Agreement [2]. Germany, the world's fourth-largest economy, is still committed to reaching climate neutrality by 2045 as outlined in its national climate strategy [3]. In view of humanity's hunger for reliable, cost-efficient and low-carbon energy, considerable additions to energy production capacity are needed. Sustainable Solar Tower (ST) technology is an essential part in global efforts for realizing global emission reduction targets [4]. One of the major challenges in modern energy systems is the integration of variable renewable sources such as wind and photovoltaics, which depend heavily on weather conditions and underlie seasonal fluctuations in their power output. In this context, Concentrating Solar Power (CSP) plants are increasingly investigated for their potential to provide dispatchable electricity. Unlike intermittent sources, CSP systems can be equipped with thermal storage, allowing for flexible electricity generation even in the absence of direct sunlight [4]. Additionally, CSP technology offers a viable option for carbon-neutral fuel production [5].

1.1 Motivation

Despite its advantages, considerable improvements in ST technology are still needed for it to become a market-competitive source of both electricity and industrial heat. The "Roadmap to Advance Heliostat Technologies for Concentrating Solar-Thermal Power" [6], published by the U.S. National Renewable Energy Laboratory (NREL), identifies inaccurate heliostat tracking as one of the primary contributors to optical losses in central receiver systems. Spillage losses caused by imperfect alignment of heliostat beams with the receiver can account for as much as 10˘20% of the total reflected solar energy. If such losses were reduced through more accurate tracking, this could translate into a proportional improvement in system efficiency and a corresponding reduction in the Levelized Cost of Electricity (LCOE). Under these consideration, heliostat calibration emerges as a critical technological lever, enabling more precise control over individual mirror alignment and thereby contributing directly to economic viability of CSP plants [7].
The past few years showed a steady improvement of the accuracy of data-driven technologies applied to the task of heliostat calibration [8, 9, 10]. These methods employ

automated learning methods to infer kinematic correction parameters from flux measurements which are typically obtained through the camera-target method [9]. These advanced calibration methods however, suffer from inaccurate training data, when the centroid position does not coincide with the heliostat's reflection axis. Such discrepancies arise from geometric deviations of the mirror surface from its ideal shape or from blocking and shading effects, which result in partial flux loss and distorted image features [7]. The concern for the reliability of camera-target methods appears justified in light of a recent publication by Pargmann et al. (2023), wherein the researches raised legitimate doubts about the reported accuracies of previously published tracking algorithms [11].

1.2 Objective

Facilitated by the scalable implementation of differentiable ray tracing in simulation environments operable on high-performance computing infrastructure, it becomes feasible to explore optimization strategies based on spatial and structural features of heliostat focal spots. To the best of the author's knowledge, the experiments presented in this work represent the first systematic attempt to develop an image-based approach to heliostat calibration.
The overarching objective of this thesis is therefore formulated as follows: To develop and evaluate an algorithm that derives a differentiable loss function from image-based flux features, suitable for guiding the alignment of heliostats. Feature selection shall prioritize robustness and reliability, even under conditions of partial flux loss caused by blocking or shading. The extracted features must yield a meaningful loss landscape that enables gradient-based optimization to converge toward effective alignment with minimal manual intervention.
To validate this framework, synthetic datasets shall be generated that permit full control over the geometric configuration and include configurable levels of flux degradation due to artificial blocking and shading. This allows the definition of reliable ground-truth references – such as the reflection axis – and facilitates the systematic testing of image-based alignment under varying optical conditions. In support of this, the thesis also introduces methods for simulating partial visibility and flux attenuation in a controlled and reproducible manner.

Finally, the performance of the proposed image-based calibration method will be evaluated across multiple degradation scenarios and benchmarked against conventional centroid-based calibration strategies. This comparative analysis will determine whether structural image features offer a viable alternative to standard flux centroid alignment, particularly in the presence of occlusion and measurement noise.

2 Theoretical Foundation and State-of-the-Art

This thesis explores an interdisciplinary research area situated at the intersection of heliostat control in ST power plants and modern Machine Learning (ML) techniques. To establish a common foundation, this chapter begins by introducing the fundamental principles, terminology, and technical challenges associated with each of these domains.

2.1 Solar Tower Power Plants

ST power plants use the principles of concentrating sun light by optical devices such as lenses or mirrors. The concentrated radiant energy is absorbed as thermal energy at the receiver and subsequently used to generate steam, which finally drives a turbine/generator-system. Alternatively, the thermal energy can directly be used as a heat source - for example in chemical or metallurgic processes. [12]

The type of collector system which is relevant for this thesis is the heliostat field. A heliostat refers to an apparatus of movable mirrors that relies on dual-axis tracking to reflect the incoming sun rays toward a fixed target area, located on the central receiver tower. [13]

2.2 Heliostat Fields

An example photo of the operational heliostat field at the Solar Tower Jülich (STJ) research and demonstrator plant can be found in figure 2.1.

A heliostat field contains a few hundred to several thousand heliostats setup around the central receiver tower. The optimal layout of these installations is an issue of ongoing scientific research. A single heliostat consists of individual mirror facets which are orientated in relation to each other along a strategy known as canting. The reflective surface is mounted on a common backing structure. In operation the reflection direction of this optical appliance is controlled to achieve sun tracking by two motors - each responsible for rotation around one of two axes. [15]

Both in design and operation the heliostat field is required to address conflicting objectives: While land use and the associated costs for installation and maintenance

Figure 2.1: Photo of the heliostat mirrors with four facets at the Solar Tower Jülich. [14]

should be minimized, the field must reliably deliver the required solar flux to the receiver with high efficiency and minimal spillage. [15]
Accurate sun tracking through proper heliostat control is one central requirement for fulfilling these performance targets.

2.2.1 Surface and Optical Properties

Reflectivity and optical quality of mirror surfaces directly determine the proportion of solar radiation that can be effectively redirected towards the central receiver. Typically, heliostat mirrors achieve reflectivity values of approximately 90-95%, but real-world optical efficiency can be lower due to surface imperfections such as micro-scale roughness, coating degradation, or dust accumulation [12]. Surface defects, which include macroscopic shape deviations and microscopic roughness, substantially influence the spatial distribution of reflected flux. These imperfections lead to angular dispersion, beam widening, and reductions in peak flux intensity on the receiver [15]. In practical operations, mirror errors such as facet misalignment, bending at facet edges, and surface waviness critically affect the flux distribution and thus, the efficiency of energy conversion in ST plants [16].

Deviations of the surface normals from their ideal orientation profile are quantified in terms slope errors. As a critical optical metric, the slope error determines the angular deviation of reflected rays and leads to the broadening and distortion of the concentrated flux [16].

2.2.2 Coordinate System

For purposes of geometric computations this thesis relies on the local East-North-Up Coordinate System (ENU), a right-handed Cartesian frame commonly used in solar tower modeling ([17, 18, 7, 9]). The ENU system aligns the x-axis with East, the y-axis with North, and the z-axis with the vertical Up direction. All coordinates are referenced relative to the base of the central solar tower, which serves as the

global origin. The fourth dimension allows rigid-body transformations (cf. 2.4.2, wherein a rotation matrix $\mathbf{R}$ and a translation vector $\vec{t}$ are applied simultaneously. A transformation $T \in \mathrm{SE}(3)$ is thus defined as:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \vec{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad \text{with} \quad \mathbf{R} \in \mathrm{SO}(3), \quad \vec{t} \in \mathbb{R}^3 \tag{2.1}$$

2.2.3 Heliostat Tracking

The sun's apparent position in the sky, defined by its azimuth α_S and elevation θ_S, is a time-dependent function. The incident ray direction $\vec{S}$ for a given sun position can be computed with equation 2.2 [19]. As is defined in ARTIST ray tracing computations, the incident's north coordinate is connoted with a negative sign.

$$\vec{s} = \begin{bmatrix} \cos\theta_\mathrm{S}(t) \cdot \sin\alpha_\mathrm{S}(t) \\ -\cos\theta_\mathrm{S}(t) \cdot \cos\alpha_\mathrm{S}(t) \\ \sin\theta_\mathrm{S}(t) \end{bmatrix} \tag{2.2}$$

Of interest is the desired concentrator normal $\vec{n_\mathrm{C}}$ which is a simple function of the aim point vector $\vec{a}$ and the present sun vector [9]:

$$\vec{n}_\mathrm{C} = \frac{1}{2} \cdot \left(\frac{\vec{s}}{\|\vec{s}\|} + \frac{\vec{a}}{\|\vec{a}\|} \right) \tag{2.3}$$

The mechanical behavior of the heliostat tracking mechanism is modeled under the assumption of rigid-body kinematics. The framework is suitable for the geometric analysis of interconnected bodies and their relative motions under the assumption that these bodies are perfectly rigid, meaning that time- and force-dependent deformations are neglected. This approach models the positions and orientations of objects by computation of geometric transformations, wherein each joint serves as origin of a new coordinate system. [20]

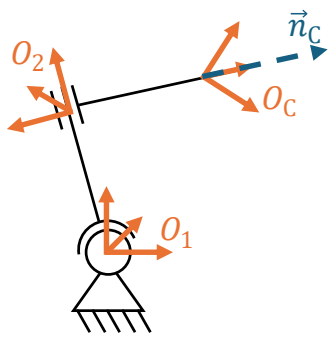


Figure 2.2: Coordinate systems in the rigid-body kinematic model. Adopted from [9].

Figure 2.2 visualizes the chained coordinate transformations that are involved in a dual-axis heliostat tracking system. O_1 and O_2 originate at the first and second actuator joints, respectively. Coordinate system O_C originates in the third joint linking the second actuator's strut with the concentrator.

2.2.4 Evaluation of the Tracking Error

In order to evaluate the alignment quality of a calibrated heliostat, this study relies on the tracking error as the metric that compares the predicted and reference reflection vector. Central to this comparison is the notion of a flux centroid, which defines the center-point of a flux distribution and serves as one of the two endpoints required to compute the measurements reflection vector. However, depending on the method of determining this flux centroid, different versions of the reflection vector arise.

The **tracking error** is defined as the angular deviation between the predicted vector $\vec{r}_\mathrm{P}$ and the measurement's ground-truth (or target) $\vec{r}_\mathrm{G}$. It is computed as the angle θ_track between two normalized vectors using the inverse cosine of their dot product:

$$\text{Tracking Error: } \theta_\mathrm{track} = \arccos\left(\frac{\vec{r}_\mathrm{P} \cdot \vec{r}_\mathrm{G}}{\|\vec{r}_\mathrm{P}\| \cdot \|\vec{r}_\mathrm{G}\|}\right) \tag{2.4}$$

Generally, the direction of reflection for a flux measurement can be approximated by drawing the vector between a flux centroid $\dot{\mathrm{C}}$ and the heliostat surface midpoint $\mathrm{P}_{\mathrm{C},0}$:

$$\text{Approximated Reflection Vector: } \vec{r} \approx \dot{\mathrm{C}} - \mathrm{P}_{\mathrm{C},0} \tag{2.5}$$

However, in practical scenarios involving field measurements, the reference centroid $\dot{\mathrm{C}}_\mathrm{COM}$ is derived from the recorded flux image $I(i,j)$ by first computing its Center of Mass (COM) pixel indices:

$$\text{COM Pixel Indices: } (i_\mathrm{com}, j_\mathrm{com}) = \frac{\sum_{i,j} I(i,j) \cdot (i,j)}{\sum I(i,j)} \tag{2.6}$$

In a subsequent step, the centroid's local ENU-coordinates are computed, where $[\mathrm{H}, \mathrm{W}]$ are the image dimensions, $\mathrm{P}_{\mathrm{T},0}$ is the target area's center coordinates, and $\Delta\mathrm{e}_\mathrm{T}$ [m] and $\Delta\mathrm{u}_\mathrm{T}$ [m] its dimensions in the east and up direction:

$$\text{COM Position: } \dot{\mathrm{C}}_\mathrm{COM} = \mathrm{P}_{\mathrm{T},0} + \begin{bmatrix} \left(\frac{j_\mathrm{COM}}{\mathrm{H}} - 0.5\right) \cdot \Delta\mathrm{e}_\mathrm{T} \\ 0 \\ \left(\frac{i_\mathrm{COM}}{\mathrm{W}} - 0.5\right) \cdot \Delta\mathrm{u}_\mathrm{T} \end{bmatrix} \tag{2.7}$$

This mapping assumes that the target surface is orthogonal to the north axis, such that all points in the target plane P_T :

$$\text{Target Plane: } (\mathrm{P}_{\mathrm{T},} - \mathrm{P}_{\mathrm{T},0}) \cdot \vec{n}_\mathrm{T} = 0 \tag{2.8}$$

$$\text{with Target Plane Normal: } \vec{n}_{\mathrm{T},} = [0, 1, 0]^T \tag{2.9}$$

This method, while straightforward, results in what is referred to here as a **non-ideal centroid**. The position of the non-ideal centroid deviates from the reflection axis due to a variety of factors such as atmospheric scattering, mirror deformation, and partial occlusions [7]. As a consequence, the reflection vector built from such centroids does not correspond to the actual geometric reflection direction of the heliostat and should therefore be regarded as an approximation:

$$\text{COM-Vector: } \vec{r}_{\text{COM}} = \dot{\text{C}}_{\text{COM}} - \text{P}_{\text{h,0}} \approx \vec{r} \tag{2.10}$$

When model calibration is attempted in the simulation environment of ARTIST, the fit of kinematic parameters is evaluated on the **accuracy on COM-vector** which approximates the tracking error metric:

$$\text{Accuracy on COM-Vector: } \theta_{\text{COM}} = \arccos\left(\frac{\vec{r}_{\text{P}} \cdot \vec{r}_{\text{G,COM}}}{\|\vec{r}_{\text{P}}\| \cdot \|\vec{r}_{\text{G,COM}}\|}\right) \tag{2.11}$$

In contrast, the ideal centroid $\dot{\text{C}}_{\text{ideal}}$ is defined as the intersection point of the actual reflection axis along $\vec{r}$ with the target plane:

$$\text{Ideal Centroid: } \dot{\text{C}}_{\text{ideal}} = \text{P}_{\text{h,0}} + t \cdot \vec{r} \tag{2.12}$$

where t is the intersection distance which can be computed to:

$$\text{Intersection Distance: } t = \frac{\text{P}_{\text{T,0}} - \text{P}_{\text{C,0}} \cdot \vec{n}_{\text{T}}}{\vec{s} \cdot \vec{n}_{\text{T}}} \tag{2.13}$$

The information of the actual reflection axis is unknown field measurement data. During raytracing-based simulation however, the model computes this vector directly from the heliostat's surface orientation $\vec{n}_{\text{C}}$ and the sun's incident ray direction $\vec{s}$ as follows:

$$\text{Reflection Vector: } \vec{r} = \vec{s} - 2 \cdot \vec{n}_{\text{C}}(\vec{s} \times \vec{n}_{\text{C}}) \tag{2.14}$$

2.3 Heliostat Calibration

Accurate heliostat tracking is critical for maximizing solar power plant efficiency and minimizing flux spillage. Even small alignment errors can significantly affect performance: for instance, a tracking error of only 1 mrad results in an aim-point offset of approximately 2 m for a heliostat located 1 km north of the receiver tower-distances which are easily reached in large ST installations [7].

The sources for errors in heliostat tracking are manifold. Significant error sources include atmospheric refraction, gravity bending, azimuth rotational axis tilt, pivot point offset, mirror alignment or canting non-orthogonality relative to the heliostat centerline and errors in mark orientation (tracking mechanism's reference) [21]. Under these considerations it is apparent, that technical measures for reducing and countering sources of tracking errors are needed.

2.3.1 Review of Methods

A comprehensive review of heliostat calibration and tracking control methods was performed by Sattler et al (2020) [7]. All novel methods are competing with the so-called camera-target method (see Figure 2.3), which is established as state-of-the-art since its disclosure by Stone (1986) [22]. In the camera-target method a selected heliostat's focus is shifted from the receiver to a Lambertian target, generating a discrete focal spot that is captured by a fixed camera system on the ground. Image processing software is used for extracting the focal spot's centroid position. In a final step the tracking system model is calibrated by comparison of the centroid position with a reference position.

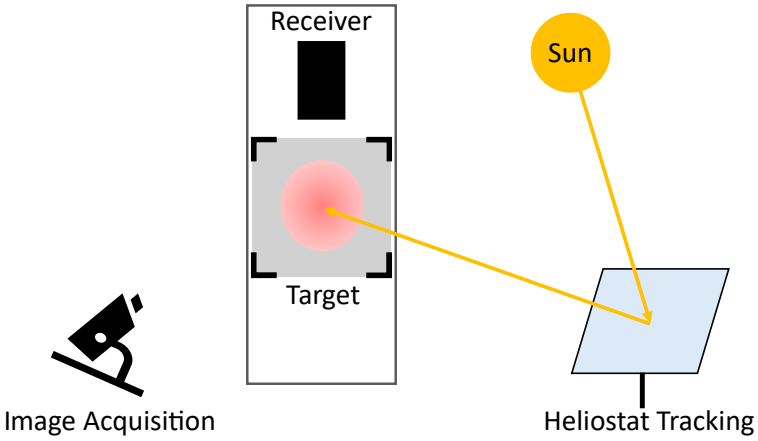


Figure 2.3: Measurement setup for the standard camera-target method. Image acquisition serves to record the flux distribution measurement.

Another class of heliostat calibration methods directly infers the orientation vector from an image recording of the actual heliostat. Herein, the camera device requires an elevated position and can either be mounted on the receiver tower or be carried by an Unmanned Aerial Vehicle (UAV). The proposed method [23] of Röger et al. (2010) applies automatic edge detection on a single image captured by a camera mounted on the receiver tower to infer the heliostat's orientation. Krauth et al. (2024) in [24] disclose an arrangement of a light-emitting diode and a camera, both mounted on a UAV. The orientation vector is deduced from the known positions of the heliostat and of the UAV-camera-LED arrangement.

A third class of measurement methods relies on an interconnected system of multiple cameras or non-optical sensors. Kribus (2004) in [25] claims to have implemented a heliostat closed control-loop based on image acquisition from four positions around the receiver aperture. Analysis of the comparative brightness assigned to individual heliostats enables determination of this heliostat's true orientation. An elaborate approach by Collins et al. (2016) in [26] relies on the same principle but realizes a more

granular measurement based on a movable array of cameras.

Due to the multitude of proposals for calibration methods, this section cannot offer a complete report. Instead it serves to illustrate the active research in the field which points at an ongoing demand for improvements for more accurate heliostat tracking.

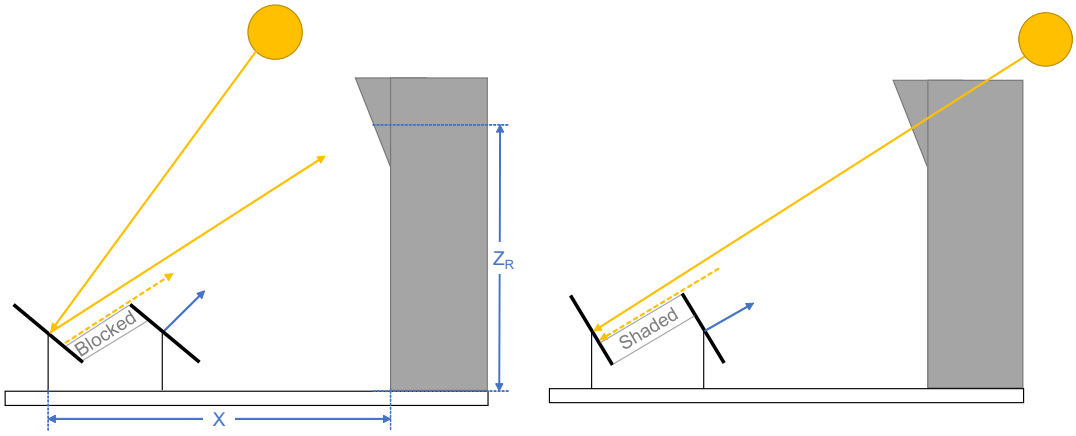
2.3.2 Limitations of Present Calibration Methods

The camera-target method is established as state-of-the-art due to its reliability and cost-effectiveness. It requires few, freely available hardware components of relatively low-technical complexity. Its main draw-backs are its low scalability to large ST. By design, the camera-target method is a sequential method, meaning that heliostats are calibrated one-by-one. For a heliostat field of many thousand heliostats the complete calibration process can take up to several years. [24]

At the same time the method delivers very high accuracies at around 0.1 mrad reported for measuring the heliostat orientation vector. [7] However, a recent study [11] by Pargmann et al. (2025) challenges the reliability of the high tracking accuracies reported in earlier publications on heliostat calibration. The authors argue that the decisive factor influencing model accuracy is not the temporal proximity of measurements, but rather their spatial distribution in terms of solar position. Their analysis demonstrates that calibration data selected based solely on time can lead to overly optimistic results, as test data may unintentionally fall within the interpolation region of the training set. This raises serious concerns regarding the validity of previous findings, especially when test set accuracies were evaluated without accounting for the angular distance between sun positions in the train and test sets.

All calibration algorithms that process image data of either the heliostats directly or of their focal spots on a Lambertian target are sensible to blocking and shading [27]. Figure 2.4 gives a simplified 2D-visualization of both cases of heliostat-to-heliostat blocking and shading. The degradation in plant efficiency due to heliostat-to-heliostat intersection has been studied thoroughly for example in [28], but its influence on tracking accuracy through degradation of the flux measurement signal has, to the authors' knowledge, not yet been quantitatively assessed. Although the existing ARTIST simulation software that is used within this study is sophisticated in modeling the kinematic behavior and mirror surface characteristics in heliostat operation, presently it does not consider optical interference between these mirrors. Figure 2.4 gives a simplified 2D-visualization of both cases of blocking and shading.

A recent publication by Abunajeeb et al. (2025) addresses the issue of measurement uncertainty due to blocking and shading by neighboring heliostats [30]. The approach involves installing curved, specular-reflective bars along the horizontal and vertical edges of each heliostat, whose reflected sunlight spots are observed by a remote camera to infer alignment deviations. While promising, this method remains unvalidated at



(a) Blocking by a front heliostat. **(b)** Shading from a front heliostat.

Figure 2.4: Sketch for heliostat-to-heliostat ray interference and subsequent occlusion in the cases of blocking (a) and shading (b) according to [29, 13]. X denotes the horizontal distance between rear heliostat and receiver, and Z_{R} denotes the vertical height of the receiver surface center.

scale and depends on additional hardware, which may increase costs and complicate retrofitting or deployment in existing ST facilities.

2.3.3 Fundamentals of Machine Learning

ML refers to a class of algorithms that improve their performance on a task through exposure to data rather than through explicit programming. The general problem in context of heliostat calibration is visualized in Figure 2.5, ML is commonly applied as a regression task by gradient-descent, where the goal is to learn parameters relating to the physical system (e.g. offset parameters) from measurement data (calibration data).

Optimization is typically carried out via gradient descent (Backward-Pass), where the model's parameters are iteratively updated in the direction of the loss function's negative gradient, scaled by a user-defined learning rate. The core training mechanism involves minimizing a loss function that quantifies the difference between the model's predicted output and the known ground-truth data along meaningful features [31, 32]. In context of heliostat calibration, a loss function is required that penalizes deviations between the predicted and ground-true reflection axes. Among the possible choices for loss functions in regression and calibration problems, the Mean Absolute Error (MAE) and Mean Squared Error (MSE) are commonly used loss functions:

$$\text{Mean Average Error: } \mathcal{L}_{\mathrm{MAE}} = \frac{1}{N}\sum_{s=1}^{N} |y_{\mathrm{P},s} - y_{\mathrm{G},s}| \tag{2.15}$$

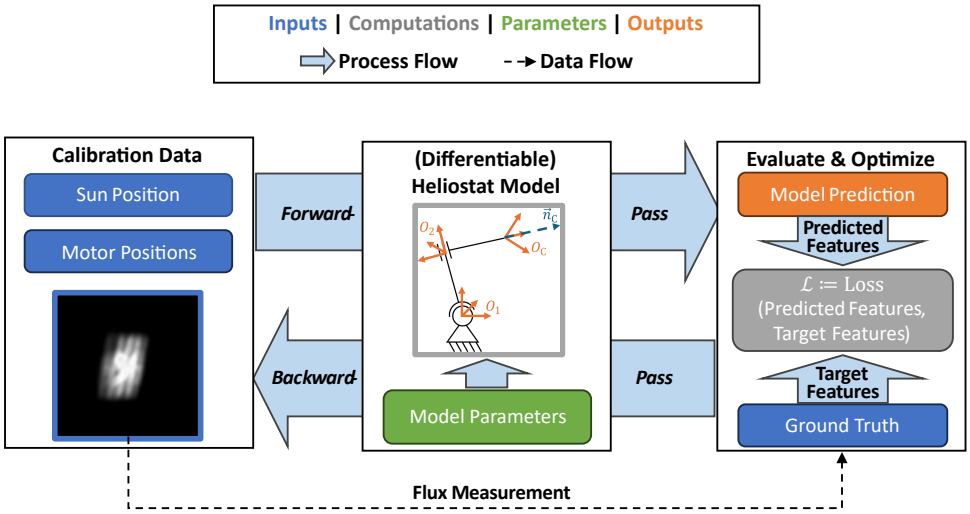


Figure 2.5: Visualization of the abstract machine-learning framework applied to the taks of heliostat calibration. Calibration Data serves as Inputs to the differentiable heliostat model (Forward-Pass). Based on the model prediction and the ground-truth feature mapping is performed to obtain predicted and target features. These are further processed to compute the loss, which is subsequently differentiated (Backward-Pass). Finally, the model parameters are updated.

$$\text{Mean Squared Error: } \mathcal{L}_{\text{MSE}} = \frac{1}{N}\sum_{s=1}^{N}\left(y_{\text{P},s} - y_{\text{G},s}\right)^2 \tag{2.16}$$

Here, $y_{\text{P,s}}$ and $y_{\text{G,s}}$ represent the predicted and ground-true values for a data sample s, respectively, and N is the number of measurement samples.

To evaluate model performance and avoid overfitting, the available dataset is divided into training, validation, and test subsets, with only the training data used for parameter updates, while validation and test data serve to assess generalization. Throughout the training-process, performance on the validation set is monitored. Finally, the model is evaluated on an independent test set to assess its generalizability to unseen scenarios (sun position angles).

2.3.4 Machine Learning in Calibration

In order to increase the accuracy of the standard camera-target method by learning latent relationships in the calibration data, novel data-driven approaches are being developed. The application of data-driven methods to heliostat calibration has seen significant evolution in recent years, moving from black-box learning toward physically informed, hybrid approaches. Pargmann et al. (2021) explored a fully data-driven method in which a deep Artificial Neural Network (ANN) was trained to predict heliostat misalignments from camera-target measurements without explicitly modeling

the system kinematics [8]. Because data acquisition through the camera-target setup is slow, approaches which in operation require large datasets are not unqualified for application to the industry [5].

In response, recent research has shifted toward hybrid modeling techniques that integrate data-driven components with differentiable physical models. Pargmann et al. (2023) and Leibauer et al. (2023) proposed a modular calibration framework that combines a rigid-body kinematic model with an ANN-based disturbance model [5, 9]. This architecture benefits from physics-informed modeling while allowing the learned disturbance-model to capture latent relationships (e.g. wind loading, thermal expansion). These hybrid approaches are more suitable to practical deployment, as they allow for calibration from sparse or noisy data and generalize more reliably [5]. A recently published approach by Sievers et al. (2025) builds upon these advances, where a self-normalizing neural network (SNN) is trained to predict fine-tuning corrections ("disturbance parameters") to the rigid-body kinematic model of the heliostat [10]. Even though the progress of such models is considerable, present approaches are sensitive to false training stimuli due to degraded flux measurements.

2.4 Heliostat Simulation

This thesis relies on simulations rendered by the ARTIST environment which is maintained by researchers at German Center for Aerospace (DLR) and Karlsruhe Institute of Technology (KIT). ARTIST is programmed in PyTorch, facilitating differentiable computations in both the kinematic model and ray tracing simulations.

2.4.1 Key Parameters

Table 2.1 lists key parameters that are used within the simulation environment. By default, all matrix-like computations within this study's simulations are performed through Tensors. In PyTorch, a Tensor is a multi-dimensional array that serves as the fundamental data structure for storing and manipulating numerical data, with optional support for Graphics Processor Unit (GPU) acceleration. Tensors can also carry metadata for automatic differentiation to facilitate gradient computation.

- As each heliostat tracking system relies on two actuators its discrete step motor positions have the format int[2].
- The sun position is given in Euler coordinates (elevation, altitude) in Float-format (cf. Section 2.2.3).
- The flux measurements are processed as Tensor-bitmaps, single-channel grey-scale in the resolution of 256x256.

Table 2.1: Key parameters and their data types.

Name	Data Type
Motor Positions	`Int[2]`
Sun Position	`Float[2]`
Flux Image	`Tensor[256, 256]`
Surface Orientation	`Tensor[4, 4]`
Surface Normals	`Tensor[`$nPoints$`, 4]`
Surface Points	`Tensor[`$nPoints$`, 4]`
Kinematic Parameters	`Tensor[27]`

- The surface orientation Tensor specifies the concentrator's coordinate transformation matrix containing its surface normal and surface origin.
- Concentrator surfaces are modeled in terms of surface points and the corresponding surface normals across four facets each being represented by $nPoints$-number of points.
- Kinematic Parameters: Geometric parameters and deviation parameters. The model considers 27 fundamental parameters which are updated in calibration (cf. Table 1).

2.4.2 Foundations of the Kinematic Model

The kinematic model framework that is presently used by the ARTIST-simulation environment was proposed by Leibauer (2023) [9]. Figure 2.6 displays the kinematic model which is used for heliostat tracking. The variable name to the parameters to each actuator $i \in \{1, 2\}$ are indicated next to the dimension which they are referencing to.

The variable stroke length x_i can be represented as a linear function over the actuator motor position s_i [9]:

$$x_{\mathrm{i}}(s_i) = \lambda_i \cdot s_i + b_i \tag{2.17}$$

Where λ_i denotes the actuator gear ratio and b_i refers to the initial stroke length. Of interest is the angular radius shift e_i around the actuator joint for the adjusted stroke length [9]:

$$e_i(s_i) = \arccos \frac{c_i^2 + d_i^2 - x_i(s_i)^2}{2c_i d_i} + a_i \tag{2.18}$$

Where c_i is the distance offset of the joint point from the actuator origin, d_i is the actuator radius, and a_i is offset in the radius shift.

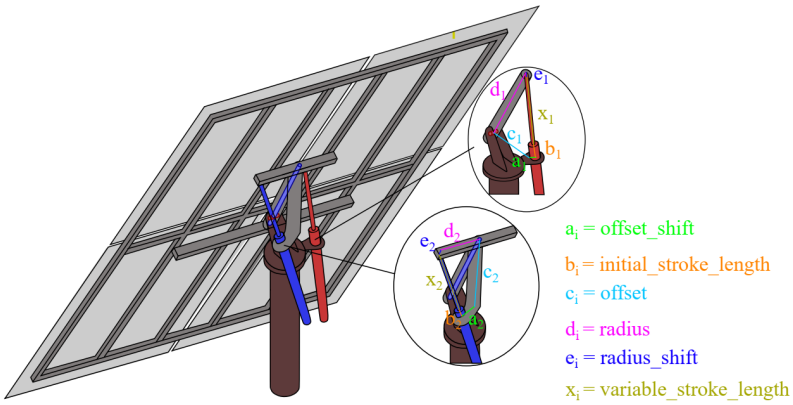


Figure 2.6: Visualization of the kinematic model for heliostat tracking and its parameters. Used at permission of Max Pargmann (DLR).

In the dual-axis model of this study, the first and second actuators are respectively enabling tracking along east- and up-direction. Based on the effective angular radius of joints 1 and 2, the corresponding rotational matrices are computed with equations 2.19 and 2.20 [17]:

$$\mathbf{R}_{\mathrm{e}}(e_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(e_1) & -\sin(e_1) & 0 \\ 0 & \sin(e_1) & \cos(e_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

$$\mathbf{R}_{\mathrm{u}}(e_2) = \begin{bmatrix} \cos(e_2) & -\sin(e_1) & 0 & 0 \\ \sin(e_1) & \cos(e_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

The heliostat surface normal is determined through the three successive coordinate transformations (equations 2.22, 2.23, 2.24 and 2.25) which assume the initial matrix T_0 (equation 2.21) encoding both the rotation $\mathbf{R}_0$ and position P_0. The first $\mathbf{T}_{0,1}$ accounts for the translations and rotations of the joint 1. $\mathbf{T}_{1,2}$ is the full transformation matrix associated with joint 2. $\mathbf{T}_{2,\mathrm{c}}$ denotes the final transformation for reaching the concentrator's origin and orientation. [9]

$$\mathbf{T}_0 = \mathbf{T}_{\mathrm{enu}}(\Delta e_0, \Delta n_0, \Delta u_0) \cdot \begin{bmatrix} \mathbf{R}_0 & \mathrm{P}_0 \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{2.21}$$

$$\mathbf{T}_{0,1} = \mathbf{R}_{\mathrm{n}}(\Delta\varepsilon_{\mathrm{n},1}) \cdot \mathbf{R}_{\mathrm{u}}(\Delta\varepsilon_{\mathrm{u},1}) \cdot \mathbf{T}_{\mathrm{enu}}(\Delta e_1, \Delta n_1, \Delta u_1) \cdot \mathbf{R}_{\mathrm{e}}(e_1) \tag{2.22}$$

$$\mathbf{T}_{1,2} = \mathbf{R}_{\mathrm{e}}(\Delta\varepsilon_{\mathrm{e},2}) \cdot \mathbf{R}_{\mathrm{n}}(\Delta\varepsilon_{\mathrm{n},2}) \cdot \mathbf{T}_{\mathrm{ENU}}(\Delta e_2, \Delta n_2, \Delta u_2) \cdot \mathbf{R}_{\mathrm{u}}(e_2) \tag{2.23}$$

$$\mathbf{T}_{2,\mathrm{c}} = \mathbf{R}_{\mathrm{e}}(\Delta\varepsilon_{\mathrm{e,C}}) \cdot \mathbf{R}_{\mathrm{n}}(\Delta\varepsilon_{\mathrm{n,C}}) \cdot \mathbf{R}_{\mathrm{u}}(\Delta\varepsilon_{\mathrm{u,C}}) \cdot \mathbf{T}_{\mathrm{enu}}(\Delta e_{\mathrm{C}}, \Delta n_{\mathrm{C}}, \Delta u_{\mathrm{C}}) \cdot \tag{2.24}$$

Above geometric transformations already consider a set of geometric error parameters which were first proposed by a publication of Leibauer (2023) [9]. For each mechanical component this parameter set contains tilts $\Delta\varepsilon_x$ and translation errors Δx_i with $x \in \{e, n, u\}$. The comprehensive register of all geometric parameters used within the present kinematic is given in table 1.

$$\mathbf{T}_{\mathrm{C}} = \mathbf{T}_0 \cdot \mathbf{T}_{0,1} \cdot \mathbf{T}_{1,2} \cdot \mathbf{T}_{2,\mathrm{C}} \tag{2.25}$$

Finally, equations 2.26 and 2.27 provide the mappings from the concentrator matrix to the concentrator's point of origin $\mathrm{P}_{\mathrm{C},0}$ and its surface normal vector $\vec{N}_{\mathrm{C}}$.

$$\mathbf{T}_{\mathrm{C}} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{P}_{\mathrm{C},0} \\ 1 \end{bmatrix} \tag{2.26}$$

$$\mathbf{T}_{\mathrm{C}} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \vec{n}_{\mathrm{C}} \\ 0 \end{bmatrix} \tag{2.27}$$

2.4.3 Differentiable Kinematics Theory

The aforementioned kinematic model is implemented in PyTorch, which enables automatic differentiation by dynamically constructing computation graphs during forward passes. Within this framework, users can explicitly define which model parameters are treated as optimization variables and which remain fixed, allowing for flexible configuration of the computational graph. PyTorch automatically records all operations performed on the optimization variables and computes exact gradients via reverse-mode auto-differentiation during backpropagation.

Figure 2.7 displays the integration of the differentiable kinematic model into an algorithm which enables determining the actuator Motor Positions (MP) to a given calibration flux measurement. The algorithm was described in a similar implementation by Leibauer (2023) in [9]. Initially, the two actuator's MP are guessed aligning

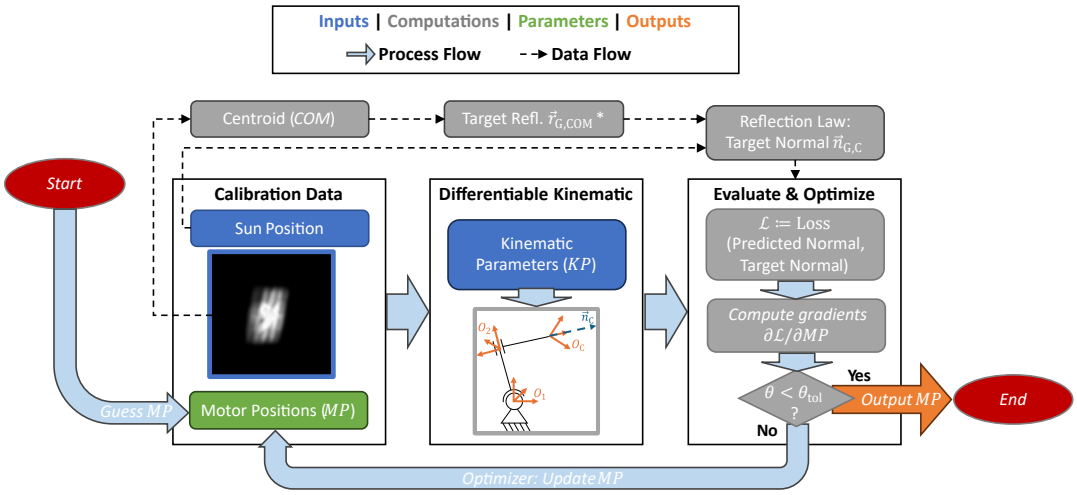


Figure 2.7: The differentiable kinematic model is used for finding actuator motor positions (MP) for a given flux centroid. The target reflection vector is spanned from the flux centroid (cf. equation 2.10) center-of-mass (COM), from which the target normal is derived (cf. equation 2.27). Subsequently, the loss is computed from the error between the kinematic's predicted surface normal and the target normal. As long as the misalignment θ is greater than a predefined tolerance θ_{tol}, MP is optimized by gradient-descent. Inspired by [9].
* The reflection vector is approximated as COM-vector.

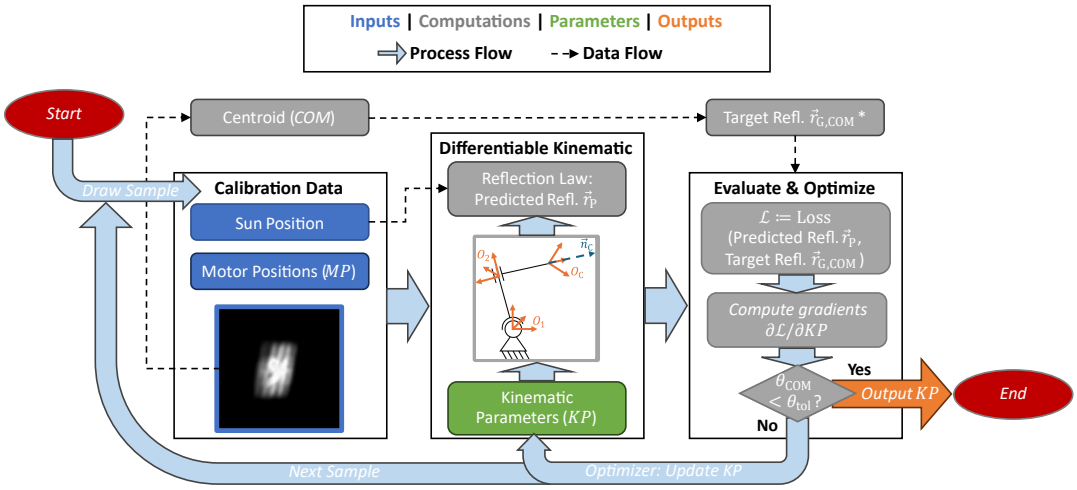


Figure 2.8: The differentiable kinematic model is used for heliostat calibration by optimizing the kinematic parameters (KP) for on sampled calibration data. The reflection vector is computed as the flux center-of-mass (COM), from which the target (desired) heliostat normal is derived (cf. equation 2.27). Subsequently, the loss is computed from the error between the kinematic's predicted and the target reflection vectors. As long as the COM-accuracy θ_{COM} (cf. equation 2.11) is greater than a predefined tolerance θ_{tol}, MP is optimized by gradient-descent.
* The reflection vector is approximated as COM-vector.

the heliostat's kinematic to a random orientation. Via COM method the measurement's centroid position is computed which is used as the desired Aim Point (AP) for spanning the target reflection vector. The desired (target) normal is computed under application of the reflection law (cf. equation 2.14). The loss is derived from the offset between target and kinematic-predicted reflection vectors. Subsequently, the MP are updated along gradient-optimization. The optimization loops until the alignment error θ is below the threshold at θ_{tol}. This algorithm is useful when the actuator motor positions are not known – or assumed to be incorrect.

Figure 2.8 displays the application of the differentiable kinematic model to the task of heliostat calibration by simulation without ray tracing. Samples of calibration data containing the MP information are used for kinematic alignment. Via reflection law the prediction's reflection vector is determined. Deviations of the predicted output from the target reflection vector (COM-based) are penalized. In this ML-task the kinematic's geometric parameters are optimized to accomplish calibration. This algorithm enables heliostat calibration, but it does not yet consider the heliostat's surface geometry and the simulated focal spot. This feature is unlocked by adding ray tracing into the optimization loop.

2.4.4 Differentiable Ray Tracing

Ray tracing is a physically-based simulation technique used to predict the irradiance distribution resulting from light reflection and scattering on optical surfaces. ARTIST's ray tracing simulation – like the kinematic model – is implemented in PyTorch, enabling automatic differentiation. In the context of concentrating ST technology, ray tracing combines specular reflection on the heliostat surfaces with diffuse reflection on a target to simulate the flux distribution on the target's surface. This process requires accurate knowledge of the heliostat's surface deviations from the mirror's ideal geometry, as each ray's reflection is determined by the local surface normal and position of incidence [27].

Mirror surface deviations can in principle be derived from fringe pattern deflectometry measurements [33, 34, 35]. Despite its high accuracy, deflectometry remains challenging to deploy in heliostat fields due to its sensitivity to environmental conditions and surface contamination (e.g., dew or dust). Additionally, its applicability is limited by procedural constraints, particularly its incompatibility with in-situ and daytime operation [5]. The surface model which is used within this study is constructed based on such deflectometry measurements which are available open-source for the STJ under [36].

Figure 2.9 displays the calibration approach introduced in figure 2.8 which is extended with an simulation module for differentiable ray tracing. With this step the heliostat model generates a prediction for the flux distribution. Based on the flux output, kinematic alignment can be evaluated on its predicted centroid position derived by

COM-method. In this automated calibration pipeline the loss is computed from the error between the target and predicted COM-vectors. This data-driven calibration framework is the present state-of-the-art facilitated by the ARTIST simulation environment.

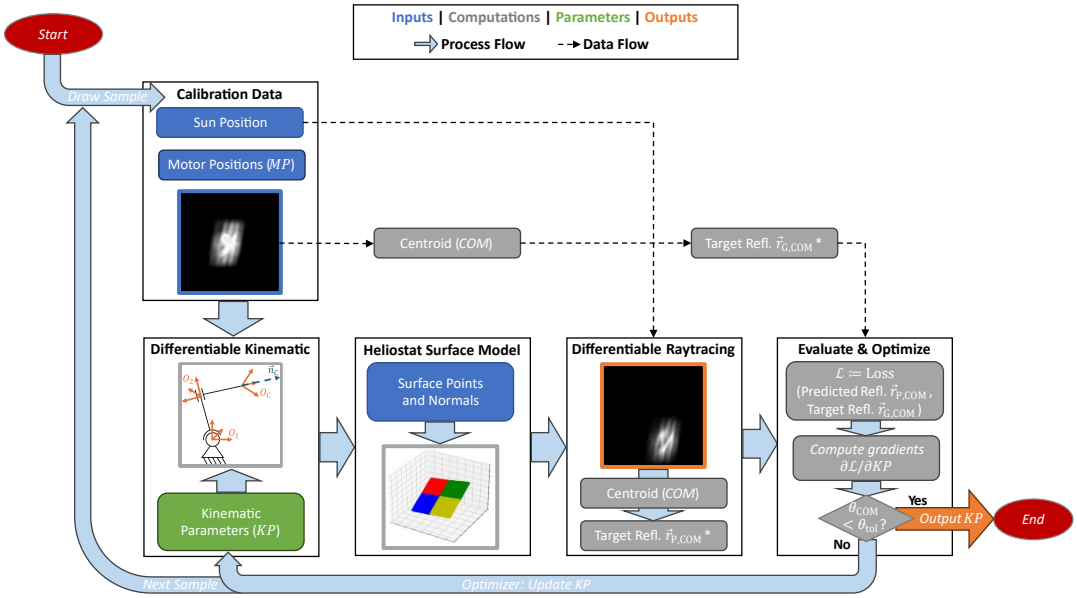


Figure 2.9: Differentiable kinematics and differentiable ray tracing are combined realizing fully differentiable heliostat simulation to facilitate data-driven calibration. The ground-true (target) reflection vector is computed as the flux center-of-mass (COM), from which the target (desired) heliostat normal is derived (cf. equation 2.27). Subsequently, the loss is computed from the offset in predicted and target reflection vectors. Optimization of the kinematic parameters by gradient-descent loops while the accuracy θ_{COM} is greater than a predefined tolerance θ_{tol}.
* The reflection vector is approximated as COM-vector.

Recent success of gradient-based optimization procedures in differentiable ray tracing simulation environments enables heliostat surface reconstruction using flux image data only [5, 27, 16]. While these developments mark significant progress toward software-only surface prediction, challenges remain due to the ill-posed nature of the inverse problem, particularly for heliostats with limited flux recordings or those situated in proximity to the receiver tower. Lewen et al. (2024) introduce inverse Deep Learning Ray Tracing (iDLR), a supervised learning pipeline that leverages synthetic training data generated via forward ray tracing to learn a mapping from flux intensity distributions to corresponding surface geometries [16]. This class of methods could offer a cost-effective, scalable and hardware-independent alternative to deflectometry in the near future.

2.4.5 Image-Based Learning

Relevant to this thesis are image-based loss functions which allow the evaluation of two input images along higher-order features. In the context of heliostat calibration the extraction of spatially and structurally meaning features from a flux measurement are of particular relevance. Fundamental to the task of feature extraction is the convolutional operation ($*$), which involves sliding a small matrix (the kernel) across a given image and computing a weighted sum at each location. Mathematically, the convolution of an image I with a kernel K is expressed as [37]:

$$\text{Convolution: } S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n) \tag{2.28}$$

where (i,j) is the output (feature) map, (i,j) are the pixel coordinates, and (m,n) index the kernel entries.

Of particular relevance for this study is the Sobel kernel, which is widely recognized in the scientific literature for its effectiveness in detecting boundary features. In [38] a Sobel kernel is used for automated sunspot detection. The specific Sobel kernel used for upper edge detection is specified as:

$$\text{Vertical Sobel: } K_{\text{Sobel},y}(m,n) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.29}$$

2.4.6 Summary and Research Question

Due to the time requirement of the camera-target method, any calibration algorithm building on such measurement setup must prioritize sample efficiency – that is, achieving accurate calibration performance with minimal data. The results from both [39, 11] suggest performing a stratified dataset split using the proposed distance metric in the sun-position space. Importantly, measurements samples at low sun elevation angles, can be linked to an increased frequency and severity of blocking and shading effects [28].
The present research aims to develop a novel image-based calibration algorithm that maintains high accuracies even in the presence of blocked and shaded measurements. As a feature for such alignment the upper contour of the recorded focal spots is chosen, because it is less affected by blocking and shading ([28, 30]). Importantly, this algorithm is applicable to readily available flux image data as obtained in camera-target measurement setups. Such systems are already in place at most commercial and research solar tower facilities, and their data acquisition processes are well established [7]. Thus, if it can be demonstrated that features extracted from flux images enable robustness against partial occlusions from blocking and shading, this work may contribute to a conceptual shift in heliostat calibration: from traditional vector-based alignment methods to image-based learning approaches.

3 Calibration Data

The calibration data processed within this study's simulations was sourced directly from camera-target measurement campaigns conducted at the research power plant STJ, which is operational since 2009. The STJ is a large-scale research facility operated by the DLR and associated partners, consisting of over 2000 heliostats that concentrate solar radiation onto a 60-meter-high tower equipped with a high-temperature volumetric air receiver. The tower serves as a platform for investigating solar thermal power generation and supports research activities focused on improving system efficiency and control strategies.

Measurement campaigns at this facility are carried out by moving each heliostat's aim point to one of three Lambertian calibration targets. Two of these targets are mounted under the primary receiver on the "Solar Tower Jülich" and one additional target is mounted on the "Multifoucs Tower", see Figure 3.1.

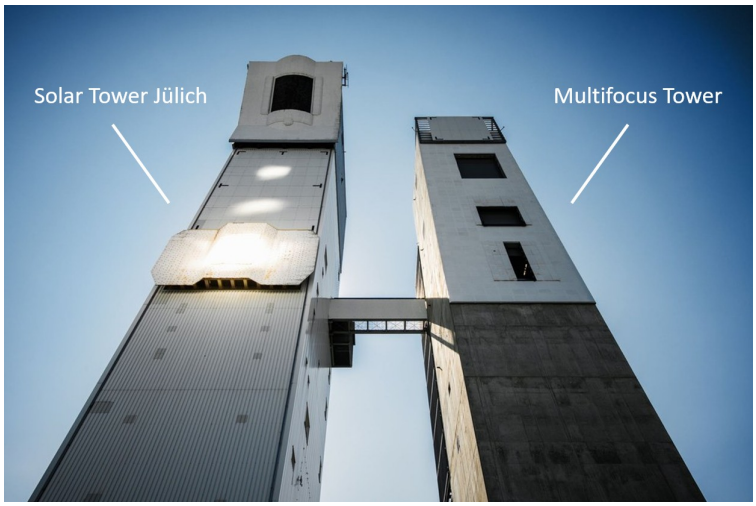


Figure 3.1: The two tower installations at STJ which are used for calibration measurements. [14]

A selection of calibration data gathered at STJ is available online on the PAINT database which is run by the ARTIST Association and maintained by the DLR and KIT [36]. For the use of this study, three of the repository's data formats are processed. These data types are described in Table 3.1.

Table 3.1: Overview of three data formats available in the PAINT repository, which are relevant for this study.

Data Type	Description	Data Format
Calibration	Grey-scale photos of the focal spots of heliostats on dedicated calibration target areas.	PNG
Deflectometry	Data on the assessment of the heliostats' reflective surface deformations. The data was gathered in strip-pattern deflectometry experiments.	HDF5
Kinematic	Data on the mechanical arrangement of heliostats and sun position at the time of photo recording the focused focal spots.	JSON

Grey-scale images of the relative flux intensities are available in 256x256 pixel resolution. These images are the result of processing raw image data within a UNet-based image-to-image neural network model which is trained to remove background radiation [40, 41]. For the subsequent experiments on image-based heliostat calibration, recent measurement samples from the year 2022 to 2024 were processed.

Deflectometry measurements are available for roughly a quarter of the heliostats at STJ (cf.Section 3.2). This measurement source provides the basis for reconstruction the concentrator surfaces within the simulation environment.

The kinematic data formats contain measurements for the mechanical properties of the heliostats, and the sun position measurements.

3.1 Motivation for Simulated Datasets

The ARTIST simulation environment is used to generate a number of simulated datasets, which finally serve to train and evaluate image-based heliostat calibration. This means that the flux "measurements" on which calibration is performed, are generated by ray tracing simulation. Three major reasons motivate the use of simulated datasets:

1. Stable development of image-based loss and testing under idealized conditions.
2. Reliable evaluation: Ground-true reflection axis is known for simulated data.
3. Though image augmentation the effects of blocking and shading can be mimicked. Thereby, the calibration performance can be evaluated at different levels of flux occlusion.

Generally, training on simulated data provides idealized conditions for validating the functionality of the developed image-based loss function, which is sensitive to irregularities in the input images. By removing uncertainties introduced by camera optics,

environmental fluctuations, or annotation artifacts, the learning algorithm can focus on the core task of matching the predicted flux distribution to the measured reference. This proof-of-concept step is essential before extending the model to real-world data domains, where additional filtering and regularization measures may be necessary to account for imperfections in the input.

As section 2.2.4 elaborates, a major advantage of working with simulated data lies in the availability of exact ground truth information. In the case of heliostat calibration, this includes the known orientation of the mirror surface and the resulting reflection axis, from which the coordinate positions of the ideal flux centroid can be determined by projecting the reflection axis onto the target area. In evaluating heliostat alignment, the information of the ideal centroid allows for computation of the actual tracking error, while in real-world scenarios only the approximated error against COM-vector (cf. equation 2.11) is known.

Thirdly, this thesis proposes an image-augmentation method to mimic the degenerative effects on flux measurements, which arises due to heliostat-to-heliostat blocking and shading (cf. Section 4.1.1). Multiple datasets are generated each characterized by different frequencies and severities of flux loss due to blocking / shading. The comparative performance of both conventional and image-based calibration methods can be evaluated along this framework.

3.2 Heliostat Selection

In ML it is a common approach to verify a model's prediction capabilities on a minimal subset of the available data, so as to reduce runtime and expenditure of computational resources. This approach however, requires that the chosen subset holds a similar distribution of features as its original source. In order to validate the new calibration method, it is applied on selection of 20 heliostats which are required to be a representative subset of STJ's heliostat field.

k-means clustering is used to divide the heliostat field into $k = 20$ clusters. The algorithm is initiated by randomly assigning all heliostat elements h within the feature space to one of the k groups. Thereafter, the algorithm follows an iterative process, wherein it reassigns one element after another to that clusters with the closest centroid C, which is derived from the mean of a cluster's feature distribution. The algorithm reaches convergence, when the Euclidian distances d of members within one cluster to their common centroid is minimal. [42]

k-means clustering is applied on the 2D-feature space of heliostat's east-north coordinates, so the Euclidian distance simplifies to:

$$\text{Euclidian Distance: } d_{\text{eucl}}(e_h, n_h) = \sqrt{(e_h - e_{\text{C}})^2 + (n_h - n_{\text{C}})^2} \tag{3.1}$$

Figure 3.2: Selection of 20 heliostats after data filtering and k-Means.

Not all heliostats satisfied the data requirements which are imposed on them and therefore are filtered out of the original pool containing all heliostats within the field. The large number of 2014 heliostats for which measurement data can be found on PAINT's database is reduced by filtering for two conditions:

First, only those heliostats are considered to comply to this study's purposes, which have known surface properties, so as to facilitate forward raytracing. Thus, the first filter is applied by removing all heliostats from the field's population, to which no deflectometry data is presently available (grey colored samples). This results in a reduced population size of 500 heliostats.

The second filter (light-blue) is based on the condition that the model is trained, validated and tested on sample sizes of 30 calibration measurement instances each, which means that only those heliostats can be considered, which have at least 90 measurement points associated to them [1]. The final size of heliostat population on which k-means clustering is applied equals 372 (dark-blue and black colored samples).

[1] It should be noted, that this is not a strict requirement and calibration can generally be performed on heliostats with viewer data samples.

Figure 3.2 shows the spatial distribution of heliostat populations at each stage of applying these filter conditions.

3.3 Sun Position Splits

k-Nearest Neighbors (k-NN) as a conservative approach for data splitting in heliostat calibration has been explored by Pargmann et al in [11, 39]. The researchers introduced a distance metric for splitting the total set of calibration data by the sun's positional features (azimuth, elevation) at the time of measurement. The presently used variant of k-NN algorithm begins by ordering the set of measurements D by each samples' x mean Euclidian distance to its k-nearest neighbors:

$$k\text{-NN Distance: } \bar{d}_i(x, D) = \frac{1}{k}\left(\sum_{i=1}^{k} \|x - d_i\|\right), \quad d_i \in D \tag{3.2}$$

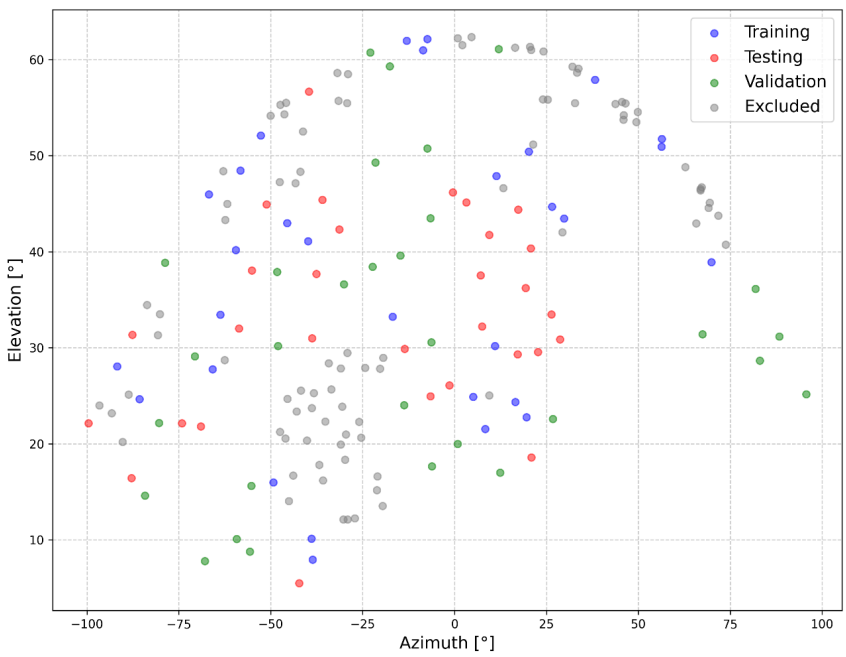


Figure 3.3: k-NN split along Euler sun angles for a heliostat's calibration measurement points.

Based on this hierarchy, the samples with the highest distances are then successively assigned to validation and testing datasets, ensuring that these sets represent the most challenging extrapolative scenarios for the model. The remaining samples, which reside closer to others in the sun position domain, form the training dataset. Throughout this study, the third-nearest neighbor is considered, ie. $k = 3$.

An example visualization of sun positions for a calibration dataset which is split by k-NN and contains 30 samples per training, validation and testing set is presented in Figure 3.3. As expected, the measurement points in the validation dataset are most remote, which means that they are located at less frequently represented sun positions. In comparison, the samples used for model testing and training exhibit sun positions with a growing frequency of representations in the total dataset. Finally, the excluded samples display a densely-populated distribution of Euler's angular features.

3.4 Image Preparation

The flux image data processed in these experiments is stored in grey-scale format with a fixed resolution of 256×256 pixels. To ensure consistent processing throughout the model pipeline, the ray tracing environment uses the same resolution for its simulated flux predictions.

Both measured and predicted flux bitmaps undergo a normalization step to scale pixel intensities $I(i, j)$ to the range $[0, 1]$ by:

$$\text{Normalized Image: } I_{\text{norm}}(i, j) = \frac{I(i, j) - I_{\text{min}}}{I_{\text{max}} - I_{\text{min}} + \varepsilon} \tag{3.3}$$

where I_{min} and I_{max} are the minimum and maximum pixel intensities, respectively and $\varepsilon = 10^{-8}$ is added to the denominator to avoid the division by zero.

This normalization yields the relative flux intensities. Scaling intensities to a fixed range ensures consistent scale across samples and has been reported to stabilize the behavior of loss-based gradient signals [43]. Also, the techniques for contour-extraction discussed in section 4.7 require a common intensity scale throughout all flux data.

Further preparation aims at removing noise from flux images using a combination of interpolation and Gaussian filtering [44]. Specifically, interpolation is applied via multiple up-sampling and down-sampling steps to increase spatial continuity in the flux distribution. In effect, the interpolation thus averages intensities across neighboring pixels. The Gaussian filter is then used to suppress high-frequency noise and enhance the detectability of structural features.

4 Methodology

4.1 Blocking and Shading

Central to the calibration experiments undertaken within this study are the herein proposed methods to include blocking and shading in the simulated datasets.

4.1.1 Implementation through Image Augmentation

To emulate the physical effect of heliostat-to-heliostat blocking and shading (cf. Section 2.3.2 within the simulated datasets, an image augmentation method is implemented. An abstract representation of the proposed method, which mimics partial flux loss due to blocking and shading is shown in Figure 4.1.

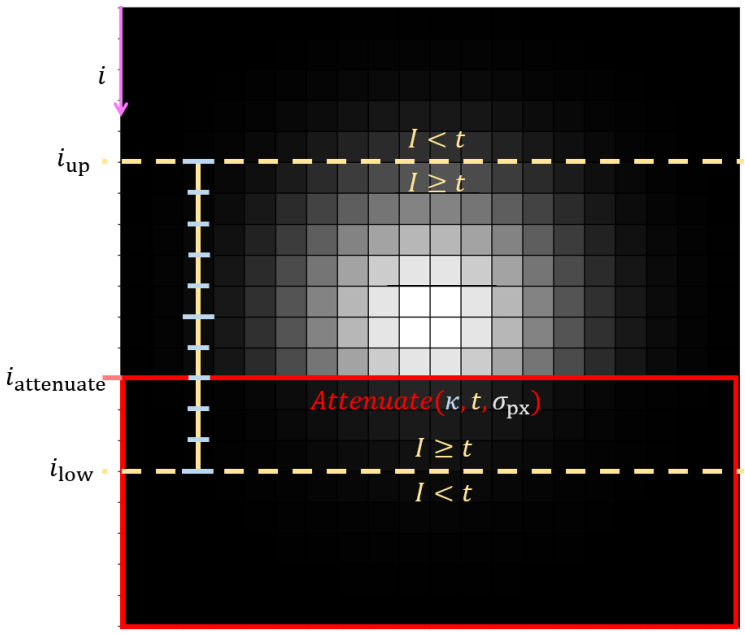


Figure 4.1: Abstract visualization of the image-augmentation function used for artificial blocking and shading with $\kappa = 0.3$. The function progresses along the pixel row index i. I denotes the pixel intensity distribution, t this is the minimum threshold value and σ_{px} is the diffusion width in pixels.

In effect, the functions aims to approximate the partial obstruction of the reflected flux that may occur when another heliostat is positioned between the source heliostat

and the receiver tower. Such partial occlusion leads to a characteristic attenuation of the flux intensity, predominantly in the lower portion of the focal spot image [30]. Within in this thesis, a function is implemented which modifies a given normalized flux image $I \in \mathbb{R}^{H \times W}$ through vertical attenuation of pixel intensities. Further input parameters which are used by the function are described in Table 4.1.

Table 4.1: Input parameters to the image augmentation function which mimics the effects of heliostat-to-heliostat blocking and shading.

Parameter	**Documentation**
flux image $I \in \mathbb{R}^{H \times W}$	2D tensor of shape $[H, W]$, representing the bitmap of the flux intensity distribution over image height H and width W. Intensity values should be normalized to range $[0, 1]$.
blocking /shading-factor κ	A float $\in [0, 1]$ defining the fraction of the focal spot height that will be affected by the blocking / shading. A value of 0.5 implies that approximately the lower half of the flux image will be attenuated.
pixel threshold t	A scalar used to detect the presence and vertical extent of the focal spot. It is applied to the normalized flux image to identify non-zero flux regions.
diffusion width σ	A parameter controlling the softness of the transition between unblocked and blocked regions. Larger values lead to a smoother decay in intensity.

In reality the scattering of rays at the intersecting edge – shown in Figure 4.2 – leads to a soft shading edge rather than an hard edge with pixel intensities smoothly fading toward the bottom. The proposed function is adapted to consider ray scattering.

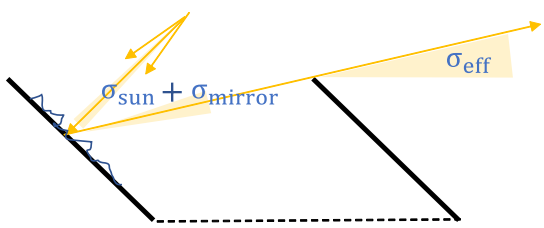


Figure 4.2: Contributors to the effective angular spread at the blocking edge of a front heliostat.

The proposed implementation first computes the vertical bounds i_{low}, i_{up} of the focal spot by thresholding based on the minimum pixel intensity t. From the detected height $(i_{\text{low}} - i_{\text{up}})$, and the blocking/shading-factor κ the pixel row i_{block} is determined, above which the image remains unaltered:

$$\text{Start Attenuation Row Index: } i_{\text{attenuate}} = i_{\text{low}} - \kappa \cdot (i_{\text{low}} - i_{\text{up}}) \tag{4.1}$$

Below this line, an exponential attenuation mask is applied, increasing in strength

with distance from $i_{\text{attenuate}}$. Above this line, i.e., for $i < i_{\text{attenuate}}$, the attenuation factor is kept at 1 leaving the area unaltered. The soft attenuation is governed by an exponential decay function using the σ_{px} as diffusion width in pixels:

$$\text{Attenuation Function: } a(i) = \begin{cases} 1, & \text{if } i < i_{\text{attenuate}} \\ \exp\left(-\dfrac{i - i_{\text{attenuate}}}{\sigma_{\text{px}}}\right), & \text{otherwise} \end{cases} \tag{4.2}$$

In order to realistically emulate the scattering of rays occurring at the edge of the intersecting object the decay parameter σ is scaled linearly based on the distance $D_{\text{H-T}}$ [m] between the heliostat's center position and the aimed calibration target's center to. The linear scaling is performed in order to approximate the increasing diffusion of rays for higher heliostat distances to the tower. To obtain σ_{px}, the diffusion width in meters σ_{m} is transformed to:

$$\text{Diffusion Width [px]: } \sigma_{\text{px}} = \frac{\sigma_{\text{m}}}{h_{\text{R}}/H} = \frac{D_{\text{H-T}} \cdot \sigma_{\text{eff}}}{h_{\text{R}}/H} \approx \frac{D_{\text{H-T}} \cdot \sqrt{\sigma_{\text{sun}}^2 + \sigma_{\text{mirror}}^2}}{h_{\text{R}}/H} \tag{4.3}$$

Where σ_{eff} is the effective angular spread of the reflected ray bundle at the blocking / shading edge, and σ_{sun}, σ_{mirror} respectively are the standard deviation of the solar intensity and the mirror slope error (cf. Figure 4.2). Other optical sources of ray diffusion are neglected within the approximation of the total angular spread [18]. From the literature the two contributors to the diffusion of reflected are assumed to be constant at $\sigma_{\text{sun}} \approx 4.65$ mrad [45] and $\sigma_{\text{mirror}} \approx 1.0$ mrad [15]. Thus, the effective angular spread can be computed to:

$$\text{Effective Spread: } \sigma_{\text{eff}} \approx \sqrt{(4.65 \text{ mrad})^2 + (1.0 \text{ mrad})^2} = 4.7563 \text{ mrad} \tag{4.4}$$

The final image is obtained by element-wise multiplication of the original flux image I with the attenuation mask $A \in \mathbb{R}^{H \times W}$, where each pixel row is constructed using the scalar function $a(i)$:

$$\text{Blocked Flux: } I_{\text{blocked}}(i,j) = a(i) \cdot I(i,j) \quad \text{for all } (i,j) \in [1,H] \times [1,W]$$

Figure 4.3 displays example images obtained as this function's output at three different levels of κ. The heliostats AA39 and BG50 are those concentrating mirrors within the selection of 20 which are positioned closest and farthest from the receiver tower.

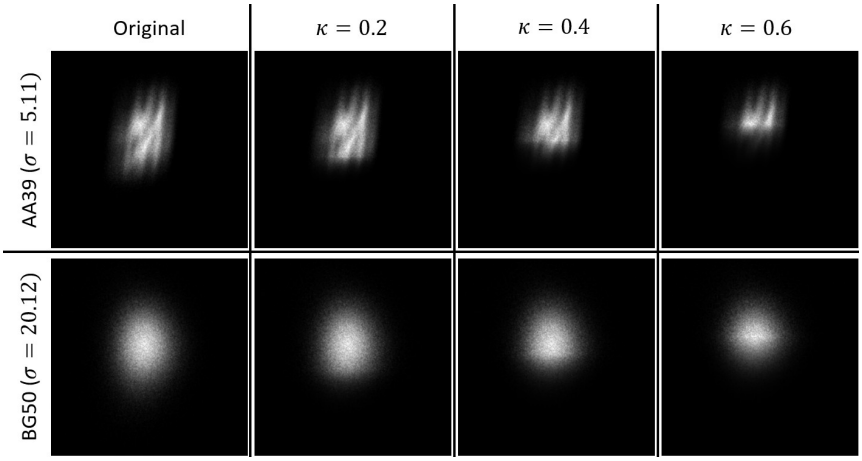


Figure 4.3: Example outputs from the image augmentation function which mimics partial occlusion of flux images at the lower flux partition. The outputs for two heliostats and three levels of κ are shown. To each heliostat the euclidean distance from the heliostat's center to the calibration target center t is given, from which σ_{px} is computed along equation 4.3.

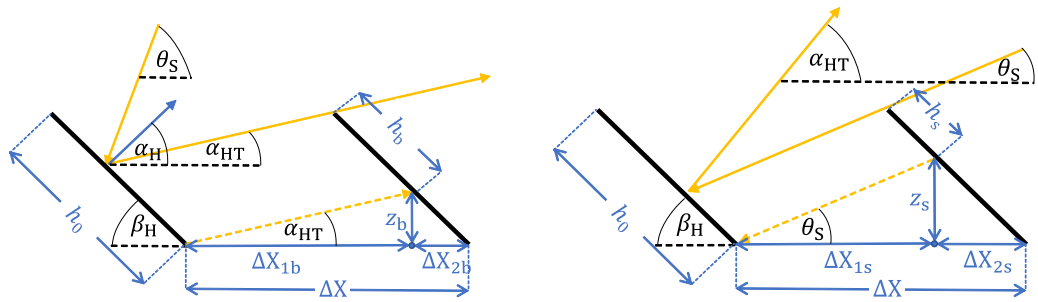
While this augmentation technique does not physically replicate the true geometric interactions that cause heliostat-to-heliostat interception, it provides a plausible approximation of the related image artifacts. Despite its simplicity, the method enables qualitative exploration of how partially blocked flux images influence the performance of the calibration algorithm.

4.1.2 Geometric Model for Realistic Evaluation

A geometric model is implemented, which allows estimates of the interactions between heliostats at specific positions in the field and in relation to the sun position at time of measurement. This model relies on a simple 2D-approximation of the relative surface factor that is exposed to partial ray interception by a front heliostat, similar to the computations in [28]. This model used for approximating the factor κ of blocked and shaded surface area is described in the following.

Figures 4.4a and 4.4b present the 2-dimensional analysis for the cases of blocking and shading which serve as the mathematical foundation for the developed model.
The analysis of the trigonometric relations portrayed in the figures, the effective surface ratio that is exposed to blocking and shading can be derived. Both figures display a simplified case, where the surface tilt β_{H} is assumed to be equal for both front and rear heliostat. The other important assumption is, that all heliostats are mounted on the same height relative to the ground. Table 4.2 lists the geometric parameters and their meaning.

The computation of the basic local parameters is performed with the following three equations (4.5, 4.6, 4.7). The target reflection vector for a given heliostat can be cal-



(a) Analysis of heliostat-to-heliostat blocking. **(b)** Analysis of heliostat-to-heliostat shading.

Figure 4.4: Visualization of the geometric 2-dimensional model used to evaluate blocking (a) and shading (b), inspired by [28].

culated from the heliostat's vertical distance X to the receiver tower and the receiver center's height z_{R} to:

$$\alpha_{\mathrm{HT}} = \arctan \frac{z_{\mathrm{R}}}{X} \tag{4.5}$$

The vertical surface normal tilt which is required for aiming along α_{HT} depends on the current sun's elevation angle θ_{S}:

$$\alpha_{\mathrm{H}} = \frac{\theta_{\mathrm{S}} + \alpha_{\mathrm{HT}}}{2} \tag{4.6}$$

Subsequently the vertical tilt of the heliostat surface is given as:

$$\beta_{\mathrm{H}} = \frac{\pi}{2} - \alpha_{\mathrm{H}} \tag{4.7}$$

For the geometric analysis of the **blocked scenario**, first a relationship for the support-variable z_{b} is found, which is then solved for z_{b}:

$$\frac{z_{\mathrm{b}}}{\Delta X_{1,\mathrm{b}}} = \tan \beta_{\mathrm{H}}, \ \frac{z_{\mathrm{b}}}{\Delta X_{2,\mathrm{b}}} = \tan \alpha_{\text{H-T}} \tag{4.8}$$

$$\Leftrightarrow z_{\mathrm{b}} = \Delta X_{1,\mathrm{b}} \cdot \tan \beta_{\mathrm{H}} = \Delta X_{2,\mathrm{b}} \cdot \tan \alpha_{\text{H-T}} \tag{4.9}$$

Generally, the sum of ΔX_1 and ΔX_2 is equal to the heliostat spacing distance ΔX:

$$\Delta X = \Delta X_1 + \Delta X_2 \tag{4.10}$$

$$\Leftrightarrow \Delta X_2 = \Delta X - \Delta X_1 \tag{4.11}$$

After inserting equation 4.11 into equation 4.9 we solve for $\Delta X_{1,\mathrm{b}}$ to:

$$\Delta X_{1,\mathrm{b}} \cdot (\tan \beta_{\mathrm{H}} + \tan \alpha_{\text{H-T}}) = \Delta X \cdot \tan \alpha_{\text{H-T}} \tag{4.12}$$

$$\Leftrightarrow \Delta X_{1,\mathrm{b}} = \frac{\Delta X \cdot \tan \alpha_{\text{H-T}}}{\tan \beta_{\mathrm{H}} + \tan \alpha_{\text{H-T}}} \tag{4.13}$$

Table 4.2: Geometric parameters in heliostat-to-heliostat blocking and shading.

Parameter	Documentation
α_{H}	Vertical tilt angle of heliostat mirror [rad] (local variable).
α_{HT}	Vertical elevation angle of the heliostat to target-center vector or reflection vector [rad] (local variable).
β_{H}	Vertical elevation angle of the heliostat normal vector [rad] (local variable).
θ_{S}	Vertical elevation angle of the sun's incident ray vector [rad] (global variable).
$h_0 = 3.22$ m	Height of heliostat surface [m] (global variable).
h_{b} / h_{s}	Height of blocked / shaded heliostat surface [m] (local variables).
$z_{\mathrm{R}} = 55.22$ m	Vertical coordinate of receiver center in local ENU-coordinate system [m] (global variable).
ΔX	Horizontal spacing between heliostats [m] (local variable).
$\Delta X_{1\mathrm{b}}$, $\Delta X_{2\mathrm{b}}$, z_{b}	Support variables in the analysis of blocking [m] (local variables).
$\Delta X_{1\mathrm{s}}$, $\Delta X_{2\mathrm{s}}$, z_{s}	Support variables in the analysis of shading [m] (local variables).

Using the geometric relationship between $\Delta X_{1,\mathrm{b}}$ and the unblocked mirror height $(h_0 - h_{\mathrm{b}})$ we solve for h_{b} and then insert equation 4.13:

$$\frac{\Delta X_{1,\mathrm{b}}}{h_0 - h_{\mathrm{b}}} = \cos\beta_{\mathrm{H}} \Leftrightarrow h_{\mathrm{b}} = h_0 - \frac{\Delta X_{1,\mathrm{b}}}{\cos\beta_{\mathrm{H}}} \tag{4.14}$$

$$\Leftrightarrow h_{\mathrm{b}} = h_0 - \frac{\Delta X}{\cos\beta_{\mathrm{H}}} \cdot \frac{\tan\alpha_{\text{H-T}}}{\tan\beta_{\mathrm{H}} + \tan\alpha_{\text{H-T}}} \tag{4.15}$$

Finally, the relative factor for the blocked heliostat is obtained by relating h_{b} to h_0:

$$\text{Block Factor: } f_{\mathrm{b}} = \frac{h_{\mathrm{b}}}{h_0} = 1 - \frac{\Delta X}{h_0 \cdot \cos\beta_{\mathrm{H}}} \cdot \frac{\tan\alpha_{\text{H-T}}}{\tan\beta_{\mathrm{H}} + \tan\alpha_{\text{H-T}}} \tag{4.16}$$

For the **shaded scenario**, a similar analysis of the geometric relations is made beginning with the support-variable z_{s}:

$$\frac{z_{\mathrm{s}}}{\Delta X_{1,\mathrm{s}}} = \tan\beta_{\mathrm{H}},\ \frac{z_{\mathrm{s}}}{\Delta X_{2,\mathrm{s}}} = \tan\theta_{\mathrm{S}} \tag{4.17}$$

$$\Leftrightarrow z_{\mathrm{s}} = \Delta X_{1,\mathrm{s}} \cdot \tan\beta_{\mathrm{H}} \cdot = \Delta X_{2,\mathrm{s}} \cdot \tan\theta_{\mathrm{S}} \tag{4.18}$$

By substituting 4.11 we obtain the following relationship, the we solve for $\Delta X_{1,\mathrm{s}}$:

$$\Delta X_{1,\mathrm{s}} \cdot (\tan\beta_{\mathrm{H}} + \tan\theta_{\mathrm{S}}) = \Delta X \cdot \tan\theta_{\mathrm{S}} \tag{4.19}$$

$$\Leftrightarrow \Delta X_{1,\mathrm{s}} = \frac{\Delta X \cdot \tan\theta_{\mathrm{S}}}{\tan\beta_{\mathrm{H}} + \tan\theta_{\mathrm{e}}} \tag{4.20}$$

We now formulate the cosine of β_{H} to obtain the formula for h_{s}:

$$\frac{\Delta X_{1,\mathrm{s}}}{h_0 - h_{\mathrm{s}}} = \cos\beta_{\mathrm{H}} \Leftrightarrow h_{\mathrm{s}} = h_0 - \frac{\Delta X_{1,\mathrm{s}}}{\cos\beta_{\mathrm{H}}} \tag{4.21}$$

$$\Leftrightarrow h_{\mathrm{s}} = h_0 - \frac{\Delta X}{\cos\beta_{\mathrm{H}}} \cdot \frac{\tan\theta_{\mathrm{S}}}{\tan\beta_{\mathrm{H}} + \tan\theta_{\mathrm{S}}} \tag{4.22}$$

The factor for the relative surface height that is exposed to shading is solved to:

$$\text{Shading Factor: } f_{\mathrm{s}} = \frac{h_{\mathrm{s}}}{h_0} = 1 - \frac{\Delta X}{h_0 \cdot \cos\beta_{\mathrm{H}}} \cdot \frac{\tan\theta_{\mathrm{S}}}{\tan\beta_{\mathrm{H}} + \tan\theta_{\mathrm{S}}} \tag{4.23}$$

Finally, the individual factors of f_{b} (4.16) and f_{s} (4.23) must be combined to a single-scalar f_{bs} which resembles the total strength of these contributors. As both effects of occlusion by definition occur at the lower section of the mirror surface, these individual terms should not be added. In order to avoid that a surface area which is already shaded contributes to blocking (and vise-versa), the total f_{bs} is found by applying a max-and subsequent min-operation:

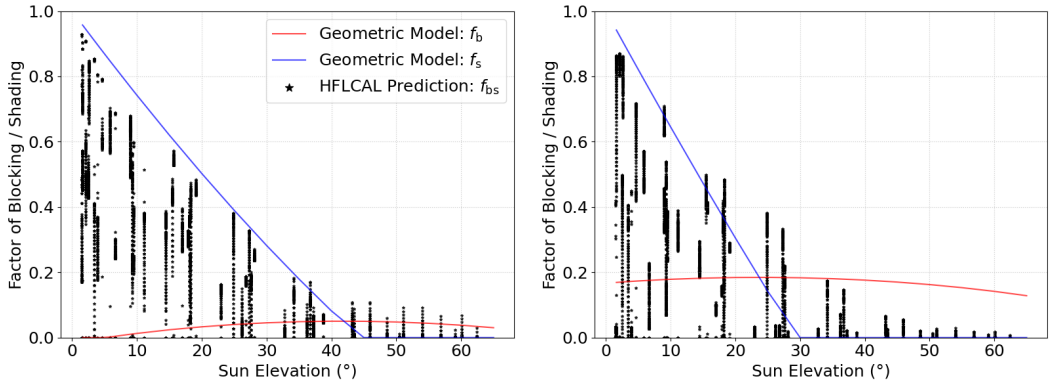
$$\text{Blocking \& Shading Factor: } f_{\mathrm{bs}} = \min(\max(f_{\mathrm{b}}, f_{\mathrm{s}}, 0), 1) \tag{4.24}$$

The combination of the min-max-operations has the purpose of ensuring that the final scalar output lies within the physical meaningful boundaries $\in [0, 1]$.

Figure 4.5 displays the evolution of f_{b} and f_{s} over the sun's elevation angle and for two heliostat rows. The graphs for the 'Geometric Model' rely on the proposed method of computation, see equations 4.16 and 4.23.

To evaluate our assumptions, we compare our analytical assessment with the simulated values for f_{bs} from HFLCAL – a software tool for layout and optimization of heliostat fields [46]. The factor for the sum of blocked and shaded surface ratio is scattered for the available heliostats within the respective row over the sun's elevation angle. The comparison of both the predictions from the proposed model and HFCAL, which performs geometric calculations within the 3-dimensional space, allows partial validation of the simplified model.

As expected, the 2D model generally overestimates the factor of blocking and shading which is the result from the simplified consideration of the geometric interactions between the heliostat rows. HFLCAL for one and the same elevation angle produces



(a) Row at $X = 61.8$ m, $\Delta X = 4.6$ m

(b) Row at $X = 125.4$ m, $\Delta X = 6.5$ m

Figure 4.5: Predictions from the geometric model which evaluates blocking and shading factors for two example heliostat rows. The blocked / shaded surface factors are plotted over the sun elevation angles and compared to the data obtained from the software tool HFLCAL [46]. X is the row's horizontal distance from the receiver tower, ΔX is the spacing between heliostat rows.

a wide range of f_{bs}-values which are the results for different east-north positions for the considered heliostat row and varying sun azimuth angles. The proposed approach views the front row of heliostats as a continuous surface with an upper intersecting edge (no gaps), although the heliostats within a row are spaced apart along the east-north direction leaving gaps which reduce the actual f_{bs}. A more complex approach – as is implemented in HFLCAL – considers each heliostat as an individual polygon which is aligned by dual-axis tracking.

Figure 4.5b suggests that the simplified model overestimates the potential contribution of the blocking factor for the here considered row at $X = 125.4$ m. Due to the above mentioned points of criticism on the proposed model for evaluating the occlusion effects of blocking and shading, the actual implementation of the programmatic algorithm uses the f_{bs} obtained from equation 4.24 as the upper limit of a uniform distribution:

$$\kappa \sim \mathcal{U}(0, f\text{bs}) \tag{4.25}$$

The effective value κ for the relative strength of occlusion which is used as an input to the image augmentation function (c.f Section 4.1.1) is randomly sampled from the above distribution.

4.2 Machine Learning Framework

This section presents the framework on the image-based calibration algorithm and introduces the underlying loss functions. The calibration model offers the functionality

to be trained both on a geometric alignment-loss and on the image-based contour-loss. For derivation of the contour-loss a function is proposed for detection of the upper edges of focal spots which is further processed in order to extract spatial and structural features of the flux distribution. The contour-loss is computed by comparing the prediction's image features against those found in the ground-truth's flux.

4.2.1 Model Structure

Figure 4.6 provides an overview to the proposed calibration algorithm which is using flux-image data to derive a contour-loss for optimization of the kinematic model.

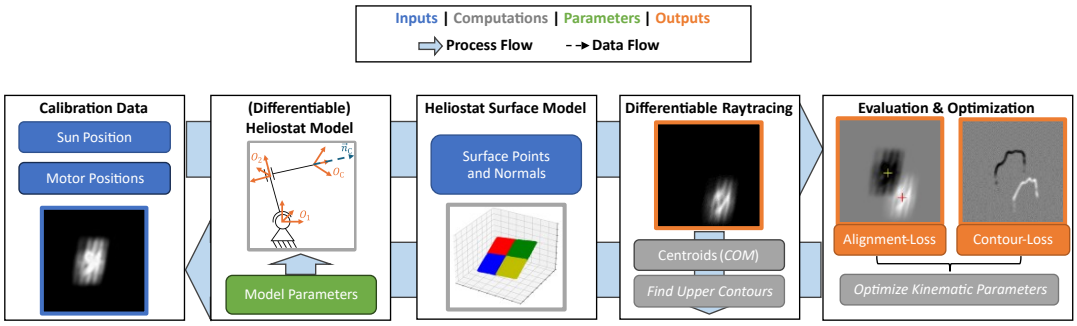


Figure 4.6: Overview of the proposed approach for image-based heliostat calibration using a contour-derived loss.

4.2.2 The Alignment-Loss

In accordance with the discussion in subSection 2.2.4, it is important to distinguish between the reflection axis and the COM-vector (approximated reflection axis), which relies on the measured flux centroid. Throughout this study, the latter is used for loss calculation and model optimization. This approach ensures that the model cannot exploit information which is unavailable in measured datasets. For this reason, the metric based on the ideal reflection axis is only computed as a final evaluation metric outside of PyTorch's computational graph [1].

The alignment loss is based on the cosine similarity between a sample's s reflection vectors $r_{\mathrm{P},s}$ and $r_{\mathrm{G},s}$:

$$\text{Cosine Similarity:} \quad \cos\theta_s = \frac{\vec{r}_{\mathrm{P},s} \cdot \vec{r}_{\mathrm{G,COM},s}}{\|\vec{r}_{\mathrm{P},s}\| \cdot \|\vec{r}_{\mathrm{G,COM},s}\|} \tag{4.26}$$

Where θ_s is the angular misalignment. In this context, the cosine similarity directly measures alignment, with a value of $\cos\theta_s = 1$ indicating perfect alignment. The

[1] Only for benchmarking purposes a single model variant was trained on the alignment-loss as derivation of the misalignment between the predicted and ground-true reflection axes (cf. Section 6.3).

MSE-loss (cf. Section 2.3.3) is then computed per heliostat of index h by applying equation 2.16 with $y_i = \cos\theta_i$ and $\hat{y}_i = 1$ over the heliostat's N measurement samples:

$$\text{Alignment-Loss: } \mathcal{L}^{(h)}_{\text{Alignment}} = \frac{1}{N}\sum_{s=1}^{N}\left(\frac{\vec{r}_{\text{P},s}\cdot\vec{r}_{\text{G,COM},s}}{\|\vec{r}_{\text{P},s}\|\cdot\|\vec{r}_{\text{G,COM},s}\|} - 1\right)^2 \tag{4.27}$$

For the conventional calibration strategies which rely purely on vectorial alignment, the single-scalar loss term is computed by averaging each heliostat's contribution to obtain a final loss across the heliostat field of H heliostats:

$$\text{Alignment-Loss (Field): } \mathcal{L}_{\text{Alignment}} = \frac{1}{H}\sum_{h=1}^{H}\mathcal{L}^{(h)}_{\text{Alignment}} \tag{4.28}$$

4.2.3 Contour Extraction

Within this study, image-based calibration is realized by implementing a pipeline for processing flux image data, which attempts to derive a loss term that provides gradients for model alignment along the upper contours of the focal spots. In this context, a contour is defined as the boundary of the flux distribution within a given image. Performing the calibration specifically along the contour's upper section is motivated by the observation that heliostat-to-heliostat occlusion most often affects the lower portion of the focal spot, whereas the upper region typically is conserved. As such, the upper contour provides a reliable and robust feature for image-based calibration.

Before a contour-loss can be formulated, it is essential to establish a robust and differentiable method for extracting the upper contour from a given flux image. This is accomplished by applying a convolutional operation to the input bitmap using a vertical Sobel kernel (cf. Section 2.4.5).

The complete pipeline as shown in Figure 4.7 for contour extraction involves further steps for smoothing and de-noising the original flux image which were introduced in Section 3.4. Within the preparation stage, the image values are normalized before undergoing q number of interpolations, wherein the spatial continuity pf the input is improved by consecutive up-sampling and down-sampling the pixel values. This step is followed by the application of a Gaussian filter for reducing high-frequency noise and further enhancing the detectability of continuous edges.

To make the contour detection process differentiable and flexible by including tunable parameters (threshold τ and sharpness η), a step for soft thresholding is applied wrapped in a sigmoid-function σ:

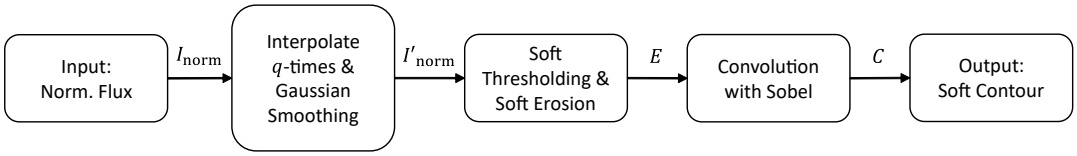


Figure 4.7: Program flowchart for extracting the upper contour of an input flux image.

$$\text{Soft Thresholding:} \quad B(i,j) = \text{sigmoid}\left(\eta \cdot (I'_{\text{norm}}(i,j) - \tau)\right) \tag{4.29}$$

$$\text{with sigmoid}(x) = \frac{1}{1+\exp(-x)} \tag{4.30}$$

$$\Rightarrow B(i,j) = \frac{1}{1+\exp\left(-\eta \cdot (I'_{\text{norm}}(i,j) - \tau)\right)} \tag{4.31}$$

In addition, soft erosion is performed by local averaging over a $3x3$-neighborhood to suppress isolated noise by mean convolution:

$$\text{Soft Erosion: } E(i,j) = \frac{1}{9} \sum_{m=-1}^{1} \sum_{n=-1}^{1} B(i+m, j+n) \tag{4.32}$$

The combination of the soft thresholding and erosion operations should result in a smooth, continuous contour representation suitable for gradient-based optimization [47]. Finally, the convolution operation with the vertical Sobel-kernel computes the image bitmap for the upper contour. It achieves this by essentially emphasizing positive gradients along the vertical image axis – representing transitions form dark to bright:

$$\text{Soft Contour: } C(i,j) = (B * K_{\text{Sobel},y})(i,j) \tag{4.33}$$

A series of example images obtained by applying the function for contour extraction is found in Figure 4.8.
As intended the choice for the threshold value influences the elongation of the contour that is extracted from the flux image. With a threshold closer to 1.0 the contour is generated on pixels of higher intensities which are found closer in the flux's center region. The sharpness factor is effective in tuning the width of the contour: With a higher sharpness value the contour becomes sparser.

To optimally tune the contour extraction parameters, a Bayesian-optimization scheme is suitable. For example, Jabour (2025) applied Bayesian optimization to determine optimal input parameters for a related application of contour extraction [48].

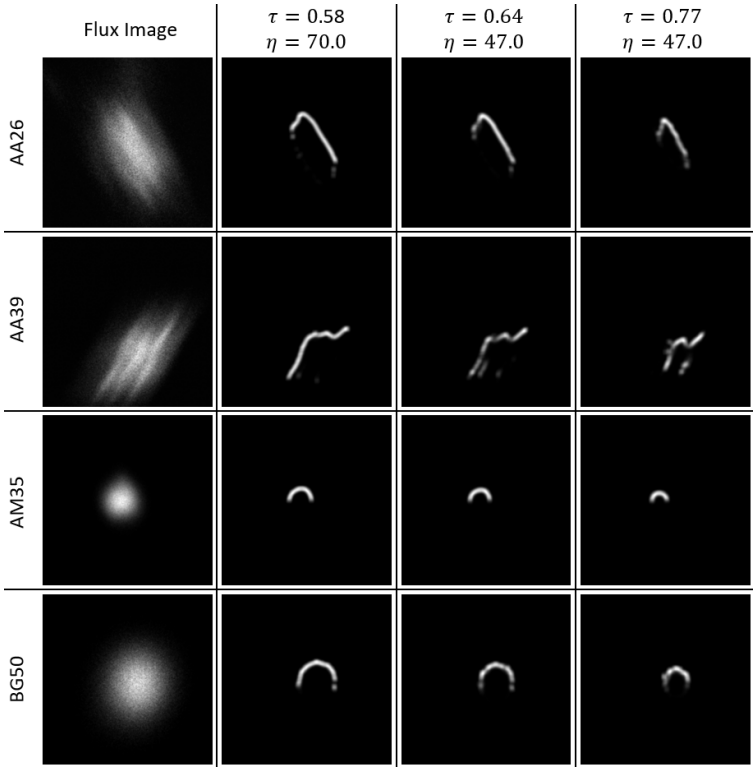


Figure 4.8: Example outputs from the contour extraction function which applies uses a Sobel-kernel for detection of the flux upper edge (contour). The contours for four heliostats and three settings for the threshold τ and sharpness η are shown.

In Figure 4.8 the first row next to the original flux images presents example contours which are generated with thus parameters settings found by Bayesian optimization ($\tau = 0.58, \eta = 70.0$). This configuration is also applied for contour detection and subsequent loss computation within the image-based calibration attempts. It should be noted that although the chosen parameters perform well on the datasets used in this study, their applicability to other datasets may required reconfiguration.

4.2.4 The Contour-Loss

The proposed calibration approach processes the detected contours in order to derive an image-based loss. With the goal of providing meaningful and stable gradients for contour-based alignment, this section proposes three distinct loss terms which will be referred to as: Coarse, Fine, and Gravity loss. The inclusion of these complementary loss terms is motivated by the goal of providing a multifaceted penalty landscape.

Coarse Loss

The conceptual design of the Coarse alignment loss term is motivated by the goal of providing gradients for heliostat calibration in situations where there is not overlap

between the predicted and ground-truth contours. For this term, the Soft Distance Field (SDF) loss is applied. The SDF loss is based on the principle of penalizing predicted contour pixels according to their Euclidean distance to the nearest ground-truth contour reference pixel to the end of providing meaningful gradients even in the absence of direct overlap [49].

First a distance transformation is applied on the ground truth contour to compute the reference distance map. The algorithm for this transformation is shown in figure 4.9.

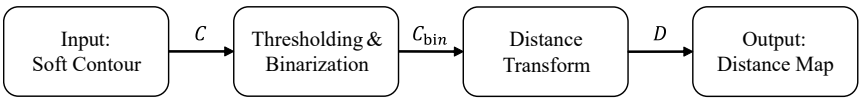


Figure 4.9: Program flowchart for computing the distance map.

The pre-processing step of applying a minimum pixel threshold τ and binarization follow the goal of producing a binarized mask with values of one indicating a positive contour pixel, while values of 0 mask the background pixels:

$$C_{\text{bin}}(i,j) = \begin{cases} 1, & \text{if } C(i,j) > \tau \\ 0, & \text{otherwise} \end{cases} \tag{4.34}$$

The distance transformation then computes a map of Euclidian distances from each pixel to its nearest contour pixel at (i', j'):

$$D(C_{\text{bin}}) = \min_{(i',j'):C_{\text{bin}}=0} \sqrt{(i-i')^2 + (j-j')^2} \tag{4.35}$$

Based on the ground truth's distance map, the SDF-loss is calculated as the element-wise product of the prediction's contour and the ground truth's distance map:

$$\text{Coarse Loss: } \mathcal{L}_{\text{Coarse}}^{(h)} = \frac{1}{N}\sum_{s=1}^{N} \text{SDF}(C_{\text{P},s}, C_{\text{G},s}) = \frac{1}{N}\sum_{s=1}^{N}\sum_{i,j} C_{\text{P},s}(i,j) \cdot D_{\text{G},s}(i,j) \tag{4.36}$$

By multiplying the reference map's distances with the predicted contour's local pixel values, the SDF specifically evaluates how far each pixel in the predicted distribution lies from its nearest neighbor in the ground-true distribution. It therefore penalizes spatially misaligned contour predictions, while allowing gradients to propagate even when the contours don't yet overlap.

Fine Loss

The Fine alignment loss encourages the predicted flux contour to closely match the reference contour at a pixel-wise level. For this purpose, the DICE-coefficient is

employed for measurement of the overlap between the two contours. The DICE-based loss is especially well-suited for situations where the regions of interest (i.e., the contours) are sparse or thin [50], as it is sensitive to both false positives and false negatives, and is less affected by class imbalance than traditional metrics like the mean squared error. The DICE coefficient for two point sets A and B is defined as:

$$\mathrm{DICE}(A, B) = \frac{2 \cdot |A \cap B|}{|A| + |B|} \tag{4.37}$$

For the soft contours $C_{\mathrm{P},s}$ (prediction) and $C_{\mathrm{G},s}$ (ground-truth) over N samples, the DICE-based loss can be computed as [50]:

$$\text{Fine Loss: } \mathcal{L}_{\mathrm{Fine}}^{(h)} = \frac{1}{N} \sum_{s=1}^{N} (1 - \mathrm{DICE}(C_{\mathrm{P},s}, C_{\mathrm{G},s})) \tag{4.38}$$

$$= \frac{1}{N} \sum_{s=1}^{N} (1 - \frac{2 \sum C_{\mathrm{P},s} \cdot C_{\mathrm{G},s} + \varepsilon}{\sum C_{\mathrm{P},s} + \sum C_{\mathrm{G},s} + \varepsilon}) \tag{4.39}$$

Where $\varepsilon = 10^{-6}$ is a small constant to avoid division by zero. The DICE-loss is effective for driving the model toward precise, fine-grained alignment of the extracted contours but requires an initial overlap of contours for providing meaningful gradients.

Gravity Loss

The third loss term is devised to support the Coarse alignment by facilitating contour matching along their COM-coordinates. Fine and Coarse loss alone do not achieve stable convergence. For that reason a third Gravity loss is deployed. Its simplistic design facilitates global alignment by providing a smooth loss landscape without local minima. Previous studies comment on the benefits in stronger convergence within point cloud registration problems by incorporation of geometric constraints such as gravity-based loss terms. The COM-coordinates for a contour is computed along the procedure discussed in Section 2.2.4. The Gravity loss is then defined as the Euclidian distance between the predicted and ground truth contours' COM:

$$\text{Gravity Loss: } \mathcal{L}_{\mathrm{Gravity}}^{(h)} = \frac{1}{N} \sum_{s=1}^{N} d_{\mathrm{eucl},}(C_{\mathrm{P},s} C_{\mathrm{G},s}) = \frac{1}{N} \sum_{s=1}^{N} \|\mathrm{COM}_{C_{\mathrm{P},s}} - \mathrm{COM}_{C_{\mathrm{G},s}}\|_2 \tag{4.40}$$

As COM-points herein are given in the local ENU-coordinate system, this function in practical terms computes the distance from the predicted contour's COM to the ground-true contour's COM in meters. The Gravity loss effectively penalizes model flux predictions whose upper contour is spatially remote from their ground-truth counterpart.

4.2.5 Combination of Loss Terms

In order to provide a single loss function that can be used by PyTorch's optimizer instance for updating the kinematic parameters after each iteration, the varies loss terms described in Sections 4.2.2 and 4.2.4 have to be reduced to one single scalar. First, the three image-based contour loss terms are combined as a weighted sum:

$$\text{Weighted Loss: } \mathcal{L}_{\text{Weighted}} = (1 - \beta - \gamma) \cdot \mathcal{L}_{\text{Fine}} + \beta \cdot \mathcal{L}_{\text{Coarse}} + \gamma \cdot \mathcal{L}_{\text{Gravity}} \tag{4.41}$$

Where $(\beta, \gamma, \beta + \gamma) \in [0, 1]$ in this function are weighing coefficients, which can be tuned as hyper-parameters for controlling how much each loss term is contributing to the single-scalar weighted loss.

In the implementation of image-based calibration, it is essential to account for cases in which the forward ray tracing simulation fails to project flux onto the target surface - i.e., when no rays intersect the calibration target, resulting in an empty flux bitmap. Such a scenario occurs when a heliostat's orientation is misaligned beyond the intended target plane's spatial limits. In order for the calibration model to handle these cases effectively, it requires meaningful gradients that provide training stimuli, so that the reflected flux eventually aligns within the boundaries of the target surface.

A commonly adopted strategy in related machine computer vision tasks is to begin training with a smooth or coarse loss objective and transition to a more detailed, discontinuous loss later in training. For example, Tian et al. (2023) propose a two-stage coarse-to-fine prior-guided segmentation framework in the field of dental-medical imagery. This approach of stage-scheduling the optimization objective can improve convergence, provided that the pre-training stage sufficiently constrains the model's output space before the shift to more detailed image-based training. [51]

For the heliostat calibration model, this strategy is realized by initially optimizing only the standard geometric Alignment-loss for the first N_1-number of training epochs. Following this, training is gradually shifted toward reliance on the image-based contour loss. At each epoch ep a weighing factor α is computed, starting from $\alpha = 0$ at $ep = N_1$ and increasing linearly until $\alpha = 1$ at $ep = N_2$. This phase for soft loss transition is followed by a final training phase, in which alignment is guided exclusively by gradients derived from the image-based contour loss.

This strategy of phased learning can be formulated in mathematical terms which allows computation of the "alpha" loss for the transition in between $ep \in (N_1, N_2)$:

$$\text{Alpha Loss: } \mathcal{L}_{\text{alpha}}(ep) = (1 - \alpha(ep)) \cdot \mathcal{L}_{\text{Alignment}} + \alpha(ep) \cdot \mathcal{L}_{\text{Weighted}} \tag{4.42}$$

$$\text{with } \alpha(ep) = \frac{ep - N_1}{N_2 - N_1}, \quad \text{for } N_1 \leq ep \leq N_2. \tag{4.43}$$

Finally, the combined loss for the implementation of staged learning by shifting training objectives, is computed for any epoch ep to:

$$\text{Combined Loss: } \mathcal{L}_{\text{Combined}}(ep) = \begin{cases} \mathcal{L}_{\text{Alignment}}, & \text{if } ep < N_1 \\ \mathcal{L}_{\text{Transition}}(ep), & \text{if } N_1 \leq ep \leq N_2 \\ \mathcal{L}_{\text{Weighted}}, & \text{otherwise,} \end{cases} \tag{4.44}$$

Guardrail Mechanism

A final processing step in Equation 4.45 is a applied within loss computation to enhance the robustness of the image-base calibration against non-convergence. To this end, a guardrail mechanism is implemented that restricts the use of the contour-based loss in cases of significant heliostat misalignment: If the mean COM-accuracy (cf. Equation 2.11) in model validation for a given heliostat $\hat{\theta}^{(h)}_{\text{test}}$ exceeds a predefined threshold $\theta_{\text{guardrail}}$, calibration for that heliostat relies on the geometric Alignment-loss only. If the guardrail mechanism for any heliostat becomes active, its contribution to the final loss is overridden being replaced by the geometric alignment loss.

$$\text{Guardrail-Loss: } \mathcal{L}^{(h)}_{\text{Guardrail}}(ep) = \begin{cases} \mathcal{L}^{(h)}_{\text{Alignment}}, & \text{if } \hat{\theta}^{(h)}_{\text{test}} > \theta_{\text{guardrail}} \\ \mathcal{L}^{(h)}_{\text{Combined}}(ep), & \text{otherwise} \end{cases} \tag{4.45}$$

This selective mechanism ensures that during post-training, when parameter updates are guided by the image-based contour loss, diverging heliostats can recover. Divergence in post-training was observed in cases, where the combination of high learning rates and unstable gradient behavior led to large, de-stabilizing parameter updates from which a heliostat could not recover once its flux prediction was null. Using the cosine loss as fallback loss, enables recovery of trailing heliostats. After the heliostat's mean alignment error again falls below the predefined threshold, the guardrail mechanism deactivates and heliostat calibration resumes using image-based contour supervision. Provided that the learning rate decreases over training cycles, convergence can then succeed.

The final scalar loss used for backpropagation is then computed as the mean over all heliostats H in the field:

$$\text{Final Contour-Loss: } \mathcal{L}_{\text{Contour}} = \frac{1}{H} \sum_{h=1}^{H} L^{(h)}_{\text{Guardrail}} \tag{4.46}$$

5 Experimental Setup

The setup for the experiments for artificial dataset generation, and for comparative evaluation of both conventional and image-based calibration attempts are presented in this chapter.

5.1 Pipeline for Generating Simulated Data with Blocking

Definition of Scenarios

To systematically evaluate the comparative performance of the proposed and the conventional calibration algorithm under partial flux occlusion, a set of simulated datasets was generated using controlled blocking scenarios. Each scenario is defined by a shared configuration of modeled heliostats and a uniform distribution of calibration samples across sun positions, which ensures comparability across cases. The scenarios differ solely in the two blocking-related parameters: the relative blocking / shading ratio, denoted by κ (cf. Section 4.1.1), and the frequency f_{bs}. This variable frequency defines the proportion of calibration samples within a scenario that undergo artificial blocking and shading.

For each scenario, first a value for κ is chosen which is a fixed constant for all affected samples. Secondly, f_{bs} is set to an integer ratio (e.g. 1/3: "1 in 3 samples") as a global parameter for each simulated scenario. This parameterization enables a comparative evaluation of the calibration model's performance under varying degrees of degradation in the flux image data.

Table 5.1: Overview of Simulated Scenarios with Artificial Blocking

No.	Scenario ID	Occlusion Factor κ [%]	Frequency f_{bs}
1	`Base`	0	0
2	`Realistic`	Equation 4.25	1/1
3	`20_on_4th`	20	1/4
4	`20_on_3rd`	20	1/3
5	`40_on_4th`	40	1/4
6	`40_on_3rd`	40	1/3
7	`60_on_4th`	60	1/4
8	`60_on_3rd`	60	1/3

An overview of the seven scenarios defined for the data-driven calibration is presented in Table 5.1.

The "Base" scenario simulates the ideal situation where no data samples are affected by heliostat-to-heliostat interception at any level or frequency. The "Realistic" scenario dynamically calculates the factor of occluded mirror surface for every sample along the algorithm detailed in Section 4.1.2. At levels of κ at 20%, 40% and 60% two scenarios each are defined exhibiting a frequency of blocked samples at a frequency of 1/4 and 1/3. This collection of scenarios enables the qualitative evaluation of how the occurrence of blocked training flux samples affects the conventional calibration method's performance. The image-based calibration model can finally be tested on its ability for improving the accuracy in prediction.

Pipeline and Framework for Simulation

The general pipeline for generating the simulated datasets with artificial blocking / shading is presented in Figure 5.1. On initialization of the simulation environment, the kinematic model and the heliostat surface model is loaded based on the available parameters. In contrast to the calibration pipeline, the kinematic parameters are used as static inputs and not updated during simulation.

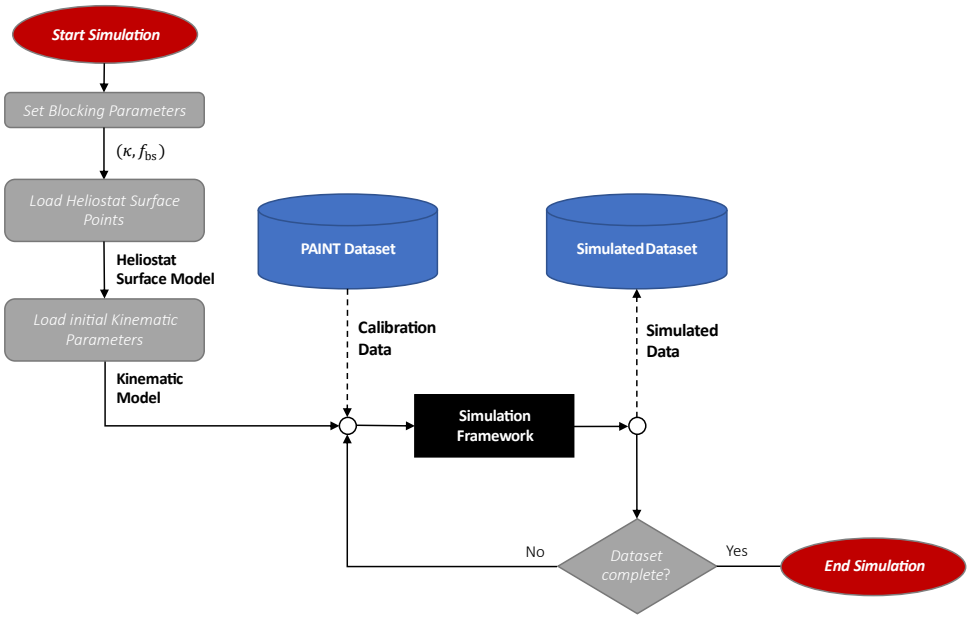


Figure 5.1: Process flowchart for creating simulated datasets.

At the start of each scenario simulation, the values for κ and f_{bs} are set. The detailed framework for data simulation is presented in Figure 5.2. From the outer-loop pipeline, the framework receives a measurement sample from the original PAINT

dataset. Within the simulation framework, the program follows a cyclic structure of processing the data sample and generating the corresponding (augmented) flux image and further specification data.

At first, the heliostat's kinematic model is **Aligned with AP** using the original flux' centroid COM as aim point. The ARTIST environment enables heliostat alignment along a given aim point by solving for the corresponding heliostat reflection axis as explained in Section 2.4.3. From the reflection axis the heliostat orientation can be computed by application of reflection law, which then allows for computation of the appropriate motor positions. This step is necessary, because the kinematic model is not yet calibrated, meaning that the motor positions stored in the measured calibration data cannot be used for valid alignment.

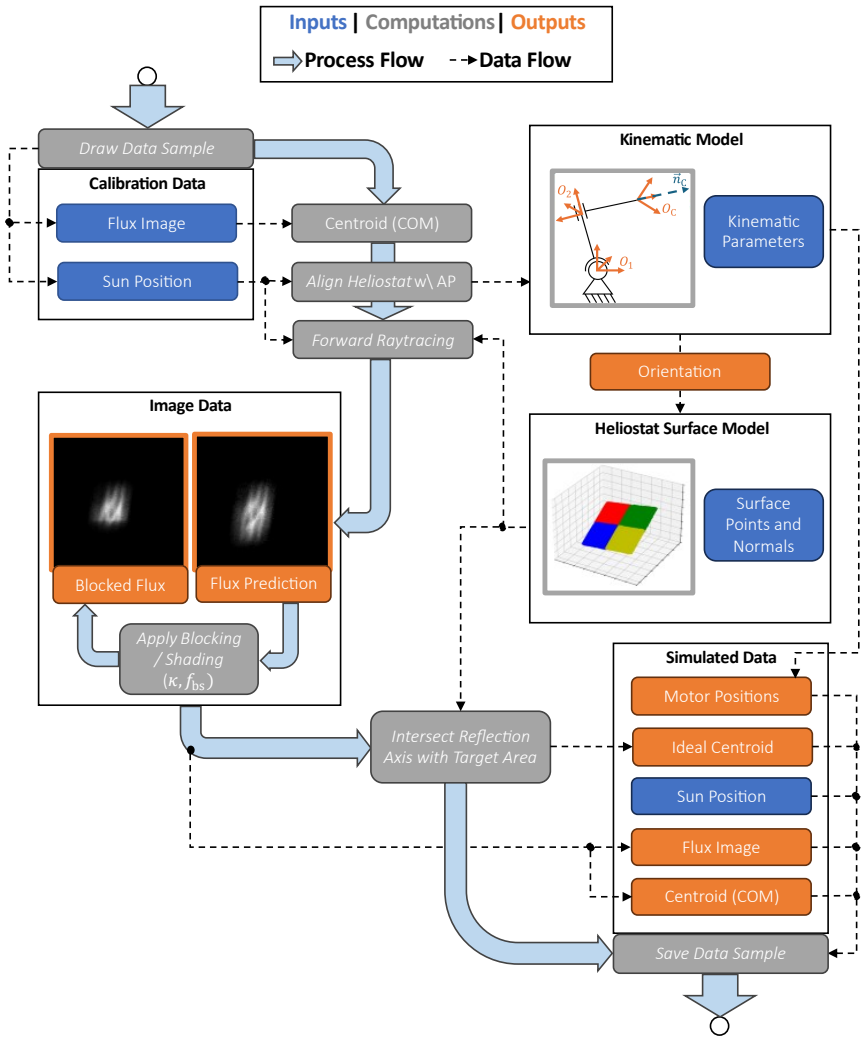


Figure 5.2: Detailed framework for generating simulated data samples with partial occlusion.

The aligned kinematic model outputs the heliostat orientation matrix (cf. Section 2.2.3) that is used for alignment of the surface points and surface normals. Based on the oriented surface model, the **Forward Raytracing** simulation is performed to output the prediction's flux image. Using the initial setup of parameters, artificial blocking and shading is applied in order to model partial occlusion (cf. Section 4.1.1). Special attention should be drawn to the fact, that this step is only performed on a subset of data samples, depending on the frequency of occlusion. The final image for the (augmented) flux, together with the image's non-ideal centroid COM, is saved to the dataset.

On the successive **Intersect Reflection Axis with Target Area** step, the ideal centroid is computed along the mathematical algorithm documented in Section 2.2.4 by reflecting the sun position's ray vector along the heliostat's orientation axis and finding the point of intersection with the target area plane. Also, the motor positions from the aligned heliostat kinematics are added to the dataset for later usage.

The loop for the simulation stops if the algorithm has finished the data generation for the last sample and the simulated dataset is complete.

5.2 Pipeline for Training and Calibration

Definition of Calibration Scenarios

Within this study, the image-based calibration strategy is tested against the conventional, angular-loss-based approach. To this end, each dataset scenario with simulated blocking runs through three calibration attempts, each optimizing on a distinct loss function. Table 5.2 defines these strategies for heliostat calibration, their respective loss function and inputs as ground-true and predicted features.

Table 5.2: Comparison of calibration strategies and their optimization targets

Calibration Strategy	**Loss function**	**Ground-True Feature**	**Predicted Feature**
Benchmark	Alignment-Loss (Equation 4.28)	Reflection Axis $\vec{r}_{\mathrm{P}}$ (Equation 2.14)	Reflection Axis $\vec{r}_{\mathrm{G}}$ (Equation 2.14)
Conventional 1	Alignment-Loss	COM-Vector $\vec{r}_{\mathrm{P,COM}}$ (Equation 2.10)	Reflection Axis $\vec{r}_{\mathrm{P}}$ (Equation 2.10)
Conventional 2	Alignment-Loss	COM-Vector $\vec{r}_{\mathrm{P,COM}}$	COM-Vector $\vec{r}_{\mathrm{P,COM}}$
Image-Based	Contour-Loss (Equation 4.46)	Soft Contour C_{P} (Equation 4.33)	Soft Contour C_{G} (Equation 4.33)

The "Benchmark" strategy is performed only once on the "Base" dataset scenario to establish a benchmark performance. This strategy relies on the Alignment-loss (cf. Section 4.2.2) for gradient-based optimization and minimizes the alignment error be-

tween both predicted and ground-true reflection axes. Within this study the results from this approach are viewed as the benchmark for the other strategies, because the exact ground-true reflection axis is not be know for real-world data and can only be approximated through the COM-vector.

The Conventional 1 calibration strategy also relies on alignment-based optimization. The loss function herein uses the flux centroid derived through COM as endpoint for approximating the ground-true reflection axis (i.e. COM-vector). For the predicted feature the actual prediction's reflection axis is used.

The Conventional 2 strategy uses the COM-derived centroids as endpoints for approximating both the prediction's and ground-true reflection axis.

This thesis proposes an Image-Based strategy, which extracts spatial and structural features from the flux image data. These image features rely on the algorithm disclosed in Section 4.2.3 for detection of the focal spots' upper contours. The loss computation then occurs along the mathematical foundation discussed in Section 4.2.4, wherein three sub-terms of losses are combined into a single scalar loss. Furthermore, the algorithm for derivation of the contour-loss uses the "guardrail"-mechanism (cf. Equation 4.45) for stabilized convergence.

Calibration Pipeline with Error Injection

Figure 5.2 presents the algorithmic pipeline used for the data-driven calibration process, where the calibration model is trained on simulated datasets using three distinct strategies (cf. Table 5.2).

The pipeline begins with loading the heliostat surface model including the reflection surface points and normals. Also, the model reads the set of initial kinematic parameters that define the nominal configuration of the kinematic assembly. In the subsequent step, **Offset Kinematic Parameters**, these nominal parameters are altered by adding small randomized offsets. This is a critical procedure to mimic misalignment or deviation between the digital model and the real system. Without such offsets, the calibration process would be trivial —- as the simulated dataset was generated from the same initial parameters, reusing these in training would result in zero tracking error from the first epoch onward.

To formally express this modification, let p_i^{init} be the initial value of the i-th kinematic parameter. The parameter is updated by adding a sampled offset Δp_i, resulting in:

$$\text{Modified Parameter: } p_i = p_i^{\text{init}} + \Delta p_i \tag{5.1}$$

The offset term Δp_i is sampled from a uniform distribution whose bounds are specified per parameter group:

$$\text{Offset Sampling: } \Delta p_i \sim \mathcal{U}(-\varepsilon_i, \varepsilon_i) \tag{5.2}$$

Where ε_i is the maximum deviation allowed for parameter p_i. Table 5.3 displays the maximum deviations that are used per parameter. For example, heliostat translation components in ENU directions have $\varepsilon_i = 0.1$ m, while joint tilts are offset within ± 0.005 rad. This offset-injection procedure ensures that the calibration model is exposed to initial misalignment and learns to adjust its predictions by minimizing the respective loss function over the course of training.

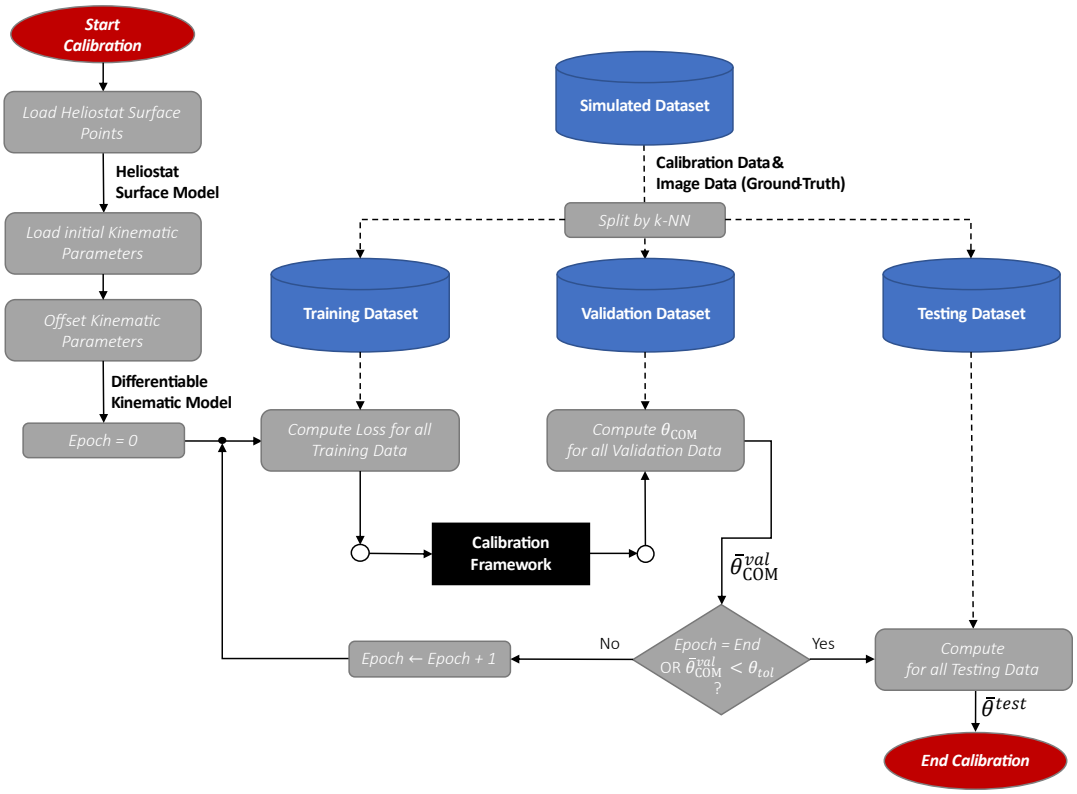


Figure 5.3: Calibration pipeline for model fitting on a simulated dataset with k-NN splitting.

Based on the offset kinematic parameters the differentiable kinematic model is instantiated, wherein the kinematic parameters are stored as PyTorch-Parameters. When creating objects of the Parameter-class, PyTorch automatically enables gradient computation for data Tensor that is passed to it. After the modeling environment is fully initialized, the pipeline enters the training phase.

During training, all data samples that were assigned to the training dataset by k-NN splitting (cf. Section 3.3) are used for loss accumulation which is performed within the **Calibration Framework**. Respectively, the samples from the validation dataset are

used to evaluate model performance at the end of each training epoch by computing the evaluation metric. This evaluation metric is the mean over all samples' COM-accuracies (cf. Section 2.2.4). This cycle of training and validation is repeated until either the epoch has reached the predetermined maximum number of epochs ("*End*"), or the mean accuracy in model prediction on the validation dataset ($\bar{\theta}^{\text{val}}_{\text{COM}}$) is lower than the predetermined error tolerance ($\theta_{\text{tol}} = 0.1$ mrad).

Table 5.3: Maximum deviation bounds for kinematic parameter offsets.
For overview of kinematic parameters refer to Table 1.

Parameter Group	**Parameters**	**Unit**	**Deviation Range ε**
Heliostat Position	Translations [3]	m	± 0.1
First Joint	Translations [3]	m	± 0.05
	Tilts [2]	rad	± 0.005
Second Joint	Translations [3]	m	± 0.05
	Tilts [2]	rad	± 0.005
Concentrator	Translations [3]	m	± 0.05
	Tilts [2]	rad	± 0.005
Actuators $i \in \{1, 2\}$	Offset Radius Shift [2]	rad	± 0.005
	Initial Stroke Length [2]	m	± 0.0
	Joint's Distance Offset [2]	m	± 0.005
	Actuator Radius [2]	m	± 0.0

If one of the two conditions are met, final model evaluation on the prediction's mean tracking error across all samples in the testing dataset is performed. This evaluation metric is represented as the mean tracking error $\bar{\theta}^{\text{test}}$ for the testset .

Calibration Framework

Figure 5.4 presents the detailed framework for image-based calibration relying on the upper contours. It should be noted, that the framework for geometric calibration by minimizing the cosine loss follows the same structure, except that no extraction of upper contours and subsequent processing to a contour-based loss term take place.

The pipeline for data splitting and sampling (cf. Section 5.3) determines which data point from the simulated dataset is processed within the calibration framework. Initially, the **Draw Sample** represents the process for loading the given samples' calibration data. In contrast to the framework for data simulation, the kinematic model is aligned along the indicated motor positions which were previously "recorded" in the simulated environment for that data sample. Afterwards, the model's image output is computed in **Forward Raytracing** wherein the information on the current sun position is processed to generate the predicted flux image.

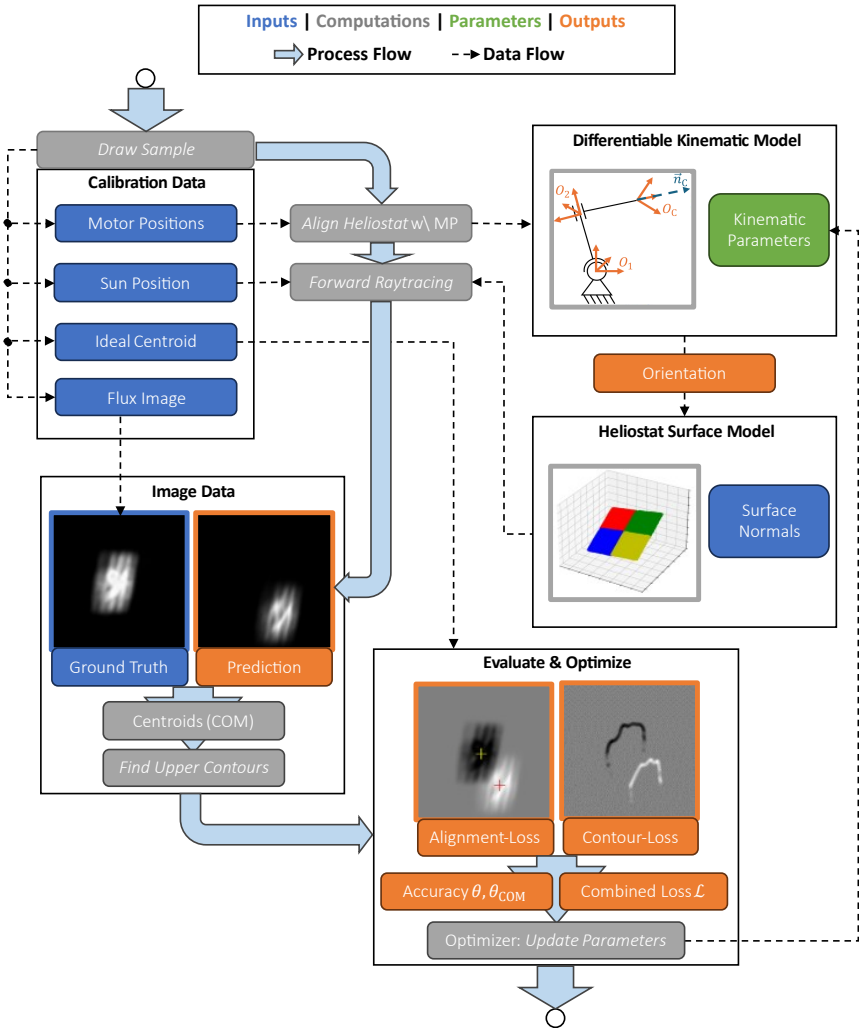


Figure 5.4: Detailed framework for the image-based calibration approach.

Both the prediction and ground-truth image data are subsequently processed in **Compute Centroids** and **Find Upper Contours** (cf. Section 4.2.3). It should be noted, that for the "Conventional 1" calibration strategies, the prediction's COM-centroid is not computed. Instead in this training mode, the model uses the heliostat's orientation for deriving the predicted reflection axis by reflection law. For both conventional strategies and the benchmark, the flux contours are not extracted as the model operates on alignment-based optimization only. The respective process chart for these modes of operation thus resembles the model displayed in Figure 2.8.

Within **Evaluate & Optimize** the model computes the loss depending on the loss function design which is either the geometric Alignment-Loss and / or the Contour-

Loss as presented in Section 6.5. Model training is monitored based on the COM-accuracy and the actual tracking error is computed as the more reliable performance metric for final evaluation. This whole framework is applied to every heliostat in the field and every data sample in the training dataset, while every sample equally contributes to the final loss (single-scalar). Once the calibration model has been evaluated on each sample in the training set the final loss is passed to the PyTorch optimizer for gradient computation. Based on the obtained gradients, the kinematic parameters are updated.

6 Results and Discussion

6.1 Evaluation of the Simulated Scenario-Datasets

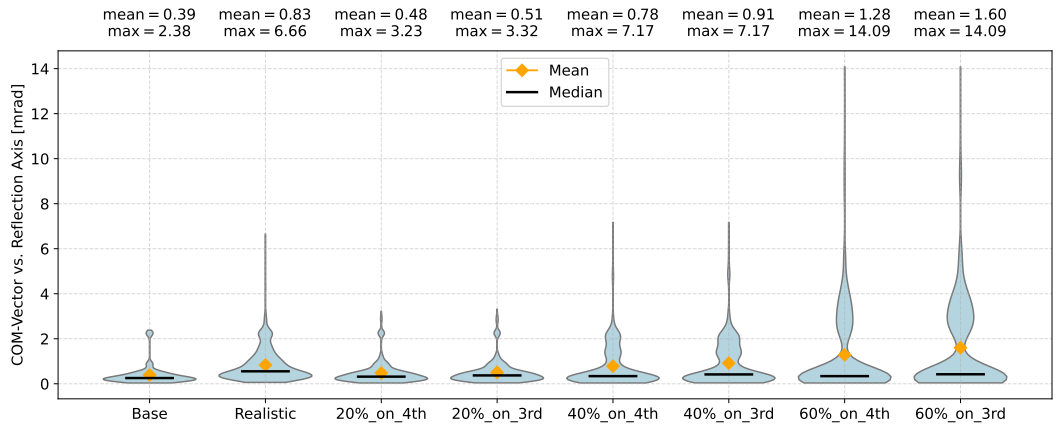
Figure 6.1a displays the range of angular offsets which are inherent to the scenario-datasets due to surface imperfections as well as blocking and shading. For each scenario the offset distribution over its data samples is shown, wherein the y-axis indicates the absolute angular offset between the COM-vector and the true reflection axis. As Section 2.2.4 outlays, a flux recording's corresponding reflection axis is typically approximated by spanning a vector from the heliostat surface center to the flux centroid. The coordinates of the flux centroid is found by determining the focal spot's center-of-mass on the target area.

Any recording of real-world physical data contains some level of noise which disturbs the true measurement. In the context of heliostat calibration with image data this noise is introduced due to inaccurate flux recording, but also due to inaccurate approximation of the corresponding reflection axis. This second effect caused by incomplete knowledge about the physical system and its mapping is commonly known in the ML science as epistemic uncertainty. In the present ML problem uncertainty is introduced in the mapping from the COM-centroid to the true reflection axis, which is the underlying system feature that is being used as ground-truth in training and validation of the differentiable kinematic model.

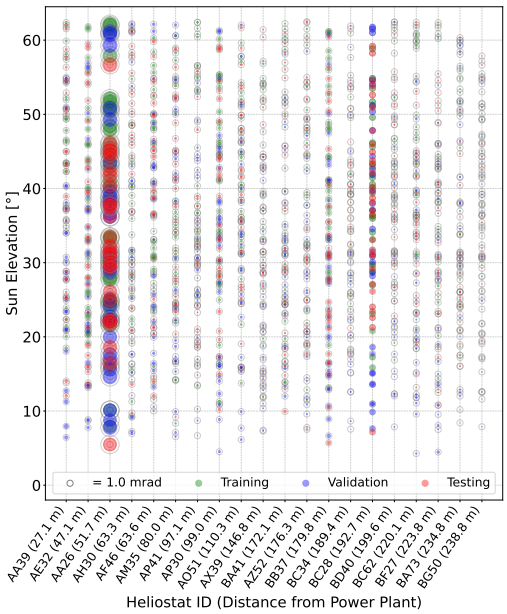
The "Base"-Scenario without blocking and shading displays the uncertainty in the reflection vector which arises due to surface deviations in the geometric shape from the ideal shape. The mean angular offset between the COM-vector and the reflection axis is 0.39 mrad with a maximal offset of 2.38 mrad. There is a noticeable accumulation of data points close to the upper limit, which can be attributed to surface properties of one outlier heliostat.

Figure 6.1b gives the detailed visualization of the angular offsets across heliostats and sun elevation angles. The heliostat mirror with ID AA26 shows relatively large offsets which are likely caused by a combination of two factors: This heliostat mirror's surface is characterized by high deviations from the ideal planar surface while its relative position in the field (cf. Figure 3.2) amplifies the distortions in the flux shape on the target areas. The front positioning of this heliostat close to the receiver towers means that the absolute distance of the flux COM from the reflection axis' intersection point with the target surface translates to a high offset in terms of angular measurement.

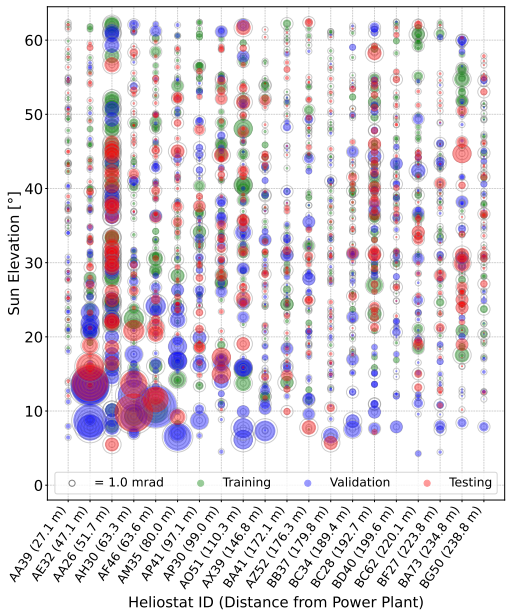
What is more, the relative heliostat position with a significant coordinate component in west direction induces a steep angle of the reflected light rays at their intersection with the target area further distorting the resulting flux shape.



(a) Evaluation of the seven datasets at different levels of blocking and shading.



(b) 'Base'-scenario dataset (no blocking/shading).

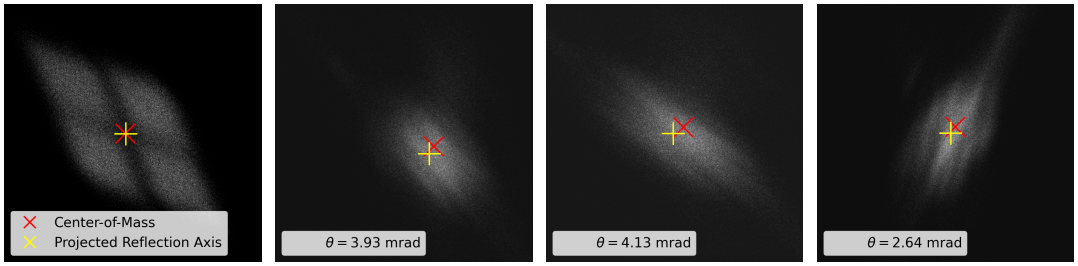


(c) 'Realistic'-scenario dataset with blocking/shading.

Figure 6.1: Comparison of angular offsets between COM-vector and true reflection axis across scenarios and sun elevations. (COM: center-of-mass)

Figure 6.2 compares the angular offset for a planar mirror surface (6.2a), and for the deflectometry-based surface, wherein 6.2b corresponds to the original heliostat position, 6.2c corresponds to a west-translated position, and 6.2d corresponds to an

east-translated position. It is apparent that the heliostat's actual surface geometric causes the COM's offset from the reflection axis, while the translation along the east axis is a less significant factor. The specific surface deviation leading to the elongation of the focal spot is referred to as astigmatism. Astigmatism occurs when a reflective surface has differing radii of curvature in two perpendicular directions, causing rays that lie in different planes to focus at different distances [15].



(a) Planar surface with east pos. at −46.8 m. **(b)** Deflectometry with east pos. at −46.8 m. **(c)** Deflectometry with east pos. at −66.8 m. **(d)** Deflectometry with east pos. at +17.2 m.

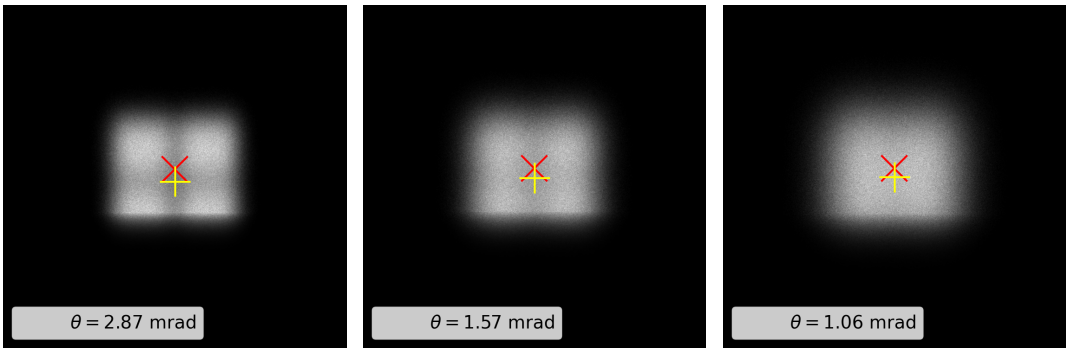
Figure 6.2: Comparison of the center-of-mass and the intersection of the projected reflection axis for a planar surface and deflectometry surface data for heliostat AA26 over three east-axis coordinates. The east coordinate is given relative to the power plant position. The heliostat's north coordinate is constant at 24.9 m. θ is the COM-vector's angle to the reflection axis.

The datasets for the two scenarios at 20% of blocking / shading show a similar distribution as the 'Base'-scenario, with a slight increase in the maximum and mean values for the distribution of angular offsets. Both scenarios are characterized by a similar range of COM-deviations from the reflection axis which is the result of their common factor for blocked / shaded surface. The increase in frequency from every 4th to every 3th sample experiencing partial occlusion results in an increase of the distribution's mean value.

Similar observations are made when considering the distributions of the datasets with blocking / shading at a lever of 40% and 60%. The maximal angular offset for both scenarios at 40% is 7.17 mrad, while the mean value increases for the scenario with the higher frequency of occlusion. As expected, the scenario with occlusion exposure of 60% exhibit the highest variation of COM-offset from the reflection axis ranging up to 14.09 mrad. The mean offset does not increase as strongly, as these extreme values occur rarely, with the large majority of samples showing much lower offsets as is indicated by the black bar for the distributions' median.

The 'Realistic'-scenario which uses a simplified geometric model (cf. Section 4.1.2) to estimate blocking and shading at DLR's STJ exhibits a mean and maximal offset of 0.83 mrad and 6.66 mrad, respectively. The detailed evaluation for this scenario and the underlying uncertainty in the reflection axis is shown in Figure 6.1c. As expected, the modeled effect of blocking and shading generally leads to higher offsets in the COM-vector from the heliostat's reflection axis. Those data samples attributed

to the first row (AA) still only show the angular offsets due to surface deformations as the heliostats in that row cannot be blocked or shaded by front heliostats. The distribution of offsets now displays a dependence on the sun elevation angle which is used as an input to the geometric functions for computing the factor of blocked and shaded mirror surface area. It is noticeable that the highest angular offset occurs at low elevation angles for heliostats which are positioned in the front rows.



(a) North pos. at 75 m. **(b)** North pos. at 125 m. **(c)** North pos. at 175 m.

Figure 6.3: Comparison of the center-of-mass and the intersection of the projected reflection axis for a blocked / shaded heliostat with planar surface as its position in the north-direction is gradually increased. The north coordinate is given relative to the power plant position, while the east coordinate is constant at 0 m. The surface factor for blocking / shading is $\kappa = 0.3$. The given θ is the COM-vector's angle to the reflection axis.

Those heliostats positioned in the back rows exhibit smaller angular offsets - even if the underlying blocking / shading factor is close 1.0 - as an absolute drift of the COM due to partial occlusion translates into a smaller angular offset for longer reflection pathways. Figure 6.3 gives a comparison of the COM's deviation from the reflection axis due to partial flux occlusion at various heliostat positions along the north axis. The factor for the blocked-shaded surface area is kept constant at $\kappa = 0.3$, while the diffusive parameter σ increases at higher distances relative to the tower along the logic presented in Section 4.1.1. It is apparent, that the COM's vector angle to the reflection axis decreases as the heliostat position is shifted further away from the receiver tower.

6.2 Observations on Loss Curves

Figure 6.4 presents the loss curves for the training and validation on the Alignment-Loss during parameter fitting for the 'Base'-scenario dataset. The details on how the Alignment-Loss is computed can be reviewed in Section 4.2.2.

Around epoch 4000, both training and validation losses begin to plateau, indicating convergence of the model parameters. The Adam optimizer is used throughout training which is an adaptive gradient-based optimization algorithm that adjusts learning

rates on a per-parameter basis, enabling efficient training across a wide range of optimization tasks [52].

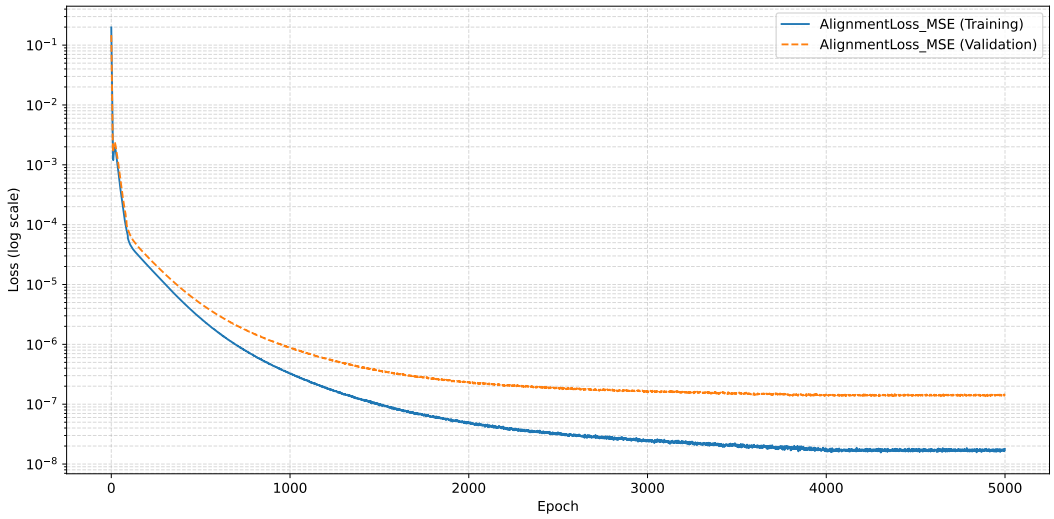


Figure 6.4: Evolution of the Alignment-Loss for training and validation on the 'Base'-scenario dataset. The y-axis uses logarithmic scaling.

Figure 6.5 displays the loss curves for the case when training and validation are performed along the Contour-Loss (cf. section 6.5). As detailed in the corresponding subsections, this image-based calibration strategy begins with a warm-up phase during which the model is trained solely on the standard alignment loss. After the epoch number surpasses $N_1 = 100$, training gradually shifts to the contour-derived loss. Within the transition interval $[N_1, N_2]$, the final Contour-Loss is computed as a weighted combination of both the image-loss terms and the Alignment-Loss , where the contribution of the latter decreases linearly.

Due to the higher computational cost associated with image-based loss terms and their gradients, an early-stopping mechanism is implemented. If the condition for early stopping are met, the model enters testing mode before the maximum number of training cycles are reached, which reduces total runtime. If the model shows insufficient improvement on the validation metric over a defined window, training is stopped prior to reaching the maximum number of epochs. Specifically, early stopping is triggered if the improvement in the validation metric (validation accuracy on COM) over the past 500 epochs is smaller than $\Delta_{\text{min}} = 0.05$ mrad. When this condition is met, the model enters testing mode for the final model evaluation. The maximum number of allowed training epochs for the image-based calibration strategy is 1000. For the example loss curve shown in 6.5, the condition for early-stopping is reached around epoch 710.

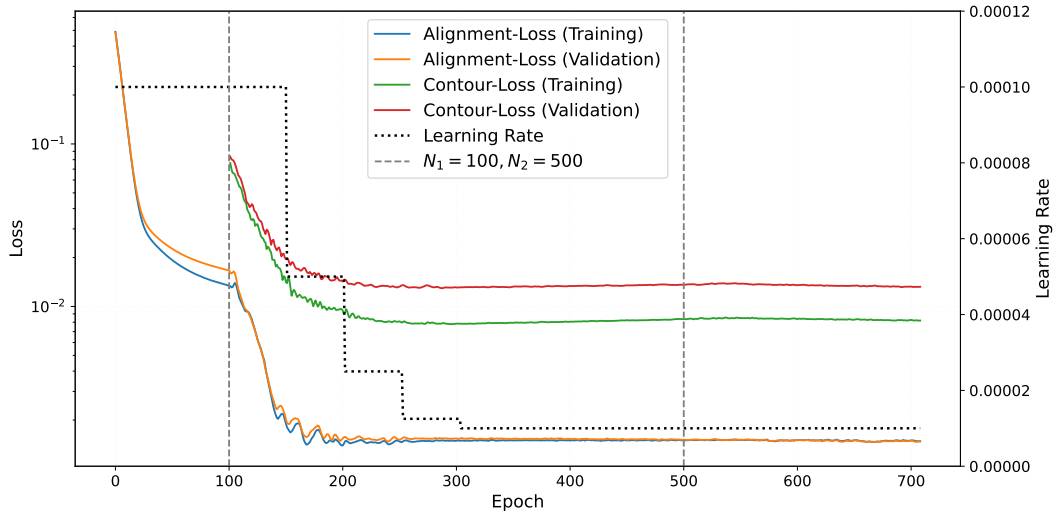


Figure 6.5: Evolution of the Alignment- and Contour-Loss for training and validation on the 'Base'-scenario. The first axis uses logarithmic scaling. The second y-axis indicates the adaptive setting of the learning rate based on PyTorch's ReduceLROnPlateau.

In image-based learning tasks such as heliostat calibration using contour-derived features, the loss landscape is often highly non-convex and exhibits noisy plateaus or slow convergence near local minima. In such settings, fixed learning rate schedules may lead to inefficient training or premature stagnation. To address this, a ReduceLROnPlateau learning rate scheduler was integrated into the training process during optimization on the Contour-Loss. This scheduling strategy, as analyzed by Al-Kababji et al. in [53], dynamically adjusts the learning rate based on validation performance and it is shown to enhance convergence behavior in training regimes involving complex, non-convex loss surfaces such as those in image-based learning.

In the specific implementation for this study, the scheduler monitors the validation loss and automatically reduces the learning rate by a factor of 0.5 if no improvement is observed over a patience window of 50 epochs. A relative tolerance of 0.0 is used as a criterion for minimum progress over the patience window. The minimum learning rate is set to 10^{-5}. This approach allows the optimizer to take larger steps when the model is far from a minimum and progressively finer steps as convergence nears.

6.3 Accuracy of Final Model Predictions

Figure 6.6 shows both the final testset accuracy against COM-vector and the tracking error for when the differentiable kinematic model is trained and tested on the 'Base'-scenario dataset. Attention should be drawn to the fact, that the histograms of equal coloring in both upper and lower halfs of the figure correspond to the same model parameters. Instead, this two-part visualization allows for the comparative evalua-

tion of the models' predictions along two similar, yet distinct metrics. The upper plot evaluates the 'Testset Accuracy on COM' which is defined as the angular error between the ground-true COM-vector and the predicted reflection vector. This quantity is the default metric for evaluating the performance of the kinematic model and the fit of its underlying parameters. Herein, the ground-true reflection axis is used for approximating the original measurement's reflection axis in the field. The lower plot evaluates the model's performance on the actual tracking error which is more accurate than the former metric. It has the disadvantage however, that for real-world datasets of calibration measurements, the ground-true reflection axis is unknown.

The figure visualizes the kinematic model's performance on the testset after converging along four distinct loss plains. Three of these loss variants are computed along the alignment-loss which is discussed in Section 4.2.2.

The red-colored histogram gives the test results for the "Benchmark" calibration strategy, where the model is trained on the alignment-loss using the ground-truth and predicted reflection axis as inputs. This training strategy is not applicable to real-world measured calibration data because the ground-truth of the reflection axis would not be known. However, within this study, which uses simulated datasets, the results of this calibration strategy serve as a benchmark for evaluating the performance of other, practically viable calibration strategies.
The other two implementations of the alignment-loss rely on the COM-vector as ground-truth, which can be extracted from measured flux images. These calibration strategies are different in which vector from the model's outputs is processed as the prediction:

- For the blue-colored results, the reflection vector is obtained from the kinematic model after alignment toward the target and evaluated against the ground-true COM-vector by application of the alignment-loss.
- For the orange-colored data distribution, the COM-vector is derived from the model-output after alignment and subsequent ray-tracing.

Finally, the green-colored data distribution presents the calibration results after fitting the model on the image-based contour-loss (cf. Section 4.2.4), which directly compares spatial features extracted from the upper contours of both the ground-truth and predicted flux images.

Of these four calibration strategies, the variant relying on the alignment-loss with $(\vec{r}_{\mathrm{G,COM}}, \vec{r}_{\mathrm{P}})$ as input vectors achieves the highest accuracy on COM-vector at a mean of 0.46 mrad. This result is expected, as the evaluation metric directly measures the alignment between the same vectors that are optimized during training. The other calibration strategies that are using the alignment-loss show a similar performance along that metric of 0.58 mrad and 0.60 mrad. The contour-loss yields the worst

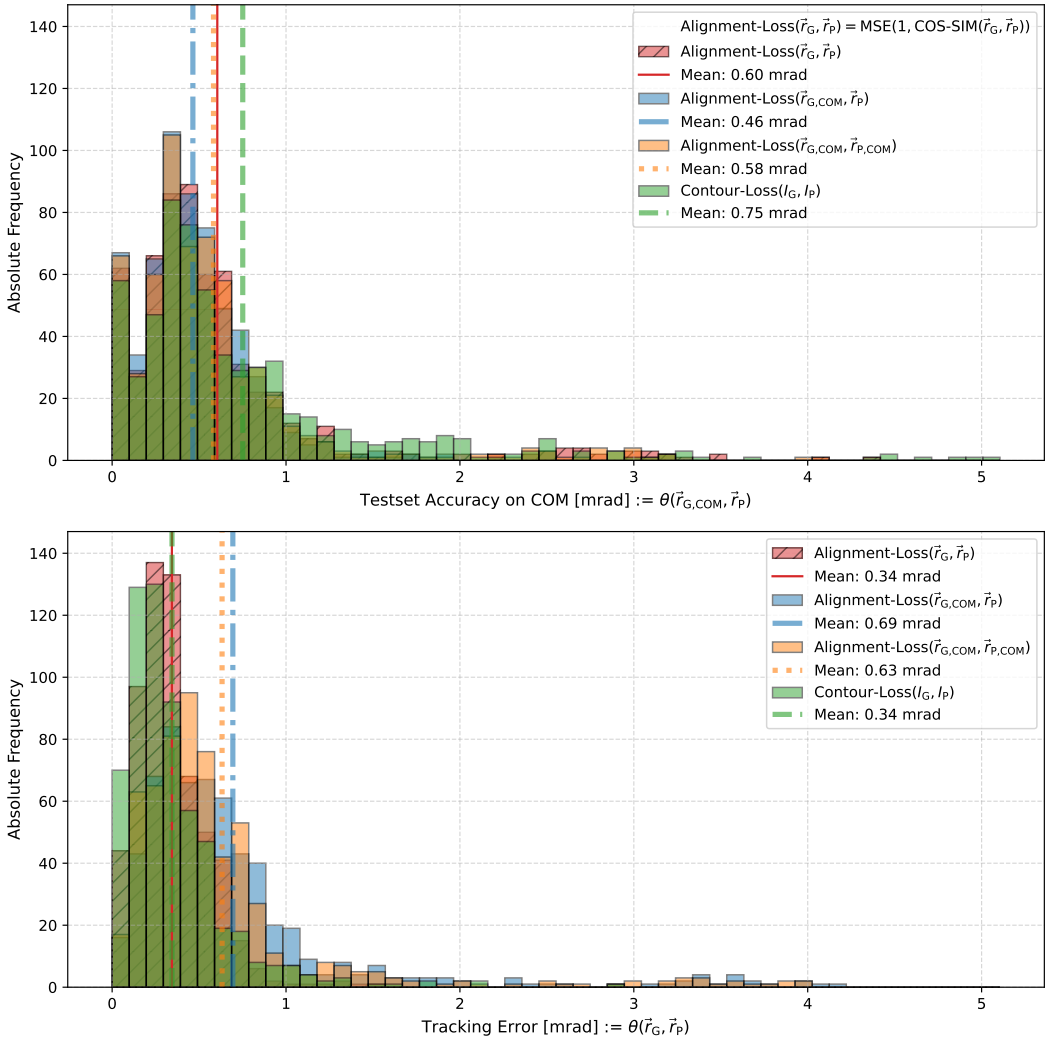


Figure 6.6: Histograms for the "Base"-scenario (no blocking / shading) evaluated on the testset. The upper plot's x-axis gives the angular accuracy of the model's reflection axis prediction ($\vec{r}_{\mathrm{P}}$) against the ground-truth center-of-mass-vector ($\vec{r}_{\mathrm{G,COM}}$). The lower plot's x-axis indicates the tracking error which measures the predicted against the ground-truth reflection axis. For model training, the geometric alignment-loss and the image-based contour-loss were used. Further comparison is drawn between the model's performance when passing the COM-vector and/or reflection axis as inputs to the alignment loss. $I_{\mathrm{G}}, I_{\mathrm{P}}$ denote the flux image ground-truth and prediction. $\vec{r}_{\mathrm{G}}$ is the ground-true reflection vector and $\vec{r}_{\mathrm{P,COM}}$ is the predictions COM-vector.

mean accuracy on COM-vector of 0.75 mrad.

The accuracy on COM-vector is not the optimal metric for evaluating the performance after calibration is completed. As discussed in Section r2.2.4, the "actual" tracking error is the appropriate metric for evaluating the kinematic model after parameter training, as it directly quantifies the model's ability to predict the heliostat's surface orientation vector after calibration. When the four calibration strategies are compared based on tracking error, the ranking changes relative to that obtained from the test accuracy metric. The two best-performing approaches under this criterion are the benchmark alignment-loss and the image-based contour-loss, both achieving a mean tracking error of 0.34 mrad. The variants of the alignment-loss achieve a similar value for the mean tracking error of 0.69 mrad and 0.63 mrad. These calibration strategies exhibit noticeable occurrences of high tracking errors in the range of 3.0 - 4.0 mrad.

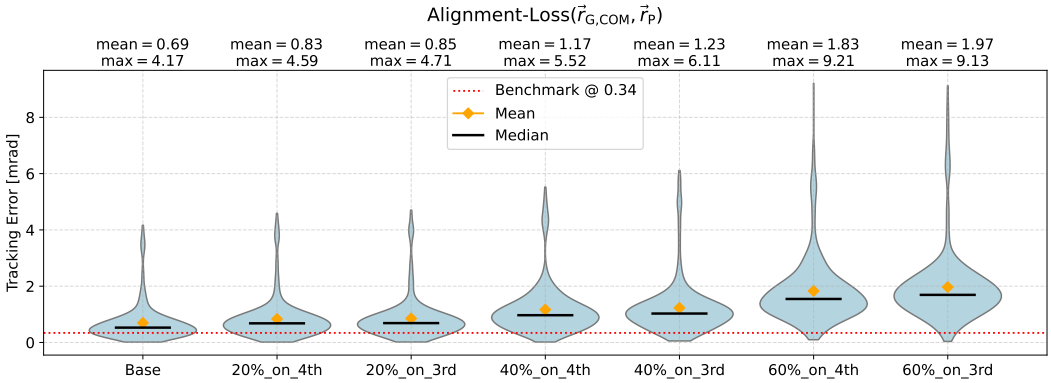
In the following considerations the results on the accuracy on COM-vector are not illustrated as the tracking error is considered to be the more meaningful metric. The tracking error is superior to serve as a performance indicator for the studied calibration approaches.

Figure 6.7 presents the distribution of tracking errors achieved by the calibration strategies which use the alignment-loss and are trained and tested on seven datasets. The benchmark mean accuracy, which was achieved by training on the alignment-loss on the ground-truth reflection axis, is given as reference. A model is always tested on the same dataset it was trained on. The model's performance is compared across the "Base"-scenario and the six scenarios with fixed factors for blocking / shading. Both approaches relying on the alignment-loss exhibit similar distribution of final tracking errors across all scenarios. As expected, the models' ability to predict the reflection axis decreases as the parameters for blocking and shading (κ, f_{BS}) are increased. As discussed in Section 6.1 the angular offset between the COM-vector deviates from the reflection axis more strongly at higher levels of surface occlusion. Due to this effect, the calibration model receives false training stimuli when the alignment-loss is used.

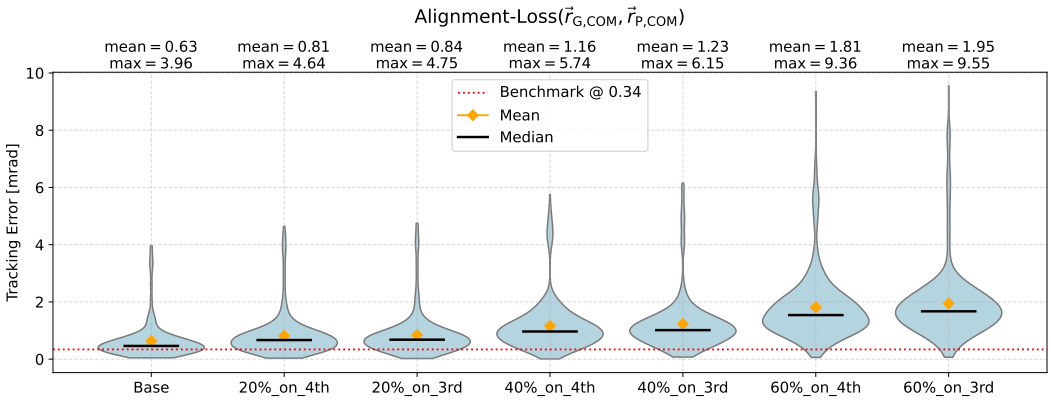
The overall distribution of tracking errors over the scenarios is very similar for both calibration approaches. The detailed results for the 'Base'-dataset (cf. Figure 6.6) under ideal conditions were discussed above. As the severity and frequency of artificial blocking and shading increases, the performance of both strategies gradually deteriorates, which is indicated by a widening of the violin plots and a moderate rise in both mean and maximum tracking errors. Notably, however, neither strategy exhibits catastrophic tracking performances in terms of the statistical mean of model accuracy, even for the most extreme scenarios of flux data distortion. The mean tracking error for the final reflection axis prediction increases moderately to above 0.8 mrad for $\kappa = 20\%$, to around 1.2 mrad for $\kappa = 40\%$, to approximately 1.8 or 2.0 mrad for the scenarios with $\kappa = 60\%$ at f_{BS}-frequencies of $\frac{1}{4}$ or $\frac{1}{3}$ flux occlusions,

respectively.

Meanwhile, the increase in the maximum tracking errors is more severe as can be observed on the scenarios at an higher occurrence of blocked/shaded data samples: This metric expresses a rise from around 4.75 mrad for $\kappa = 20\%$, to around 6.15 mrad for $\kappa = 40\%$, to 9.13 mrad or 9.55 mrad for $\kappa = 60\%$ when the model uses the reflection axis (c.f. figure 6.7a) or the COM-vector (cf. figure 6.7b) in evaluating its predictions, respectively.



(a) Distribution of final tracking errors when the alignment-loss evaluates the prediction's reflection axis against the ground-true COM-vector.



(b) Distribution of final tracking errors on testset when the alignment-loss evaluates the prediction's COM-vector against the ground-true COM-vector.

Figure 6.7: Comparison of two calibration strategies which both are trained on the alignment-loss, but evaluate the model's performance along its prediction for (a) the reflection axis and (b) the COM-vector. The y-axis indicates the final tracking error that compares the prediction's reflection axis against the ground-true reflection axis, which typically is not known for real-world calibration measurements.

An interesting conclusion from the comparison between the two calibration strategies

is, that the strategy which is trained on the predicted reflection axis generally performs better in terms of the maximum tracking errors, while the strategy relying on the predicted COM-vector performs better on the mean of model accuracies. Only for the "Base"-scenario one strategy performs better in both metrics. The approach using the alignment loss on both predicted and ground-true COM-vectors achieves higher accuracies both in terms of mean and maximum values. The reason for this increased performs in comparison to the strategy based on predicting the reflection axis is, that the center-of-mass for non-distorted (i.e. ideal) flux images provides better training stimuli as the ground-true information is also COM-based. As concluded from figure 6.1b, inferring the reflection axis from the COM-vector introduces an error – even when the simulated dataset does not include fragments from artificial blocking / shading. Averaged over the data samples this also holds true for those scenarios which do include a certain portion of partially distorted flux recordings. However, this COM-based prediction shows lower performance on individual data samples, which are understood to represent outliers.

Figure 6.8 shows the distributions of final tracking errors in testing the calibration strategy which relies on the image-based contour-loss for fitting the kinematic parameters. Again, the evaluation results for seven scenarios are compared, including the "Base"-scenario and six scenarios which assume a constant level of blocking / shading (κ). The main observable difference to the previous results for training with the alignment-loss is, that the contour-loss reliably achieves higher accuracies over all cases of flux occlusion. This conclusion is reached by observing the mean and maximum tracking errors for each scenario dataset. Both metrics remain at significant lower levels compared to those performance indicators seen in Figure 6.8.

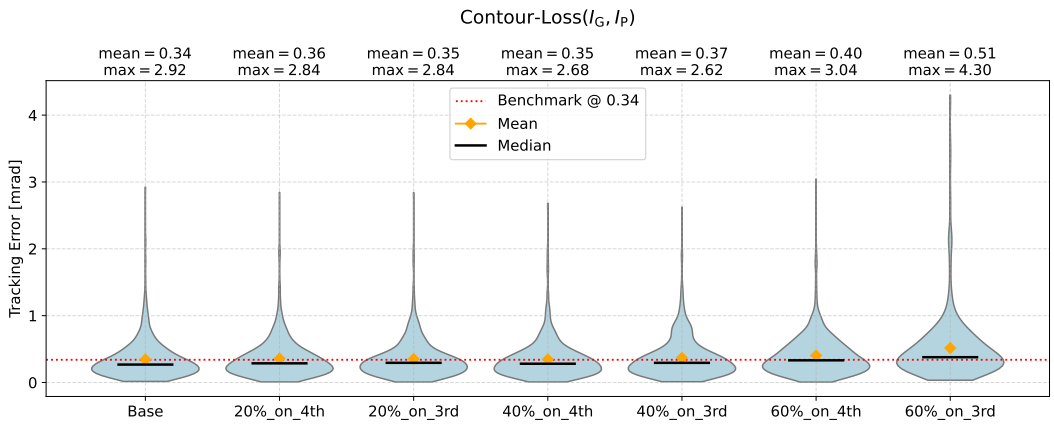


Figure 6.8: Distribution of final tracking errors on testset when the model is trained on the image-based contour-loss. The final mean accuracy is much closer to benchmark value at 0.34 mrad compared to when alignment-loss is used for fitting the kinematic parameters.

It is quite apparent, that the contour-loss is much more robust to distortions in the

training data which are introduced by artificially adding blocking and shading in the post-processing of the flux images. The mean tracking error throughout all scenarios stays way below 1 mrad while the outliers in model accuracy are also prone to low variance. For the scenarios with $\kappa = 20\%$ and $\kappa = 40\%$ the mean tracking error deviates only slightly from the benchmark value at 0.34 mrad, which the contour-based calibration approach is able to achieve on the "Base"-dataset. On the last two scenarios at $\kappa = 60\%$ the model predicts the reflection axis with an average accuracy of 0.40 mrad and 0.51 mrad for the scenarios at f_{BS}-frequencies of $\frac{1}{4}$ and $\frac{1}{3}$ flux occlusions, respectively. The maximum value for the tracking errors actually decreases from the "Base"-dataset to the experiments performed on the datasets at $\kappa = 20\%$; and then it decreases further on the datasets at $\kappa = 40\%$. For the last two scenarios at $\kappa = 60\%$ the kinematic model which is trained on the contour-loss shows its lowest accuracy at outliers with 3.04 mrad and 4.30 mrad. The latter achievement on this performance metric indicates that the contour-loss becomes less reliable only for the extreme scenario of severe blocking / shading at high frequencies of occurrences in the underlying dataset. This is caused by the increasing degradation of the flux feature which is used for optimization: the upper contour. At levels blocking / shading as high as $\kappa = 60\%$, pixels in the upper contour are attenuated, adding noise to the chosen feature. Thus, the performance of this optimization strategy suffers, although it still achieves significant improvements in comparison to conventional alignment.

Finally, the histograms to tracking errors observed for each calibration strategy when they are trained and tested on the "Realistic"-scenario dataset is shown in figure 6.9. Again, the mean prediction accuracy that was achieved by the alignment-loss using both the ground-true and predicted reflection axis is also plotted as a benchmark value. The contour-loss achieves the benchmark with a mean tracking error of only 0.34 mrad. Also, the visual evaluation of the results indicates a low frequency of outlier samples with elevated tracking errors, as there are only two testset instances with $\theta > 2.0$ mrad.

The two calibration approaches relying on the alignment-loss for parameter training both achieve a mean tracking error of 1.09 mrad. Also, their histograms display significant number of occurrences in the testset with accuracies over 3.0 mrad and outliers around 4.0 mrad. With the present distributions tracking errors, it is not possible to conclude that neither of the calibration strategies which use the prediction's reflection axis and the COM-vector yield better results in terms of final accuracy on this "Realistic"'-scenario dataset. It is clear however, that the image-based loss function – which is trained on the features extracted from the flux' upper contours – is superior to the alignment-loss function in achieving accurate calibration of the kinematic model.

From Figure 6.8 it can be concluded, that the contour-loss provides constructive training stimuli even at moderate to high distortion levels in flux image data. By evaluation of the results in Figure 6.9 it is shown, that this specific loss design is also robust to varying levels of flux occlusion which are considered in dataset generation

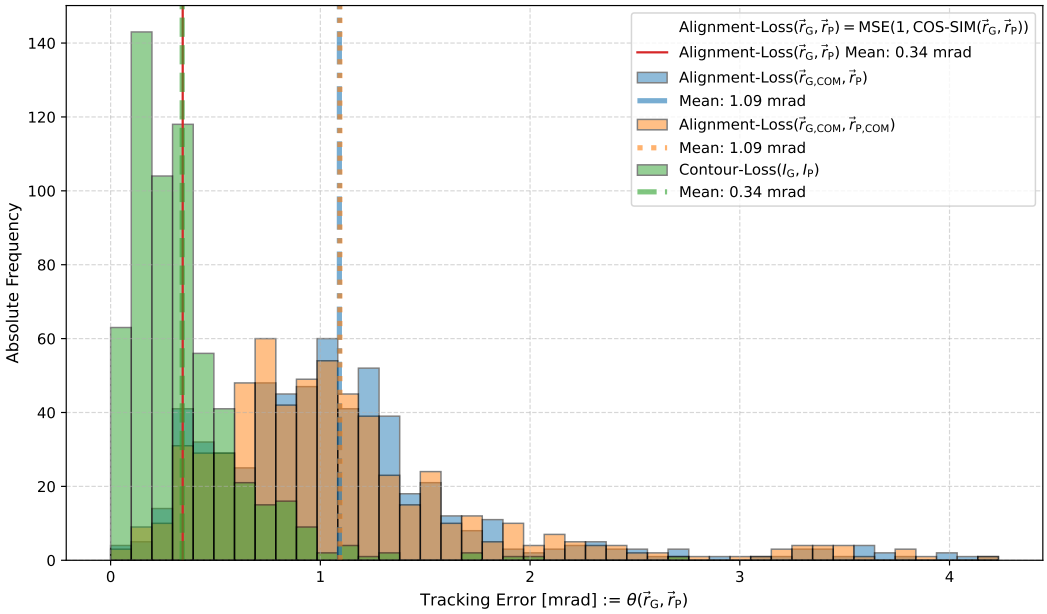


Figure 6.9: Histograms for final tracking errors evaluated on the testset for the 'Realistic'-scenario using a geometric model to approximate blocking and shading at DLR's STJ.

by approximating the effects of realistic blocking and shading throughout the range of heliostat and sun positions.

7 Conclusion and Outlook

7.1 Summary of Results

Within this thesis, an approach to image-based calibration is developed by devising a suitable ML-framework which receives training stimuli through a loss function derived from structural features of the flux distribution – specifically, its upper contour. To the knowledge of the author the here documented attempt is the first of its kind in the context of heliostat calibration. The proposed image-based loss function achieves stable convergence by matching along the flux's upper contour – a feature which is extracted by applying a convolution with a vertical Sobel-kernel. As framework for the loss function a multi-faceted approach is chosen building on three concepts: coarse, fine and gravity-based alignment. The optimal parameter configuration for feature extraction and weighing of the loss terms was found for the studied simulation environment by Bayesian optimization.

This thesis initially debates the crucial distinction between the centroid-based approximation of a calibration measurement's reflection axis and the heliostat's actual reflection axis. Deviations in the geometric shape of concentrators from their ideal shape in the simulated scenarios lead to an error in the approximated COM-accuracy to the actual tracking error of up to 2.5 mrad. The overall distribution of this offset in accuracy metrics however, was below 0.5 mrad for the majority of samples. The observed outliers can be attributed to one individual heliostat affected by astigmatism and positioned in close proximity to the receiver. The additional offset introduced in the "Realistic" scenario with blocking and shading, displayed significant increase in the error of the COM-metric. This epistemic uncertainty in the reflection axis means that the conventional optimization strategies receive false training stimuli which leads to decreasing performances of such models over the increasing severity of flux degeneration due to blocking and shading.

Relying on the upper contour as feature for optimization is advantageous due to its suggested consistency in flux measurement recorded in the camera-target method. As suggested in the literature ([30, 28]) and supported through simulations using a simplified geometric model (cf. Section 4.1.2) it is shown, that heliostat-to-heliostat blocking and shading occur over a wide range of sun angles at significant severity. These quantitative results were validated to a large extent by comparison with results obtained from HFCAL – a sophisticated software tool. This thesis assumes that throughout the degenerative effects of blocking / shading on the flux measure-

ment, the upper contour is generally preserved, thus offering a reliable feature for alignment. Through observations obtained on calibration experiments on artificially created datasets which were simulated under different scenario settings of blocking and shading, this hypothesis is supported.

In the evaluation of the final accuracy achieved by applying the image-based approach it is shown, that the novel strategy achieves benchmark-level accuracy for almost all the considered scenarios including severe levels of blocking / shading. Only for the last scenarios with 60% occlusion a slight impairment in the reported accuracy is observed, while the image-based calibration strategy still performs much more reliably than conventional optimization. The results are especially relevant for the "Realistic" dataset which contains a scenario of blocking and shading that should reproduce real-world conditions. When this dataset is used in training, the contour-loss achieves benchmark-level accuracy at 0.34 mrad while conventional strategies relying on vectorial alignment show a final accuracy of 1.09 mrad.

7.2 Recommendations for Future Research

This thesis shows promising results for the potential of novel image-based calibration approaches. At the same time it is evident that further research is required before such algorithms become applicable to the industry.

Within this thesis a function for artificial blocking and shading is used which is approximates the severity of flux degeneration based on overly-simplified assumptions in the 2D-plane. Further advancements in the area of differentiable ray tracing simulation software should encompass the consideration of heliostat-to-heliostat blocking and shading. This would enable a more complete prediction of focal spots, thus offering a more realistic environment for the development of image-based calibration frameworks.
Additional progress in relevant simulation techniques is anticipated from advancements in simulation techniques, particularly through the use of data-driven surface prediction methods. Such an approach is proposed by Lewen et al. (2024) through inverse Deep Learning Raytracing (iDLR) [54] and was recently validated on real-world measurement data [16]. In the near future, such methods could enable accurate surface reconstruction for heliostat mirrors from flux image data only, alleviating the requirement for laborious deflectometry-like measurements. The integration of scalable surface reconstruction, state-of-the-art hybrid disturbance models (cf. Pargmann 2023 [5]), and image-based loss functions offers a compelling path forward—bringing the field closer to achieving high-precision, data-efficient heliostat calibration.

The application of the proposed framework to measured flux image data represents the logical next step in advancing image-based heliostat calibration. While this thesis

focused on synthetic datasets to enable controlled blocking and shading scenarios with known reflection axes, this setup provided a necessary foundation for validating the core algorithm under ideal conditions. Building on this proof-of-concept, future research should focus on adapting the feature extraction process and refining the loss formulation to handle the higher noise levels and variability inherent in real-world flux images. In particular, it remains to be demonstrated how image-based loss functions – such as the proposed contour-based alignment – perform when integrated into a training pipeline that relies on measured calibration data. Addressing this will require a more robust and adaptive pre-processing pipeline, capable of extracting structurally meaningful features under realistic conditions. Nevertheless, the promising results presented in this work strongly support the feasibility of this transition. An initial implementation of the algorithm is planned at the research facility STJ, where its effectiveness on measured data can be evaluated. Success in this context could mark the beginning of a new class of calibration methods grounded in computer vision and flux-feature-driven alignment—offering a conceptually novel and potentially more resilient alternative to conventional centroid-based approaches.

Acronyms

ANN Artificial Neural Network. 11, 12

AP Aim Point. 17, 44

COM Center of Mass. 6, 17, 18, 22, 34, 39, 41, 44–46, 48–51, 53, 54, 56, 57, 59, 64

CSP Concentrating Solar Power. 1

DLR German Center for Aerospace. 12, 20

ENU East-North-Up Coordinate System. 4, 6, 39, 47

GPU Graphics Processor Unit. 12

KIT Karlsruhe Institute of Technology. 12, 20

LCOE Levelized Cost of Electricity. 1

ML Machine Learning. 3, 10, 17, 22, 51, 64

MP Motor Positions. 15, 17

NREL National Renewable Energy Laboratory. 1

SDF Soft Distance Field. 38

ST Solar Tower. 1, 3, 4, 7, 9, 10, 17, III, IV

STJ Solar Tower Jülich. 3, 17, 20–22, 66

UAV Unmanned Aerial Vehicle. 8

Bibliography

[1] International Energy Agency, "Global energy review 2025," https://www.iea.org/reports/global-energy-review-2025, 2025, iEA, Paris. Licence: CC BY 4.0.

[2] United Nations Framework Convention on Climate Change, "Paris agreement," 2015, adopted at COP21, Paris, 12 December 2015. [Online]. Available: https://unfccc.int/files/essential_background/convention/application/pdf/english_paris_agreement.pdf

[3] European Parliament, European Parliamentary Research Service, "Climate neutrality by 2050: Zeroing in on the target," European Parliament, Tech. Rep. EPRS_BRI(2024)767182, 2024, briefing. [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/BRIE/2024/767182/EPRS_BRI(2024)767182_EN.pdf

[4] A. H. Alami, A. G. Olabi, A. Mdallal, A. Rezk, A. Radwan, S. M. A. Rahman, S. K. Shah, and M. A. Abdelkareem, "Concentrating solar power (csp) technologies: Status and analysis," *International Journal of Thermofluids*, vol. 18, p. 100340, 2023.

[5] M. Pargmann, J. Ebert, D. M. Quinto, R. Pitz-Paal, and S. Kesselheim, "In-situ solar tower power plant optimization by differentiable ray tracing," 2023.

[6] D. S. Codd, A. Gray, M. Mehos, and C. K. Ho, "Roadmap to advance heliostat technologies for concentrating solar-thermal power," National Renewable Energy Laboratory (NREL), Tech. Rep. NREL/TP-5500-81606, 2022, technical Report, September 2022. [Online]. Available: https://www.nrel.gov/docs/fy22osti/81606.pdf

[7] J. C. Sattler, M. Röger, P. Schwarzbözl, R. Buck, A. Macke, C. Raeder, and J. Göttsche, "Review of heliostat calibration and tracking control methods," *Solar Energy*, vol. 207, pp. 110–132, 2020.

[8] M. Pargmann, D. Maldonado Quinto, P. Schwarzbözl, and R. Pitz-Paal, "High accuracy data-driven heliostat calibration and state prediction with pretrained deep neural networks," *Solar Energy*, vol. 218, pp. 48–56, 2021.

[9] M. Leibauer, "Enhancing heliostat calibration in solar tower power plants: A novel dataset evaluation metric and hybrid kinematic modelling technique," Master's thesis, RWTH Aachen University, Jülich, Germany, 2023, german

Aerospace Center (DLR), Institute of Solar Research. [Online]. Available: https://elib.dlr.de/202957/

[10] L. T. E. Sievers, M. Pargmann, D. Maldonado Quinto, and B. Hoffschmidt, "End-to-end sensitivity analysis of a hybrid heliostat calibration process involving artificial neural networks," *Solar Energy*, vol. 287, p. 113219, 2025.

[11] M. Pargmann, M. Leibauer, V. Nettelroth, D. Maldonado Quinto, and R. Pitz-Paal, "Questioning the reliability of open-loop calibration methods: Introducing a robust data sampling for year-round high accuracy," *Solar Energy*, vol. 286, p. 113094, 2025.

[12] A. Boretti, S. Castelletto, and S. Al-Zubaidy, "Concentrating solar power tower technology: present status and outlook," *Nonlinear Engineering*, vol. 8, no. 1, pp. 10–31, 2019.

[13] R. A. Meyers, Ed., *Encyclopedia of sustainability science and technology*, living edition ed., ser. Springer reference. New York, NY: Springer, 2019. [Online]. Available: https://link.springer.com/referencework/10.1007/978-1-4939-2493-6

[14] German Aerospace Centre (DLR), "Solar towers jülich," accessed: 2025-06-10. [Online]. Available: https://www.dlr.de/en/research-and-transfer/research-infrastructure/solar-towers-juelich

[15] W. A. Landman, "Optical performance of the reflective surface profile of a heliostat: Dissertation presented for the degree of doctor of philosophy in the faculty of engineering at stellenbosch university," Ph.D. dissertation, 2017.

[16] J. Lewen, M. Pargmann, J. Jitsev, M. Cherti, R. Pitz-Paal, and D. M. Quinto, "Scalable heliostat surface predictions from focal spots: Sim-to-real transfer of inverse deep learning raytracing." [Online]. Available: http://arxiv.org/pdf/2504.03712v1

[17] J. Freeman, B. Shankar, and G. Sundaram, "Inverse kinematics of a dual linear actuator pitch/roll heliostat," ser. AIP Conference Proceedings. Author(s), 2017, p. 030018.

[18] "An integration model for flux density distribution formed by a heliostat," *Applied Sciences*, vol. 12, no. 20, p. 10191, 2022.

[19] M. Guo, Z. Wang, J. Zhang, F. Sun, and X. Zhang, "Accurate altitude–azimuth tracking angle formulas for a heliostat with mirror–pivot offset and other fixed geometrical errors," *Solar Energy*, vol. 85, no. 5, pp. 1091–1100, 2011.

[20] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 1st ed. London: Springer Science & Business Media, 2010.

[21] Kenneth W. Stone and Scott A. Jones, "Analysis of solar tower heliostat tracking error sources," *Sandia National Laboratories*, no. SAND99-0239C, 1999.

[22] Kenneth W. Stone, "Automatic heliostat track alignment method," Patent US4 564 275, 1986.

[23] M. Röger, C. Prahl, and S. Ulmer, "Fast determination of heliostat shape and orientation by edge detection and photogrammetry," *Journal of Solar Energy Engineering*, vol. 132, no. 2, pp. 1–7, 2010.

[24] J. J. Krauth, C. Happich, N. Algner, R. Broda, A. Kämpgen, A. Schnerring, S. Ulmer, and M. Röger, "Heliopoint – a fast airborne calibration method for heliostat fields," *Journal of Solar Energy Engineering*, vol. 146, no. 6, 2024.

[25] A. Kribus, I. Vishnevetsky, A. Yogev, and T. Rubinov, "Closed loop control of heliostats," *Energy*, vol. 29, no. 5-6, pp. 905–913, 2004.

[26] M. Collins, D. Potter, and A. Burton, "Design and simulation of a sensor for heliostat field closed loop control," ser. AIP Conference Proceedings. Author(s), 2017, p. 030009.

[27] M. Pargmann, J. Ebert, M. Götz, D. Maldonado Quinto, R. Pitz-Paal, and S. Kesselheim, "Automatic heliostat learning for in situ concentrating solar power plant metrology with differentiable ray tracing," *Nature communications*, vol. 15, no. 1, p. 6997, 2024.

[28] A. A. Rizvi and D. Yang, "A detailed account of calculation of shading and blocking factor of a heliostat field," *Renewable Energy*, vol. 181, pp. 292–303, 2022.

[29] M. Kleemann and M. Meliß, *Regenerative Energiequellen.* Berlin: Springer, 1993.

[30] I. Abunajeeb, G. Stein, and C. Rotschild, "Heliostat accurate control method robust under shading and blocking," *Energy Reports*, vol. 13, pp. 4087–4094, 2025.

[31] Y.-H. Zhuang, C.-Z. Yuan, C.-Y. Lin, V. Peconkova, and M.-F. R. Lee, "Calibration for kinematic control of differential-drive mobile robots: A machine learning approach," in *2023 International Conference on Advanced Robotics and Intelligent Systems (ARIS).* IEEE, 2023, pp. 1–6.

[32] F. L. Tagliani, N. Pellegrini, and F. Aggogeri, "Machine learning sequential methodology for robot inverse kinematic modelling," *Applied Sciences*, vol. 12, no. 19, p. 9417, 2022.

[33] S. Ulmer, T. März, C. Prahl, W. Reinalter, and B. Belhomme, "Automated high resolution measurement of heliostat slope errors," *Solar Energy*, vol. 85, no. 4, pp. 681–687, 2011.

[34] Holger H. Rapp, "Reconstruction of specular reflective surfaces using auto-calibrating deflectometry," 2013.

[35] T. März, C. Prahl, S. Ulmer, S. Wilbert, and C. Weber, "Validation of two optical measurement methods for the qualification of the shape accuracy of mirror panels for concentrating solar systems," *Journal of Solar Energy Engineering*, vol. 133, no. 3, 2011.

[36] "Paint – fair concentrating solar power plant database." [Online]. Available: https://paint-database.org/

[37] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual understanding of convolutional neural network- a deep learning approach," *Procedia Computer Science*, vol. 132, pp. 679–688, 2018.

[38] Esraa Zeki Mohammed, Hussain Salih Akbar, "Automated identification of sunspots area size using sobel edge detection and image histogram," *American Journal of Science and Technology*, no. 2(6), pp. 329–334, 2016.

[39] M. Pargmann, M. Leibauer, V. Nettelroth, D. M. Quinto, and R. Pitz-Paal, *It is Not About Time - A New Standard for Open-Loop Heliostat Calibration Methods*, 2023.

[40] M. Kuhl, M. Pargmann, M. Cherti, J. Jitsev, D. Maldonado Quinto, and R. Pitz-Paal, "Flux density distribution forecasting in concentrated solar tower plants: A data-driven approach," *Solar Energy*, vol. 282, p. 112894, 2024.

[41] M. Kuhl, M. Pargmann, M. Cherti, J. Jitsev, D. Maldonado Quinto, and R. Pitz-Paal, "In-situ unet-based heliostat beam characterization method for precise flux calculation using the camera-target method," *Solar Energy*, vol. 279, p. 112811, 2024.

[42] Y. T. Quek, W. L. Woo, and T. Logenthiran, "Dc equipment identification using k-means clustering and knn classification techniques," in *2016 IEEE Region 10 Conference (TENCON).* IEEE, 2016, pp. 777–780.

[43] D. Li, T. Jiang, and M. Jiang, "Norm-in-norm loss with faster convergence and better performance for image quality assessment," pp. 789–797, 2020. [Online]. Available: http://arxiv.org/pdf/2008.03889v1

[44] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2018.

[45] Y. Fa, C. Zhang, Y. Song, and R. Zhang, "Model analysis and optimization design strategy for a class of heliostat field systems," in *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS).* IEEE, 2024, pp. 297–302.

[46] Peter Schwarzbözl , Mark Schmitz and Robert Pitz-Paal, "Visual hflcal – a software tool for layout and optimization of heliostat fields," *SolarPACES*, 2009.

[47] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1395–1403.

[48] G. Jabour, S. Muravyov, and V. Efimova, "Layerwise image vectorization via bayesain-optimized contour," in *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2025, pp. 831–838.

[49] J. Bertels, T. Eelbode, M. Berman, F. Maes, D. Vandermeulen, and M. D. Bruyne, "Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice," in *MICCAI 2019*, 2019, pp. 92–100.

[50] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings of 3DV*, 2016, pp. 565–571.

[51] Y. Tian, Z. Zhang, B. Zhao, L. Liu, X. Liu, Y. Feng, J. Tian, and D. Kou, "Coarse-to-fine prior-guided attention network for multi-structure segmentation on dental panoramic radiographs," *Physics in Medicine & Biology*, vol. 68, no. 21, p. 215010, 2023.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." [Online]. Available: http://arxiv.org/pdf/1412.6980v9

[53] A. Al-Kababji, F. Bensaali, and S. P. Dakua, "Scheduling techniques for liver segmentation: Reducelronplateau vs onecyclelr." [Online]. Available: http://arxiv.org/pdf/2202.06373v1

[54] J. Lewen, M. Pargmann, M. Cherti, J. Jitsev, R. Pitz-Paal, and D. M. Quinto, "Inverse deep learning ray tracing for heliostat surface prediction." [Online]. Available: http://arxiv.org/pdf/2408.10802v1

A. Kinematic Parameters Table

Table 1: Geometric parameters used within the kinematic model.

Parameter Group	Parameters	Unit	Description
Heliostat Position	$\Delta e_0,\ \Delta n_0,\ \Delta u_0$	m	Initial translations in east, north, and up direction.
First Joint	$\Delta e_1,\ \Delta n_1,\ \Delta u_1$	m	Joint translations on east, north and up axis.
	$\Delta\varepsilon_{\mathrm{n},1},\ \Delta\varepsilon_{\mathrm{u},1}$	rad	Joint tilts around the north and up axis. 5
Second Joint	$\Delta e_2,\ \Delta n_2,\ \Delta u_2$	m	Joint translations on east, north and up axis.
	$\Delta\varepsilon_{\mathrm{e},2},\ \Delta\varepsilon_{\mathrm{n},2}$	rad	Joint tilts around the east and north axis.
Concentrator	$\Delta e_{\mathrm{C}},\ \Delta n_{\mathrm{C}},\ \Delta u_{\mathrm{C}}$	m	Concentrator translations on east, north and up axis.
	$\Delta\varepsilon_{\mathrm{e,C}},\ \Delta\varepsilon_{\mathrm{n,C}},\ \Delta\varepsilon_{\mathrm{u,C}}$	rad	Concentrator tilts around the east, north and up axis.
Actuators $i \in \{1, 2\}$	a_i	rad	Actuator radius offset.
	b_i	m	Initial stroke length.
	c_i	m	Distance from actuator origin to joint.
	d_i	m	Actuator radius.
	e_i *	rad	Variable radius shift.
	s_i *	-	Actuator motor position.
	x_i *	m	Variable stroke length.
	λ_i **	m	Actuator gear ratio.

** Dependent parameters computed from fundamental geometric parameters.*
*** The gear ratio is a fixed parameter and not optimized in the calibration.*