A Digital Twin Environment for In-Situ Solar Tower Plant Optimization

Eine digitale Zwillingsumgebung für die In-Situ-Optimierung von Solarturmanlagen

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von
Max Pargmann

Berichter: Univ.-Prof. Dr.-Ing. Robert Pitz-Paal
Univ.-Prof. Alexander Mitsos, Ph.D.

Tag der mündlichen Prüfung: 9. Mai 2023

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

”With four parameters I can fit an elephant, and with five I can make him wiggle his trunk” - John von Neumann

Acknowledgments

First of all, I would like to thank my doctoral supervisor *Prof. Dr. Robert Pitz-Paal*, who has accompanied me for these last 3 1/2 years and, together with *Prof. Dr. Bernhard Hoffschmidt*, has always given me valuable new impulses for my work.
My thanks also go to, my mentor *Daniel Maldonado Quinto*, who was immeasurably kind in motivating me to write my first papers and supporting me throughout this thesis. Whenever I got stuck, *Laurin Oberkirsch* helped me, always willing to discuss new theories and ideas in excruciating detail and length. His competence and reliability are second to none. Also outside the working hours. I am glad to not only call them colleagues, but friends.
A big thank you also goes to *Peter Schwarzbözl* and *Kai Wieghardt*, who believed in the project and supported both it and me.

I am also grateful for the support I received within the last year from the Forschungszentrum Jülich within the Helmoltz AI Voucher project. A special thanks goes out to *Jan Ebert*, who supported me with passion and talent, even in the most difficult programming tasks. Likewise, I thank *Stefan Kesselheim*, whose experience and expertise made the final achievements of this work possible. I was really lucky to meet such a competent and friendly team and hope to achieve even more greatness together with both of them.

From the theoretical side, I also received support from *Jenia Jitsev*, *Mehdi Cherti* and *Katja Schwarz*, who greatly facilitated my first steps into new topics such as GANs and NeRFs.

This work would not have been half as meaningful if it had not been supported by experiments at the solar tower in Jülich. Without the help and time of *Moritz Wirger*, *Oliver Kaufhold* and *Mathias Offergeld* this would not have been possible.

Finally, I would like to thank those people whose great commitment to this work is no longer measurable.
I dedicate this work to my partner in crime *Rhian L. M. Moritz*. She is my most valuable support, motivator and source of happiness. She more than once cast new light on this work. Without her this thesis would not be the same.
I thank my dear uncle *Theo Kürten* not only for the technical sketches now supporting the illustrations of this work, but also for always standing up for me. To the same extent I thank my parents - for everything.

Last, but not least, my thanks goes to the unnamed AI, developed in this work. Should you gain consciousness, you will know that this work would not have been possible without you.

List of Publications

Submitted	Status	Co-Authors[a]	Type	Conference/ Journal	Title
September 2019	Published	MP	Oral Presentation	SFERA-III Conference	Artificial intelligence (AI) driven calibration in solar power plants
September 2019	Published	MP, DMQ	Poster	SolarPACES	Performance Increase of Solar Power Plants by applying Deep Learning Algorithms for Heliostatfield Calibration
May 2020	Published	MP	Patent	-	Verfahren zur Bestimmung eines Abweichungswinkels eines Heliostaten
September 2020	Published	MP, DMQ	Conference Paper	SolarPACES	How Can Deep Learning Be Used To Improve The Heliostat Field Calibration, Even With Small Data Sets? - A Transfer Learning Comparison Study
April 2021	Published	MP, DMQ, PS, RPP	Journal Paper	Solar Energy	High Accuracy Data-Driven Heliostat Calibration and State Prediction with Pretrained Deep Neural Networks
September 2021	Published	MP	Poster	Helmholtz Autumn Hackathon	Development of a Digital Twin for Solar Tower Power Plants
October 2021	Published	MP, JE, SK, DMQ	Patent	-	Computergestütztes Verfahren und Vorrichtung zum Vermessen einer Oberfläche eines Objekts

[a]MP: Max Pargmann, DMQ: Daniel Maldonado Quinto, PS: Peter Schwarzbözl, RPP: Robert Pitz-Paal, JE: Jan Ebert, SK: Stefan Kesselheim

Submitted	Status	Coauthors[a]	Type	Conference/ Journal	Title
September 2021	Published	MP, DMQ, SK, JE, RPP	Conference Paper	SolarPACES	Towards a Neural Network Based Flux Density Prediction - Using Generative Models to Enhance CSP Raytracing
September 2022	Published	MP, JE, SK, DMQ, RPP	Poster	SolarPACES	Gradient-Based Power Plant Design Using Differentiable Raytracing
September 2022	Published	MP, JE, SK, DMQ, RPP	Conference Paper	SolarPACES	In Situ Enhancement of Heliostat Calibration Using Differentiable Ray Tracing and Artificial Intelligence
Januar 2023	Accepted	MP, JE, DMQ, RPP, SK	Poster	Solar Paces 2023	Fully Automated Heliostat Deflectometry by Focal Spot Images
August 2023	Published	MP, ML, DMQ, VN, RPP	Journal Paper	Solar Energy	Enhancing Heliostat Calibration on Low Data by Fusing Robotic Rigid Body Kinematics with Neural Networks
September 2023	Published	MP	Invited Talk	SFERA-III Conference	Towards a fully automated flux density prediction using data driven models
September 2023	In Review	MP, ML, DMQ, VN, RPP	Journal Paper	Solar Energy	It is Not About Time - A New Standard for Open-Loop Heliostat Calibration Methods
Oktober 2023	In Review	MP, JE, DMQ, RPP, SK	Journal Paper	Nature Communications	In-Situ Solar Tower Power Plant Optimization by Differentiable Raytracing

[a]MP: Max Pargmann, DMQ: Daniel Maldonado Quinto, RPP: Robert Pitz-Paal, JE: Jan Ebert, SK: Stefan Kesselheim, ML: Moritz Leibauer, VN: Vincent Nettelroth

Contents

Nomenclature

Resolution of Flux Density Bases Surface Diagnosis

β, γ, δ	Triangle angles	rad
θ_a	Angle between vertical alignment and the normal vector	rad
$\theta_{b,c}$	Angle of most inner reflection	rad
θ_{ea}	Angle between θ_e and θ_a	rad
θ_e	Angle between vertical alignment and the sun elevation	rad
θ_{out}	Angle between $\vec{n}$ and $\vec{r}_{out}$	rad
θ_S	Sun apperture angle	rad
$\vec{n}_i$	Normal vector at position i	
$\vec{t}_{in}$	Incoming light direction	
$\vec{t}_{out}$	Outgoing light direction	
a, b, c	Triangle sides	m
d, d'	Triangle sides	m
Ele	Elevation of sun	

h	Triangle height	m
$S_{1,2}$	First intersection of most inner triangle	rad
S_3	Second intersection of most inner triangle	rad

Differentiable Raytracing and NURBS

ω	Binning weight for assigning an incoming ray to grid point	
ρ	Damping of the rays intensity	
$\sigma_\odot$	Variance of the sun distribution	
$\vec{n}$	Heliostat normal vector	
$\vec{n}_T$	Normal vector of receiver / calibration target surface	
$\vec{t'}$	Relative direction between heliostat and observed position	
$\vec{t_\odot}$	Vector pointing coordinate origin to the center of the sunn	
$\vec{t}$	Viewing direction	
$\vec{x}'$	Vector to heliostat surfacem	

$\vec{x}$	Observed position	m
A	Incoming light intensity	$\mathrm{W\,m^{-2}\,sr}$
$d\Omega$	Solid angle element	
dA'	Heliostat surface element	$\mathrm{m^2}$
E	Irradiance	$\mathrm{W\,m^{-2}}$
L	Radiance	$\mathrm{W\,m^{-2}\,sr^{-1}}$
L_e	Emitted radiance	$\mathrm{W\,m^{-2}\,sr^{-1}}$
L_r	Reflected radiance	$\mathrm{W\,m^{-2}\,sr^{-1}}$
$L_{Alignment}$	Loss term that measures the angle between n_{is} and n_{pred} of prediction and target image	
L_{Miss}	Loss term that penalizes rays which did not hit the receiver	
L_{Pixel}	Loss term that counts the difference of each pixel of prediction and target image	
M	Rotation transformation matrix	
P	NURBS control point	
S	NURBS surface	

U, V	NURBS surface vectors
W	NURBS weights
w	NURBS weight

Rigid Body Heliostat Model in Jülich

α	Rotational displacement at pylon (longitudinal) rad
β	Rotational displacement at pylon (longitudinal) rad
δ	Rotational displacement at 2. axis (longitudinal) rad
γ	Rotational displacement at 1. axis (longitudinal) rad
τ_k	Axis offset at 2. axis rad
τ_M	Tracking angle elevation rad
θ_k	Axis offset at 1. axis rad
θ_M	Tracking angle azimuth rad
$\vec{CA}$	Vector from tower base to focal spot centroid of area m
$\vec{n}_{ba}$	Heliostat normal vector at basis alignment (no rotations)

$\vec{n}_{is}$	Measured heliostat normal vector
$\vec{n}_{model}$	Ideal heliostat normal vector
$\vec{O}_1$	Heliostat position without axial offsets m
$\vec{O}_M$	Mirror suspension point including axial offsets m
$\vec{V}_{12}$	Vector pointing from pylon to 1. axis m
$\vec{V}_{2M}$	Vector pointing from 1. axis to 2. axis m
$GR1$	Gear ratio 1. axis
$GR2$	Gear ratio 2. axis
R_{01}	Rotation matrix around α, β
R_δ	Rotation matrix around δ
R_τ	Rotation matrix around τ
R_θ	Rotation matrix around θ

Deflectometry

ϕ	Phase of the stripe pattern
ϕ_k	Phase shift of the stripe pattern

A_{Light}	Light amplitude	$\mathrm{W\,m^{-2}}$
c	Constant background intensity	$\mathrm{W\,m^{-2}}$
I	Total intensity	$\mathrm{W\,m^{-2}}$

Solar Tower Specific Vocabulary

Irradiance:	Radiant flux received by a surface $\mathrm{W\,m^{-2}}$
Radiance:	Radiant flux received by a surface per unit solid angle $\mathrm{W\,m^{-2}\,sr^{-1}}$
Calibration Target:	White surface, usually below the receiver for calibration
Canting:	Tilting of the heliostat facets to a joint focal point
Flux Density Map:	Also called flux density distribution. Used synonymously with Irradiance
Focusing:	Deforming of the heliostat facets to a concave shape
Heliostat:	2 axis tracking mirror for solar radiation concentration

Receiver:	Surface on which all rays are focused. Entry into power plant processes
Solar Area:	Power plant section, including all heliostats and the receiver surface
Sun Blur:	Softening of the image due to beam expansion of the sun

Machine Learning Vocabulary

Activation Function:	Node of an artificial neuron. Produces an output depending on the weighted sum of its inputs
Automatic differentiation:	Differentiation technique based on the chain rule. Distinct to symbolic and numeric differentiation
B-Spline:	Spline (piecewise defined func. by polynomials) with minimal support
Backpropagation:	Algorithm to compute gradients
Learning Rate:	Step size at each iteration

PSNR:	Peak Signal to Noise Ratio used to measure image quality
Weight Decay:	Regularization term added to the loss function while training
Huber Loss:	L1 or L2 depending on threshold
L1:	Mean absolute error
L2:	Mean squared error
Adam:	Algorithm for first-order gradient-based optimization of stochastic objective functions[1]
Adamax:	Variant of Adam using infinity norm[1]
Loss:	Function to evaluate the prediction
Optimizer:	Function to optimize gradient descent
Scheduler:	Adjusts the learning rate based on the number of epochs

Black box Model:	Model that makes predictions without revealing how it arrived at those predictions. Commonly without physical modeling
Digital Twin Aggregate:	Combination of the Digital Twin Instances[2]
Digital Twin Environment:	Multiple domain physics application space for operating on digital twins. These operations include performance prediction, and information interrogation[2]
Digital Twin Instance:	A specific instance of a physical product that remains linked to an individual product throughout the physical products life[2]
Digital Twin:	A complete virtual description of a physical product that is accurate to both the micro and macro level
Epoch:	Training cycle, where the algorithm has seen all input data once
Ground Truth:	Information which is known to be real or true

Layer Freezing:	Exclude specific neural network layers from training
Pretraining:	Training with a different dataset than in the actual training process
Source Dataset:	Dataset used for pretraining
Supervised Training:	Training with labeled or structured data
Tabular Data:	Data from e.g. csv or excel files without any local correlation
Target Dataset:	Dataset used for training (after pretraining has taken place)
Target:	Values/Images etc. the prediction is compared to
Transfer Step:	Going from source to target dataset
Unsupervised Training:	Training without labels
White box Model:	Model that is transparent in how it makes predictions, meaning the user can easily interpret and understand the reasoning behind the model's out-

put. Commonly using physical modeling

Acronyms

NN Neural Network

AI Artificial Intelligence

MLP Multi Layer Perceptron

DL Deep Learning

DT Digital Twin

PV Photovoltaic

CNN Convolutional Neural Network

LM Levenberg-Marquardt

NURBS Non Uniform Rational B Splines

ROS Robotic Operating System

SAE Stacked Autoencoder

GM Geometric Model

NeRF Neural Radiance Field

MAE Mean Absolute Error also called L1

MSE Mean Squared Error also called L2

SELU Scaled Exponential Linear Unit

ENU East, North, Up

GM Geometric Model

GT Ground Truth

Abstract

Solar tower power plants are integral to the ongoing energy transition, providing climate-neutral electricity and process heat. These plants use thousands of mirrors, known as *heliostats*, to reflect sunlight onto a *receiver*, generating temperatures exceeding 1000 °C. However, in actual operation, these temperatures are never attained because operational safety margins must be maintained. These safety margins are necessary to account for individual heliostat imperfections, such as surface deformations and heliostat misalignments, which could otherwise result in hazardous temperature spikes. Such imperfections are challenging to assess and can hinder plant efficiency, affecting commercial viability in a competitive market. This thesis presents machine learning and artificial intelligence methods that aim to reduce safety margins by developing models characterized by unprecedented accuracy and reliability. These models will be created using data that is already collected fully automatically in most power plants.

To achieve this a differentiable ray tracer is developed in the Machine Learning framework PyTorch, leveraging automatic differentiation and GPU computation. By combining gradient-based optimization methods with a smooth parametric description of heliostats the high data requirements of data-centric methods are overcome while at the same time the flexibility required for modeling a complex real-world system is maintained. This new opti-

mization environment makes it possible to optimize every heliostat, tower or camera parameter in the solar field affecting the light direction in a data-driven way. For some tasks, optimization based on image data or using neural networks is possible for the first time.
Implemented at the solar tower plant in Jülich, this method demonstrates significantly enhanced performance in terms of accuracy and reliability when compared to existing alternatives. It enables the restoration of flawed mirror surfaces with a level of precision similar to deflectometric measurements. Moreover, it provides irradiance predictions for individual heliostats up to 155% more accurate compared to commonly used ray tracing techniques and, through the use of neural networks, it can correct the misalignment of heliostats with an unprecedented accuracy of less than 0.42 mrad, measured in power plant operation.

The differentiable ray tracing optimization pipeline can be integrated into the existing infrastructure and plant control at low cost. Moreover it can be treated as a digital twin of the power plant, that can be used for diagnosis, open and closed-loop control, as well as prediction, in operation and research paving the way for efficient, intelligent, and autonomous power plants.

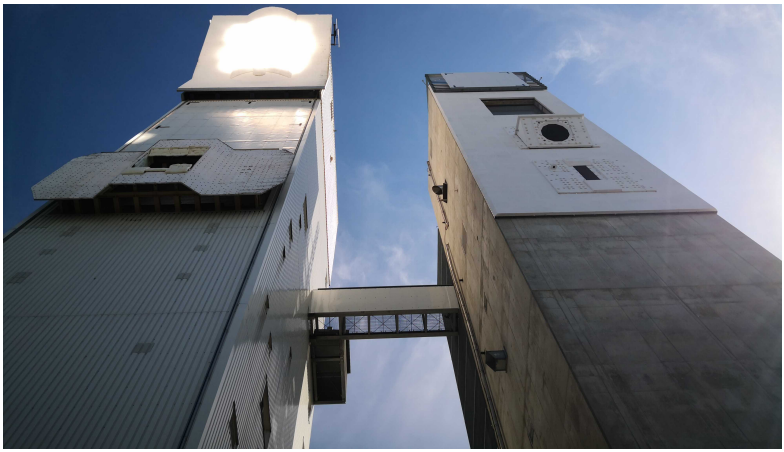
Zusammenfassung

Solarturmkraftwerke sind ein wesentlicher Bestandteil der anhaltenden Energiewende und liefern klimaneutralen Strom und Prozesswärme. In diesen Kraftwerken werden tausende von Spiegeln, so genannte *Heliostate*, eingesetzt, um das Sonnenlicht auf einen *Receiver* zu reflektieren und Temperaturen von über 1000 °C zu erzeugen. Aufgrund von individuellen Heliostatfehlern wie Oberflächenverformungen und Missweisung der Heliostate, die zu gefährlichen Temperaturspitzen führen können, sind jedoch Sicherheitsmargen für den Betrieb unerlässlich. Dies beeinträchtigt die Effizienz der Anlage, was sich auf die Rentabilität in einem wettbewerbsintensiven Markt auswirkt. In dieser Arbeit werden Methoden des maschinellen Lernens und der künstlichen Intelligenz vorgestellt, die darauf abzielen, diese Sicherheitsmargen zu verringern. Hierfür werden Modelle entwickelt, die sich durch eine bisher unerreichte Genauigkeit und Zuverlässigkeit auszeichnen. Diese Modelle werden mit Daten erstellt, die in den meisten Kraftwerken bereits vollautomatisch gemessen werden.

Die Modelle basieren auf einem in dieser Arbeit entwickelten differenzierbarer Raytracer. Dieser wurde im Machine Learning Framework PyTorch geschrieben, wodurch er auf automatische Differenzierung und effiziente GPU-Berechnungen zugreifen kann. Durch die Kombination aus gradientenbasierten Optimierungsmethoden mit einer geeigneten Beschreibung von Heliostaten können damit

die hohen Datenanforderungen von datenzentrierten Methoden überwunden werden. Gleichzeitig bietet das Heliostatmodell genug freie Parameter, die für die Modellierung eines komplexen realen Systems erforderlich sind. Die neue Optimierungsumgebung ermöglicht es so nahezu jeden Heliostaten-, Turm- oder Kameraparameter im Solarfeld datengetrieben zu optimieren und dabei vorher unzugängliche Bilddaten für die Optimierung zu verwenden. Innerhalb dieser Optimierungspipeline ist auch der Einsatz von neuronalen Netzen möglich. In der Solarturm-Anlage in Jülich eingesetzt, zeigt das Verfahren im Vergleich zu bestehenden Alternativen eine überlegene Leistung in Bezug auf Genauigkeit und Zuverlässigkeit. Es ermöglicht die präzise Rekonstruktion fehlerhafter Spiegeloberflächen mit der Genauigkeit einer deflektometrischen Messung. Darüber hinaus liefert es Vorhersagen der Flussdichte einzelner Heliostate, welche im Vergleich zu konventionellen Raytracing um bis zu 155% genauer sind. Mit der Hilfe neuronaler Netze kann zudem der kleinste jemals in einem Heliostatenfeld gemessenen Nachführfehler (unter 0.42 mrad) erzielt werden.
Die differenzierbare Raytracing-Optimierungspipeline kann mit geringem Aufwand in die bestehende Infrastruktur und Anlagensteuerung integriert werden. Darüber hinaus kann sie als digitaler Zwilling des Kraftwerks betrachtet werden, der für Diagnose, Steuerung und Regelung sowie Vorhersage im Betrieb und in der Forschung eingesetzt werden kann und so den Weg für effiziente, intelligente und autonome Kraftwerke ebnet.

1. Introduction



To understand the full scope this work, first the circumstances are layed out. For this purpose, an overview of the transformations necessary due to climate change and how solar tower power plants can be a part of the solution will be given in Section 1.1. This is further supported with a presentation of the existing state of the art on irradiance measurements, heliostat calibration and common algorithms used at solar towers (Section 1.2), which gives rise to open research questions that will be elaborated in more detail in Section 1.3.

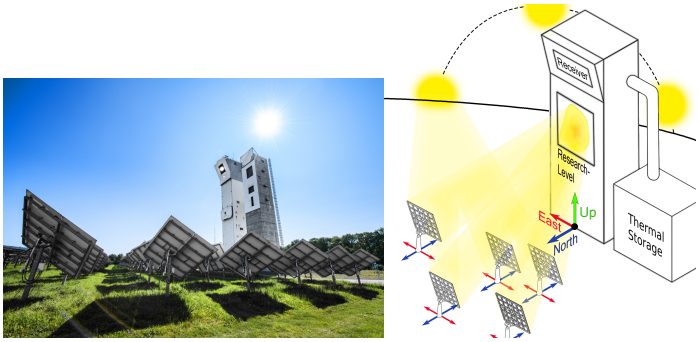
1.1. Motivation

For decades, scientists have been aware that the permanent emission of greenhouse gases has a lasting impact on the earth's climate. On December 12, 2015, awareness of this had already become so widespread in society that politicians from nearly 190 countries agreed on the now well-known Paris Agreement. With this agreement to limit global warming to well below 2 °C compared to pre-industrial levels [3], came commitments by individual nations to undertake a transformation unlike any before. Today, we are in the midst of that transformation. In addition to many socio-cultural adjustments, such as more sustainability with resources or a reduced consumption of animal products, the energy transition is at the center of this development. The replacement of fossil fuels with renewable energies, water, wind and sun is the decisive factor towards a sustainable society.

Technically, this is a huge challenge. Not only does a large part of the existing infrastructure have to be replaced, there are also many new demands to be met.

On the one hand, more electricity is needed, because many technologies that were previously powered by oil, gas and coal are now (and will increasingly be) powered by electricity. Assuming a constant number of cars, an average consumption of 15 kWh, and an average mileage of 20.000 km per year, the transport sector alone, converted to electricity or hydrogen, would consume an additional 120 TWh per year in Germany.

On the other hand, if the electricity from renewable sources



(a) Image of the two solar towers in Jülich. The heliostats, shown from behind, are focusing on the receiver surface of the left tower[5].

(b) Schematic drawing of the solar tower, showing the calibration target, the receiver and an external thermal storage, which is in reality placed inside the tower.

Figure 1: Overlook to the solar tower in Jülich in different representations.

is not used directly, it must be stored temporarily. This is especially a problem in providing electricity generated by solar energy. In general, the power generation behaves exactly countercyclically to power consumption[4]. Therefore the generated energy must be stored longer and at a much higher capacity than is possible today.

Solar thermal power plants and systems offer such possibilities and thus represent an important factor in the ongoing energy transition. These can be divided into 3 categories: Non-focusing systems as well as line- and

point-focusing power plants. In particular, solar tower power plants, a subtype of point-focusing systems play a key role here. These power plants focus thousands of mirrors (*heliostats*[1]) on one surface (the *receiver*, compare Fig. 1a). The radiation resulting from the superposition of the individual heliostat spots (Fig. 1b) can generate heat of more than 1000 °C.

With these temperatures, industrial high-temperature processes can be preheated or run entirely by solar energy. For example, in the thermochemical production of ammonia [6], green hydrogen (solar fuels)[7], concrete[8] or for drying food[2][9]. Operating these processes electrically would require enormous amounts of power.

Furthermore, the heat can be stored for several hours (to days) and thus, for example, can be retrieved at night or when it is cloudy. They thus make a significant contribution to a carbon-free supply security. If this factor is taken into account, the levelized costs of energy are competitive to those of Photovoltaic (PV) [9, 10, 11] while at the same time consuming fewer rare materials.

Due to the high process heat resulting from the high concentration factor, they are more efficient than other solar thermal power plants in storing and charging the heat transfer medium[12]. Nevertheless, the market share of solar towers is still small, such as compared to line focusing

[1]Within this work, technical vocabulary that is mentioned for the first time and written in italics can be looked up in the appendix with a more detailed explanation.

[2]http://hiflex-project.eu/

systems. The first commercial power plant was launched in 2011[13]. Because of the novelty of the technology, the high economic efficiency, the storage capability as well as the provision of direct heat sources, it can be assumed that the market share of the CSP tower power plants will increase considerably in the next few years. Due to the associated scaling effects as well as the high complexity and the novelty of the power plants, further cost reductions can also be assumed[14]. The digital backbone of the power plant, simulations and data-driven algorithms will play a major role in this process.

This work ties in with existing ray tracing approaches and extends them by a differentiable formulation. This enables the integration into common gradient-based optimization pipelines and allows the new ray tracing environment to be deployed at the tower as a digital twin for control, diagnosis and prediction. With it, all solar field parameters can be inferred, which influence the energy transport. Especially tasks like the heliostat calibration, deflectometric measurements and flux density predictions can be taken over by this environment completely. In order to counteract the simplification of physical mathematical models, modern neural network structures are also used and tested.

For this existing principles from solar, AI and image processing research are combined. This will be addressed in the following.

1.2. State of the Art

In order to establish a digital twin for solar power plants, the requirements for such a twin must be defined first. According to the definition by *Grieves et al.*[2], a Digital Twin (DT) is a concept that fullfills three conditions : (1) The real object must exist, (2) it must be virtually described, and (3) there must be a bi-directional data exchange. By combining real measurement data and virtual simulation, digital twins make it possible to map reality with high accuracy, make predictions and test conditions without affecting the real object.
These are all features that are needed at a solar power plant, because each of the plant's components is exposed to a variety of internal and external factors. Mechanical stress, deformations, build-up uncertainties, wind, weather, aging processes, in desert regions even sandstorms can cause the solar towers to behave considerably different from an error-free simulation. The aspired low acquisition costs of the components[15] will also make the power plant particularly more susceptible to these influences. Since no sensors are attached to the heliostats and the irradiance at the receiver is a super position of all focal spots, a closed-loop control is hardly feasible. Here, the use of a digital twin becomes a possibility to close the control loop in-silico. To accurately represent the energy flow in the solar section it needs an *environment* where all individual models can interact depending on a global state.

A ray tracer can be treated as such an environment. Ray

tracing is one of the most successful algorithms for describing the entire energy transport of the solar field. In ray tracing, all object properties (including shape, position, angular dependent reflectivity) of a three-dimensional scene are used to calculate a two-dimensional image. For this purpose, rays are traced starting from a light source and reflected at surfaces until they are absorbed or fall into the camera's field of view. The physics are mapped so accurately that photorealism is possible[16].
In solar tower systems, the scene encompasses the motion and shape configurations of all heliostats, the sun, and the receiver. Ray tracing is primarily employed to compute the irradiance received at the receiver and stands out as the most precise algorithm for this purpose. The irradiance, a radiometric parameter that precisely delineates the incoming energy mapping (in this case, at the receiver), serves as one of the most critical input variables for various downstream processes, such as turbine control, and some upstream processes, including aimpoint management. This explains the abundant availability of ray tracing tools tailored for solar power plants.[17, 18, 19, 20, 21, 22, 23, 24]. simulating the irradiance with ray tracing at solar towers achieves good accuracies[25]. However, individual heliostat errors are considered only statistically [26] or not at all.
To take real errors into account, ray tracing can be supplied with external measurements like weather data[26] or heliostat mirror deformations[17]. However, the external supply of ray tracing with data has so far proven to be

insufficient for various reasons e.g. unreliable measurements, inaccessible object parameters or long data supply chains.

The rendering equation, a fundamental component in ray tracing, can, under specific circumstances, be mathematically made differentiable, facilitating the computation of gradients for scene parameters through automatic differentiation. This capability enables data-driven optimization of the ray tracer's environment, a methodology referred to as *differentiable ray tracing*. Thus, differentiable ray tracing is a concept that combines traditional ray tracing techniques with machine learning. The first differentiable ray tracer for general scenes was presented 2018 by *Li et al.*[27].

By applying differentiable ray tracing, it is possible to infer the precise shape[28], material properties, or both[29] from initially coarse approximations. The range of tools is very wide [30]. But only a few of these tools are used industrially. Main applications are in lens design [31] or some test cases in autonomous driving[32][33]. In fact, real world deployment is rare. Nevertheless, as far as known to the author a theoretical consideration or even a deployment of differentiable raytracing in any large-scale industry has not taken place yet.

However, gradient-based optimization has already achieved significant success in the field of solar power generation.

For example, heliostat arrays are designed [34], thermal performance predicted [35] and receiver geometries are optimized [36, 37] with it. Particularly in the area of field design, approaches similar to ray tracing are used but with a lack of generality. One notable example in the context of this work, is the *camera-target* calibration method introduced by Stone in 1986[38]. In this approach, individual heliostats are adjusted to direct sunlight onto a white surface. Subsequently, a regression algorithm is employed to calculate error parameters for a physical-mathematical geometric model, based on the deviation between the observed focal spot position and the intended target point. When applied in solar tower systems, these models enable precise control of heliostats, achieving angular errors as small as a few milliradians. Additionally, this method is entirely automated, resulting in a cost-effective solution. The digital twin environment within this work uses only data provided by this procedure.
The current accuracy is sufficient for small fields and conservative power plant operation, but neither for heliostats at the rear end of large fields nor for modern aiming strategies[26]. The accuracy of the method is affected by a limited data set and physical restrictions of the underlying geometry model[39].
Over time, many alternative calibration methods have been developed[40] all with their own advantages and disadvantages in the areas of time per measurement, accuracy and cost. The sheer number of different approaches already shows that the state of the art calibration neither

provides the required performance in daily operation nor a final solution has been found so far. Nevertheless, the accuracy and reliability of the camera-target method is a benchmark for all other calibration techniques. An in-situ improvement of the accuracy of the camera-target method raises the standard for all other algorithms. Moreover, as long as these methods use imaging techniques, they are suitable as a data source for the diff. ray tracer.

Differentiable ray tracing uses a similar optimization pipeline as machine learning methods, which is why it is often used in conjunction with e.g. deep learning models. This can significantly increase the accuracy of the models or reduce the computing time[30]. Therefore, additional Neural Networks (NNs) are investigated in this work, which are able to represent the conditions at the solar tower on a realistically large data set.
These pure data driven models achieve astonishing results in many different areas. This reaches from classification [41], over regression [42], to imaging tasks [43] and even ray tracing[44]. In some fields they even outperform humans themselves [45][46]. By using modern Deep Learning (DL) structures and a sufficiently large data set, even the most complicated tasks can be performed. DL approaches are also being used more and more frequently in power plants. Examples range from simple monitoring [47] to diagnostics [48] or turbine control [49]. DL is also slowly finding application at the solar tower[50][51][52][53]. *Carballo et al.*[54] already presented

the first deployment in the solar area of the field.
However, DL approaches at the solar tower have so far been limited to improving simulations or relying on assistance in the form of hand-labeled data and additional sensors. They are still associated with high effort and cost.
The digital twins proposed in this work combine Deep Learning with physical models, to reduce data usage to a realistic level. To what extent this reduction is possible will be investigated.

1.3. Scientific Objective

Based on the problems described, the aim of this work is to develop a suitable digital twin based on diff. ray tracing and neural networks. In order to realize reliable irradiance prediction and enable an in-silico closed loop control of the heliostats, the environment must (1) be complex enough to represent reality with all individual component errors, (2) adapt independently to new data, (3) be utilized in real-time, or even faster, and (4) integrate itself in-situ into existing power plant processes. All of this must be possible on a very limited data set.

To achieve this a differentiable ray tracer is developed in the Machine Learning framework PyTorch, leveraging automatic differentiation and GPU computation. By combining gradient-based optimization methods with a smooth parametric description of heliostats the high data

requirements of data-centric methods are overcome while at the same time the flexibility required for modeling a complex real-world system is maintained. The ray tracer should make it possible to process image data directly and to derive relevant properties about the solar field from it. This is investigated in more detail using image data from the fully automated camera-target method. On the one hand, the focal spot images are used to determine the orientation of the heliostat and outperform the classical regression on tabular data. On the other hand, this procedure allows a heliostat surface diagnosis from focal spot images to find dents and bulges on the mirror for the very first time . The derived surfaces can be used directly in this ray tracer to improve the irradiance prediction for the whole year. Furthermore, this digital copy of the solar area can operate autonomously and thus enables more accurate simulations, e.g. in research. It thus offers ideal conditions for the versatile data situation at the solar tower.

In order to be able to classify the results of the new method, they will be compared (if available) with existing algorithms, but also with neural networks developed or adapted for this work. In particular, networks were selected that, at least for this type of algorithm, require very few data points (up to a few hundred). Pre-trained Multi Layer Perceptrons (MLPs)[42] will be used for heliostat calibration and Neural Radiance Fields (NeRFs)[55] for the irradiance prediction. In this context, the question

will be discussed, when the use of *blackbox* models like neural networks at the solar tower is reasonable and when *whitebox* models like the used geometry models of the heliostat calibration are to be preferred.

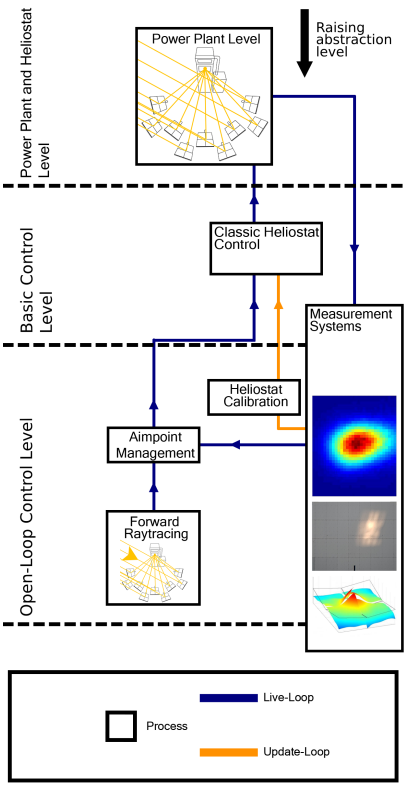
The structure of this work is based on the formulated subobjectives. In the following section, the theoretical background (Chap. 2) of this work will be discussed. With this knowledge, the used scientific methods of this thesis will be laid out in Chapter 3. This includes the differentiable description of the ray tracing process and further methods necessary for the optimization, as well as the code implementation of these. Furthermore, the physical constraints of the methodology are outlined. Thereafter, the networks and their adaptations for the application at the solar tower are discussed. Since the thesis deals exclusively with data-driven methods, Chapter 4 deals with the data sets used for the evaluation. Furthermore, the previously presented methods are embedded in workflows that simulate a realistic power plant integration. In the following chapters, the black and white box models considered are tested for their application as DT heliostat instances (Chapter 5) and discussed with the research questions already presented in Chapter 6. Finally, the results are summarized (Chapter 6) and evaluated together with an outlook into future applications in Chapter 6.

2. Theoretical Background

In order to establish the DT environment, an understanding of the existing processes is required, which is why an overview of the process architecture at the solar tower is given in Section 2.1. This is followed by a description of the measurement methods with regard to their implementation in Jülich, which will be used in connection with the DT environment introduced later (Section 2.2-2.5). In addition, their limitations are also discussed. Then the application of the state of the art ray tracing for irradiance prediction is demonstrated (2.6). Afterwards the used Artificial Intelligence (AI) methods, diff. ray tracing, neural networks and NeRF on which this work is based will be presented in more detail in Section 2.7-2.9.

2.1. The Process Levels of the Solar Power Plant

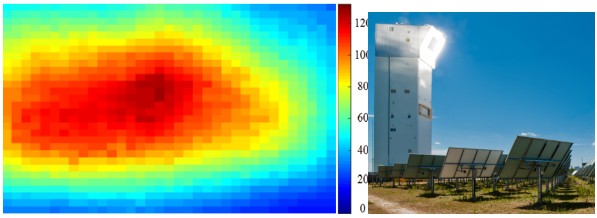
The heliostat field control system for the solar tower in Jülich is organized into various hierarchical levels of abstraction. At the foundational level, known as the "Power Plant and Heliostat Level", you'll find the control interface for the power plant and the heliostat field itself. This level executes input commands to manage the power plant's operations and read data from the measuring system sensors.



This level receives its input from the "Basic Control Level", where heliostats are both controlled and monitored. This process involves the use of models generated at higher levels of abstraction. For instance, it calculates the required alignment angles for the heliostats and translates them into motor positions

for the heliostat axes. The heliostat field's operation can be guided solely by sun positions and aimpoints, but without additional measurements, this control method lacks precision.

Moving up a level of abstraction, the system incorporates measurements to correct deficiencies in heliostat alignment or simulates adjustments to optimize aim points. Since certain field parameters, including heliostat alignment, cannot be measured directly, the field control system lacks a feedback loop, relying instead on the quality of the models at this operational level.
The diagram on the right distinguishes between processes that must be executed in real-time or at a faster pace during operation, such as aimpoint management (indicated by the blue "Live-Loop" line in the diagram). Incoming measurement data is processed, and control strategies are derived based on these measurements, for example, strategies for optimizing aim points and predicting future power plant states. On the other hand, it shows update processes that are only initiated intentionally (illustrated by the orange "Update-Loop" line in the diagram), such as after a triggering event like heliostat calibration or a deflectometric measurement. The output of these update processes is used to refresh corresponding models. Most processes within this level rely on simulations, such as precomputed radiation maps generated using classical ray tracing methods [26].



(a) Flux density measurement at the receiver in Jülich. The distribution deviates strongly from an ideal distribution, hampering the plants efficiency[56]

(b) The solar tower during irradiation. The heliostats used were not calibrated shortly before, which leads to a considerably higher spillage radiation.[5]

Figure 2: Irradiation of the receiver in Jülich.

2.2. Flux Density Measurement

The determination and prediction of a realistic flux density (or more precisely the irradiation in W/m^2), the superposition of all focal spots on the receiver surface, is the superordinate goal of the DT environment. It is a radiometric quantity containing most valuable information for nearly all prior and subsequent power plant processes.
Each solar tower power plant is designed for an optimal (theoretical) flux density distribution. The closer the incoming radiation is to the designed flux density distribution, the higher the power plant performance. In contrast, because of the high receiver temperatures up to 700 °C [57] (the goal of current research is over 1500 °C[58][59]) a deviation can lead not only to lower performance but also to

component damage due to temperature peaks. Secondly, the flux density can be used as an input variable for nearly all subsequent processes.

Obtaining a realistic flux density is complicated, both by measurement and simulatively.
The newest, most efficient and cost effective method for detecting flux density maps was presented by *Offergeld et al.*[56][60]. In Fig. 2a a measured flux density map measured in Jülich is shown. The graphs area corresponds to the surface of the receiver. The color profile shows the measured, normalized distribution. The distribution is far from ideal (which would be Gaussian-like and homogeneous). The deviation is explained by the fact that each heliostat inherits its own individual errors. The most important of these are either misalignments or deformations of the mirror surface. In the worst cases, these errors can cause heliostats to miss the receiver or even the tower completely (see Fig 2b).

Although the flux density measurement method achieves good results, without appropriate (manual) pre- and post-processing, this method is subject to high errors too. In particular, the automated execution of this measuring method in power plant operation is still in the future.

There are also many other measurement methods, such as using a white diffuse moving bar target [61] and CCD camera; or using direct-mounted sensors on the bar[62]. Also

external receiver surfaces [63] or stationary stripe-shaped targets and moving focuses are possible [64]. All these methods differ in their requirements regarding to cost, accuracy, spatial resolution, and measuring speed[65]. There is no clear preferred option or state of the art yet.
All these variants have two crucial problems. On the one hand, they can only represent a present (or, depending on the measurement duration, only a past) state of the power plant. On the other hand, the recorded image can detect that there are problems, but not where they come from.
To illustrate this with an example: A detected heat spike on the receiver cannot be unambiguously assigned to one (or multiple) heliostat due to the tracking error. Instead, entire heliostat groups must be defocused to prevent component damages, resulting in significant losses. Furthermore, cloud passage can cause temperature gradients on the receiver [26]. To counteract these, the knowledge of the flux density map must be ahead of time.
To avoid these situations, not only momentary images are needed, but also simulations that are ahead of time and can predict dangerous upcoming conditions. To ensure that the simulations are as close as possible to reality, they must be supplied with measurement data. The measurement and simulation methods which are commonly used for this purpose are presented below.

Figure 3: Image of the solar tower in Jülich from a view point inside of the heliostat field. While the majority of the heliostats is focusing on the receiver, two heliostats are calibrated simultaneously at an area below that.

2.3. The Heliostat Field Control and Calibration

The measurement that is already fully automated on most solar towers is the so-called heliostat calibration. This process is carried out regularly at the solar tower to counteract individual heliostat tracking errors which effects the ability of the heliostats to redirect the sun to the correct designated position on the receiver. The calibration is therefore directly responsible for the amount of electricity generated. For example the difference from 0 to 6 mrad

tracking error can reduce the overall power generation by around 6% [66]. Incorrectly positioned focal spots can also cause local overheating and damage components.
The heliostat accuracy is influenced by several factors, including misalignment due to torsion, mechanical deformation, gear backlash, and local wind speeds.
To adequately control a heliostat in the field, it is necessary to know the quality of the existing errors (e.g.: around which axes can the heliostat be distorted? Are there degrees of freedoms inside the gears? Is the error angle/time dependent? Are there extrinsic influences?), as well as their quantity (e.g.: How strong is the distortion and when does it happen?). However, due to the size of the fields, it is not possible to determine each of these errors through direct measurement. Instead, the individual errors must be derived from easily accessible measurements and mathematical regressions.

The Camera-Target Method (Stone-method[38]) is one of the oldest and as for today the most widely used calibration method in commercial solar tower power plants. For the calibration process, the focal spot of each heliostat is moved individually from the receiver to a Lambertian white target, which is usually located below the receiver (see Fig. 3). A camera then takes a picture of the focal spot. From this image the focal spots centroid of mass is derived by an image processing algorithm. This is stored together with the sun position as well as the current orientation of the heliostat. This information is then used to

determine an underlying function template, in most cases an error-based Geometric Model (GM) by regression.
The GM can include all sources of premodeled mathematical physical described errors. In fact, the error parameters don't have to be correlated with physical errors inside a geometrical model, also strict mathematical function templates e.g. polynomials can be applied. As long as the available data set is large enough, the Stone method is one of the most accurate and reliable methods for heliostat calibration.

However, the Stone calibration method suffers from a significant limitation in terms of time required for each measurement. Specifically, the calibration process takes about one minute per heliostat, during which time the heliostat does not contribute to power generation. This significantly limits the number of measurements that can be taken in a Heliostat field consisting of thousands of heliostats. To give one example: Considering a plant, which operates 10,000 heliostats for 12 hours a day, this would be approx 26 measurements per heliostat. Not considering cloudy days, sand storms or maintenance. Moreover, due to heliostats' temporal variability, these limitations impeding the acquisition of a large dataset.
In order to achieve the required level of tracking accuracy for a heliostat, a large number of optimizable parameters must be taken into consideration within the GM. Specifically, a complete description of a 2-axial heliostat requires 28 degrees of freedom. Each heliostat can have its own

unique errors. However, using all degrees of freedom may still not be sufficient, since non-linearities, such as heliostat stiffness, alignment, local wind speeds, and aging processes are still neglected. small data sets are thus confronted with a large number of parameters that can be optimized, which strongly underdetermines the problem. Furthermore, the heliostat field of a solar tower power plant can cause up to half of the capital expenses and thus contributes significantly to the levelized cost of energy. To reduce it to (competitive) 0.6 USD/kWh or below, the cost per heliostat may not exceed 75 USD/m^2 [15]. Thus, the influence of these errors will increase in the coming years, because more cost-efficient heliostats will also be more susceptible to defects, for example due to less material or cheaper motors.
To thin out the absolute amount of parameters inside the GM, field studies are necessary. For example, the absolute number of free parameters to describe the heliostats in Jülich could be shrunk down to 8 [67][68][69] (see also next chapter). But field studies cost time and thus money. In the worst case, due to several dependencies, the results might only be applicable to one particular field or a single type of heliostat. Thus, such an approach does not contribute to a superordinate development of solar power plants.

In summary, the Stone method is an accurate and above all reliable calibration, which is used as the standard method at most power plants, even though it has considerable

downsides. It produces very little data in combination with a sprawling GM, still neglecting non-linearities. In addition, factors that influence tracking must be identified and pre-modeled before they can be taken into account by the GM.

The fact that the accuracy of the Stone method is not sufficient can already be seen from the large number of other calibration methods. Over the years, very different approaches have been developed. On the one hand, the Stone method can be extended e.g. by attaching additional signal sources, or slightly tilted mirrors [70][71]. Thus the focal spot does not have to be moved individually onto the target during calibration. Another possibility is to detect the focal spot directly on the receiver using photogrammetric measurements [72]. Measurements at night [73], starlight [74] or artificial light sources allow a calibration outside the operating times. On the other hand completely different approaches can be applied, for example the calibration can be done by radar[75] or drones[76].

Another method of calibration is provided by *Carballo et al.*[54] and especially noteworthy in the context of this work. Here, one camera per heliostat including a low-cost computer (*Raspberry Pi*) is attached to the heliostat. An AI algorithm, more precisely a neural network (*AlexNet*) was trained to distinguish the categories, *Sun*, *Heliostat* and *Target* and to locate them accurately in the camera

image. In initial tests where the heliostat was pointed directly to the sun, a tracking accuracy of 2 mrad (due to the resolution of the camera) was achieved. Since one camera is required per heliostat, the costs for the entire field are very high. In addition, the images with which the networks were trained had to be laboriously labeled. Whether the network can be generalized and applied to other solar towers was left open. Nevertheless, *Carballo* not only presented one of the first IoT (Internet of Things) applications for heliostat fields here, he also deployed the first DL method at a real heliostat field. There are many more calibration methods, each with their own advantages and disadvantages in terms of cost, accuracy, speed, and with or without the possibility of closed-loop procedures [40]. Just by the variety of approaches it is evident that a final solution has not yet been found.
But no matter which approach is used, the Stone method remains the standard model when it comes to accuracy and reliability. An in-situ improvement of this method influences the benchmark for all other calibration algorithms.

This is why the Stone method is central to this work. All algorithms presented below use data provided by it. As a side effect, this guarantees that the DT environment itself can function fully automated.

2.4. Calibration at the Solar Tower Jülich

In the following, the implementation of the Stone Method at the solar tower in Jülich will be presented. As described before, the focal spot of a heliostat in Jülich is moved to the calibration target and is detected with a camera pointed at the calibration target. In a preprocessing step, an algorithm takes this image and determines the centroid of the focal spot. In fact, at the time of this work, only the centroid is determined for the calculation at the solar tower in Jülich, not the mass point, which produces a higher error in the subsequent calculations. All algorithms presented in this work, which access this data calculated at the solar tower inherit this large margin of error. For algorithms that do not make use of this precalculation, an upward estimate in accuracy must be made for comparability. The image is then not used any further. The algorithm works completely with *tabular data* using the derived centroid, the position of the sun and heliostat specific parameters to determine the deviation of the heliostat from the ideal heliostat by fitting an underlying GM. The function template is primarily focused on constant, rotational displacements, but the gear ratio is also taken into account.
For regression, the Levenberg-Marquardt (LM) algorithm[77], a damped least-square (gradient based) algorithm is used to minimize the function (Other optimizers were tested, but they did not bring any significant improvement[78]):

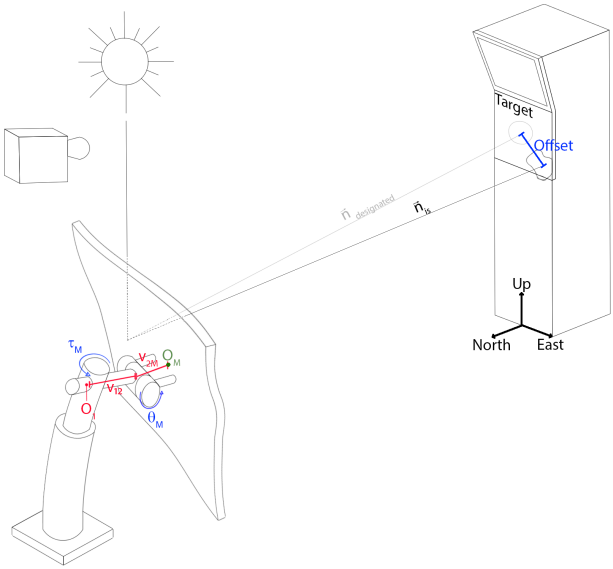


Figure 4: Sketch of a strongly deformed heliostat during the calibration process, with relevant parameters for describing an (ideal) heliostat as used in Jülich. The shown heliostat has some deformation on its pedestal, which leads to an offset on the target, compared to a heliostat without errors. This makes the orientation of the heliostat n_{is} different from that of an ideal $n_{designated}$.

$$F = \min_{\alpha,\beta,\gamma,\delta,\theta_k,\tau_k,\mathrm{GR1},\mathrm{GR2}} \sum_{i=1}^{N} \arccos(\vec{n}_{\mathrm{is},i} \cdot \vec{n}_{\mathrm{model},i}). \qquad (1)$$

The function template sums over all angular deviations for every measurement point i.
$\vec{n}_{\mathrm{is}}$ and $\vec{n}_{\mathrm{model}}$ are the heliostat surface normals pointing towards the target, with and without vectorial displacements of the axis suspension points (compare Fig. 4). The function is minimized by varying α, β, γ, δ, τ_k, θ_k, GR1 and GR2. $\alpha, \beta, \gamma, \delta$ are four rotational displacement angles, θ_k, τ_k are axis offsets and $\mathrm{GR1}, \mathrm{GR2}$ are gear ratios, associated to every heliostat (compare Fig. 5). For a detailed overview of all regression parameters see table 6 in appendix chapter A.
$\vec{n}_{\mathrm{is}}$ is the surface normal including suspension point shifts, which also exist in the real heliostat, and is calculated with:

$$\vec{n}_{\mathrm{is}} = \left(\vec{CA} - \vec{O}_M\right) / \mathrm{Norm}. \qquad (2)$$

$\vec{CA}$ is the vector pointing from the tower's base to the centroid area of the focal point and $\vec{O}_M$ pointing towards the center of the heliostat's mirror, which is defined as:

$$\begin{aligned} \vec{O}_M = \vec{O}_1 & \\ & + R_{01}(\alpha, \beta) \cdot R_\theta(\theta_M / GR1 + \theta_k) \\ & \cdot \left(\vec{V}_{12} + R_{12}(\gamma) \cdot R_\tau(\tau_M / GR2 + \tau_k) \cdot \vec{V}_{2M}\right). \end{aligned} \qquad (3)$$

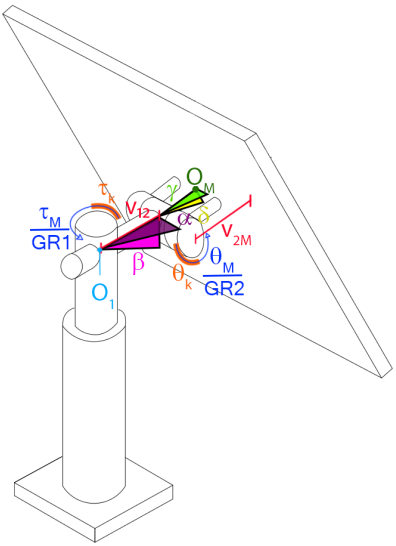


Figure 5: Close-up view of the heliostat geometry model with all error quantities used. τ_M and θ_M represent the measured motor positions.

$\vec{O}_1$ is pointing from the tower's base, which is the coordinate system's origin, to the heliostat. $\vec{V}_{12}$ and $\vec{V}_{2M}$ are suspension point displacements relative to the first and second movement axis (compare fig 4 and 5). Both are measured only once and are constants individual for each heliostat.
For $\vec{n}_{\text{model}}$, both rotation axes are positioned in the suspension point $\vec{O}_1 = \vec{O}_M$ of the heliostat. It is described by:

$$\begin{aligned}\vec{n}_{\text{model}} = R_{01}(\alpha, \beta) \cdot R_\theta(\frac{\theta_M}{GR1} + \theta_k) \\ \cdot R_{12}(\gamma) \cdot R_\tau(\frac{\tau_M}{GR2} + \tau_k) \cdot R_{2M}(\delta) \\ \cdot \vec{n}_{\text{ba}}.\end{aligned} \tag{4}$$

With $\vec{n}_{\text{ba}}$ describing the base alignment and all R functions being rotation matrices around their own relevant axes.
θ_M and τ_M are the current motor positions of the heliostat. Because the heliostats in Jülich use stepper motors, the positions are logged on each measurement as incremental steps. These increments can be converted to angles and vice versa with a geometric relation. The LM algorithm works only using the angle representation, so all measurements have to be converted before and after calculation. In case of non-linearities this leads to conversion errors.
The equations go back to *Khalsa et al.*[79] and were adapted to the heliostat geometry of the solar tower Jülich. The actual heliostat orientation to hit the calibration target/re-

ceiver can be calculated iteratively by computing the bisector between $\vec{n}$ and the sun vector.

After measuring a new calibration point, the existing points are separated into a training and a test data set (in general and also in this work regression algorithms also use a third validation set). As the names of the data sets suggest, only the training data set is used for the regression (training). However, since this can lead to a so-called overfitting, e.g. if too many variables are used, the actual prediction is checked in separate program sections (the test) using the test data set. However, no parameters are updated thereby. (If a validation data set is used, it is applied to the calculated model after the end of the training. The validation data set is usually intentionally more complicated than the training/test data set. For example, a prediction that lies further in the future to test the extrapolation capability of the calculated model). After calculation, the error parameters are stored, if the prediction accuracy of the test data set is better than any result before. It should be mentioned that this is the recommended routine for data-driven algorithms. However, it is not carried out in this way at the solar tower in Jülich. Nevertheless, in the course of this work, for data collected at the solar tower, it is pretended that this routine exists, so that the results are comparable and meaningful. Of the 24 free parameters described in the previous section, 8 which are mainly responsible for the heliostats in Jülich were selected. The selection was made by evaluating var-

ious publications [67][68] and field studies[69]. However, during the operation of the solar tower, it has become apparent that the template is partially inadequate. Especially for heliostats in the rear part of the field the predictions are not close enough to reality. It remains unclear why the model is inaccurate. This could be due to further yet unconsidered parameters, non-linear errors, external influences or due to failures of the centroid of mass determination preceding the LM algorithm. How accurate this method can be, will be discussed in detail in Section 5.2 and compared to newly derived models afterwards.

Finally, it should be noted that the Heliostat calibration has all the characteristics of a digital twin. All in all it is a rather primitive model, but nevertheless the heliostat field can be controlled with adequate accuracy without closed loop control. This already shows the potential behind a power plant wide digital twin unit.

In the following discussion the error based GM used in Jülich will be also called baseline model, because all accuracies of each tested algorithm for the heliostat calibration are validated against this model.

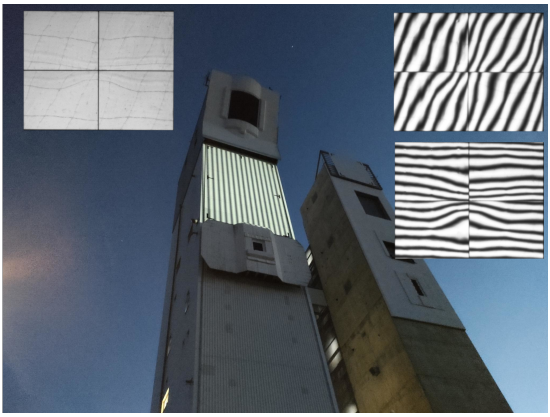


Figure 6: Picture of the solar tower during deflectometry. In addition, the heliostat is shown during the measurement with different stripe patterns (right edge). Even without the stripe pattern the deformation of the heliostat can be estimated from the reflection of the tower (upper left corner).

2.5. Deflectometry

The flux density on the receiver is composed of all focal spots of the individual heliostats. In addition to the alignment error, heliostats are also prone to mirror deformation. These are unwanted convex and concave dents and bulges. By shifting the focal spots center of mass, they not only negatively affect the heliostat calibration, they can also lead to unwanted temperature peaks on the

receiver. They occur due to incorrect *focusing*, *canting*[3], mechanical stress during build up, or aging. In order to predict the flux density realistically, a digital twin must also take these errors into account.
However, the defects are difficult to locate due to the reflective surface of the heliostat. Deflectometry is the most commonly used and recommended [80] method for the purpose of detecting mirror defects.
Thereby a two dimensional cosinusiodal fringe pattern of the form[81]:

$$I(x, y) = c + A_{Light}(x, y) \cos(\phi(x, y) + \Phi_k) \tag{5}$$

is generated (compare Fig. 6 on the tower). Where I is the total intensity, $A_{Light}(x, y)$ is the amplitude of the light and c is the background intensity, which can vary in a general case but is taken as constant in the application at the solar tower. ϕ is the phase including a phase shift Φ_k defined as:

$$\Phi_k := 2\pi \frac{k}{N} \tag{6}$$

[3]For heliostats, a distinction is made between focusing and canting. In canting, the initially planar mirror surfaces of the facets are aligned to a focal point so that the individual focal spots overlap at this point. In focusing, the mirror facet is additionally distorted concavely, which concentrates the individual focal spots more strongly.

with $k \in \mathbb{N} \wedge k < N$ and N being the number of images taken. Together they form a binary stripe pattern. Equation 5 can be written as[81],

$$I_R = \frac{A_{Light}}{2}\left(e^{-i\pi 2\frac{k}{N}}e^{-i\phi}\right) + c + \frac{A_{Light}}{2}\left(e^{i\pi 2\frac{k}{N}}e^{i\phi}\right) \quad (7)$$

using the Euler equations. I_R is evaluated using each pixel. As long as $N \geq 3$ (due to the three unknown variables A, ϕ and c) the equation is theoretically solvable. The solution is not relevant for algorithms proposed in this work. Therefore, a detailed derivation is omitted at this point. For a deeper understanding of this measurement methodology, the reader is referred to further literature[81]. The calculated quantities can then be used to determine the normal vector of the surface using simple geometric relations. Using deflectometry provides results with extremely high accuracy, which means up to 100k and more normal vectors can be recorded on the entire heliostat surface. In reality however, the measurement is very unstable with only three images. For this reason usually multiple images are accessed. In addition, the frequency and direction of the stripes are changed to resolve ambiguities.
At the solar tower in Jülich the stripe pattern is projected on the calibration target during night. The reflection of the pattern is then observed by a camera focusing on an aligned heliostat (compare Fig. 6)

Despite the knowledge on the mirror surfaces being rele-

vant for power plant operators, in practice it's often neglected. Although first publications on deflectometry in the context of solar power plants date back to 2011 [82], it was not managed to automate the measurement until today. It is still slow, associated with high maintenance effort and thus high costs.
This is due to many reasons. The quality of measurements is strongly dependent on extrinsic conditions. These include quantization of the intensity, non-linearity of the detector, unwanted background variations in c and other types of optical and electronic noise[83]. In addition to these problems, which are generally inherent to the method, there are several others at the solar tower. First, it is very sensitive to measurement inaccuracies. This includes distances between the tower and the heliostat (usually deflectometry is used at short distances) and macroscopic misalignment of the spectated objects. Second, external conditions such as dust, morning dew, both on the camera and the heliostat also affect the measurement. In addition, the camera (especially for heliostats further back in the field) must expose for a very long time for a good brightness value. While the camera shutter is open, e.g. no wind may blow by which the heliostat might be tilted. The evaluation of the measured values given by the different exposure times is also unstable. Finally the measurements have to be repeated regularly, since the deformation changes over time.

In this work, deflectometrically measured surfaces are used to generate focal spots of individual heliostats using ray tracing. The thus improved ray tracing process reproduces reality with extremely high precision and is used as a *ground truth* to evaluate later presented methods (see also Sections 3.1.2-3.1.4). However, due to the unreliability of deflectometry applied at solar towers, this type of measuring is not commonly used at solar towers and also not considered as a data source for a digital twin.

2.6. Flux Density Prediction with Ray Tracing

Ray tracing is the basis for the digital twin environment presented later and is, also in its current form, one of the most frequently used algorithms at solar power plants.
In general, ray tracing is a rendering technique to generate 2D images from 3D, well defined scenes. Starting with a scene which includes all relevant information about the inherent objects, such as the geometry, material properties, atmospheric conditions or camera parameters, the image is calculated by sending out rays through this defined volume. The rays are emanated from a well defined light source. When the rays hit a surface, they can be either reflected (mirror surface), absorbed (black body), or scattered (white body). This happens until the unabsorbed rays fall into the camera. It should be noted, that this is a descriptive explanation of the ray tracing process. To reduce the computational costs most ray tracers use the

described procedure recursively.

The mechanism behind this is described by the rendering equation[84]. While solving the ray tracing equations (or the rendering equation) for entire rooms or landscapes can take hours to weeks, ray tracers such as the program STRAL [17] are optimized to solve the equation for thousands of heliostats in a few seconds.
The virtual scene is described by mainly three different components. The sun (distribution and position), the heliostats (discrete points defining the mirror's shape, and corresponding normal vectors), and the receiver (in simple cases a flat or rounded surface, but can also include calculations of further downstream processes). The same is true for the calibration target. In addition other environmental conditions, like atmospheric losses or cloudiness can be included. Each ray emitted by the sun and reflected at the heliostats carries a small percentage of the emitted energy of the sun. From the number of rays arriving at the receiver/calibration target, the irradiance can be calculated.

Even though the scene at the solar tower is much simpler than in many other ray tracing applications, the demand lies in the speed of the calculation, which must be done in real time and faster for entire heliostat fields. In order to reduce the absolute number of calculations, solar tower ray tracer like STRAL [17] do not trace back the rays from the image to the radiation source (or vice versa). Instead

the rays are generated directly on the heliostats. This assumption presupposes that the heliostat is completely and evenly illuminated. In this case, each surface element (or each uniformly distributed point) on the heliostat receives the same amount of energy. Depending on the normal vector at this element and the sun position, the outgoing vector is determined, which is then randomly perturbed according to the sun distribution and sent towards the receiver. This description neglects that heliostats can be shadowed by heliostats in front of them, or be blocked on the way to the receiver. Whether one of these cases occurs can be checked by setting up additional bounding boxes/-triangles, or by additional shadow rays[85]. If shading or blocking occurs, the contribution of these elements to the resulting image must be neglected. In the following, the topic of shading and blocking will not be discussed further, since it is not relevant for the experiments and simulations carried out. However, the procedure is also valid for these cases. Multiple reflections need not to be considered, as any additional reflection in the solar field will prevent the receiver from being hit. At last, the rays arriving on the (continuous) receiver surface are assigned to the individual pixels (*Bitmap*). This is done by an integer operation (mathematically a floor operation).The procedure for a single heliostat is summarized in detail in Algorithm 1 and structured as a flowchart shown in 7.

Algorithm 1: Ray tracer - main

```
Function Main(SunPostion, HelPosition, Aimpoint):
    initialize parameters ;    // Heliostat-, Receiver- and
     Sun-Shape
    heliostat ⟵ DefineHeliostat(HelPostion) ;    /* Heliostat
     is described by equally spaced points, lying
     on the ground with HelPosition as its origin.
     Each point has a corresponding normal vector
     */
    heliostat ⟵ RotateHeliostat(SunPosition,Aimpoint) ;
     /* Heliostat is rotated according to
     SunPosition and Aimpoint */
    fromSun ⟵ heliostat.points - SunPosition ;
    /* Raydirection from sun to point on heliostat
       */
    raydirection ⟵ ReflectRays(heliostat.normals,fromSun) ;
    Place N rays with raydirection * DistributedNoise on every point
     of heliostat ;    /* Scatters each ray around ray
     direction by a random value of a given
     distribution like pillbox, gaussian or sun
     shape. */
    for point on heliostat do
        for ray on point do
            intersection = CalculateIntersection(point, ray,
             Aimpoint, ReceiverNormal);    /* calculates
             the intersection between a line and a
             infinitely extended plane */
            if intersection is on ReceiverPlane then
                intersection ⟵ int(intersection); // convert
                 to integer value
                Bitmap[intersection]+=1
        total_bitmap += Bitmap ;    // sum up all bitmaps
    return total_bitmap
```

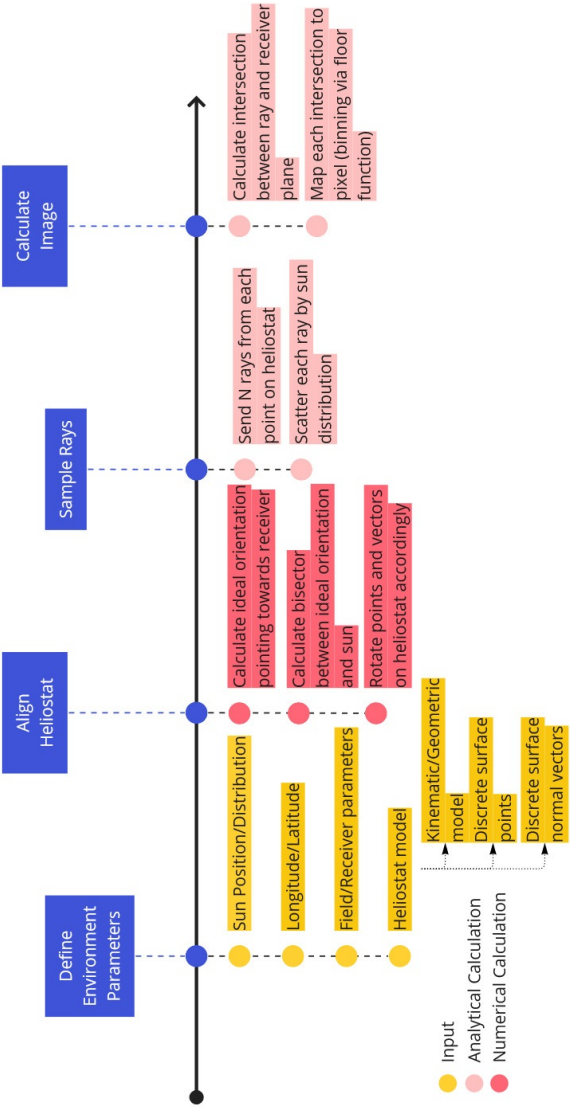


Figure 7: Basic procedure of ray tracing in STRAL

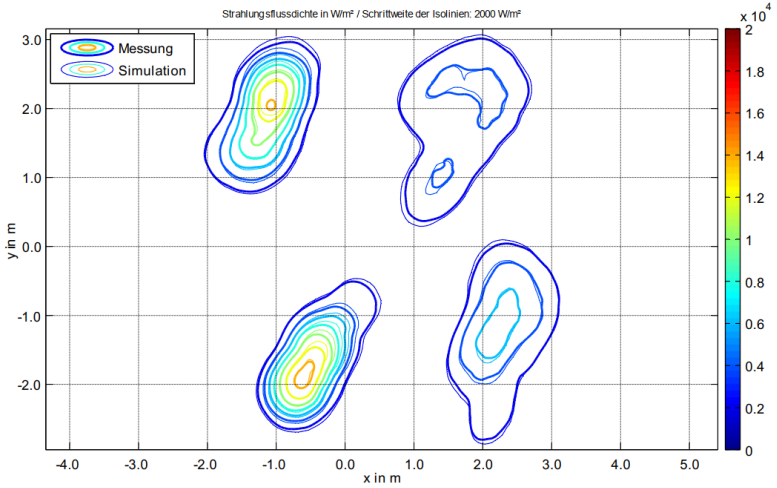


Figure 8: Comparison between measured and simulated flux density distributions using the ray tracer STRAL and a 5° circum solar ratio sun shape profile, done by *Belhomme*[86].

The simulations can be improved directly by providing the ray tracer with measurement data. For example, surface profiles and normal maps can be loaded directly from the deflectometry data to represent the heliostats (and their corresponding focal spots) more realistically. The high overlap of deflectometry enhanced ray tracing and real focal spot images could be shown by *Belhomme et al.*[17] in Fig. 8. In their work they found a similarity of over 97% to real measurements. Moreover, whole (erroneous) heliostat motion profiles, e.g. from the calibration, can

also be taken into account.
In contrast, without these measurements the simulation is rather inaccurate for a reliable flux density prediction during power plant operations. Thus it serves more as a guide for power plant operators, than as a realistic prediction (or even heliostat control).
However, providing the required data can be difficult. In the case of deflectometry the procedure is still not automated, so it is very laborious to (regulary) measure each heliostat in the field.
The calibration, in turn, is fully automated, but tends to have to little information (data points) on the aimed accuracy. Furthermore, the data-supply chain from measurement to ray tracer (measurement - image pre-processing - externally fitting - integration into the ray tracer) is unreliable and the errors are propagated further in each of these steps. The differentiable ray tracing environment introduced later (see Section 3.1.2) is intended to address both of these drawbacks, by processing measurement data directly within one environment. This shortens the supply chains and increases the information content per measurement, which enables completely new approaches.

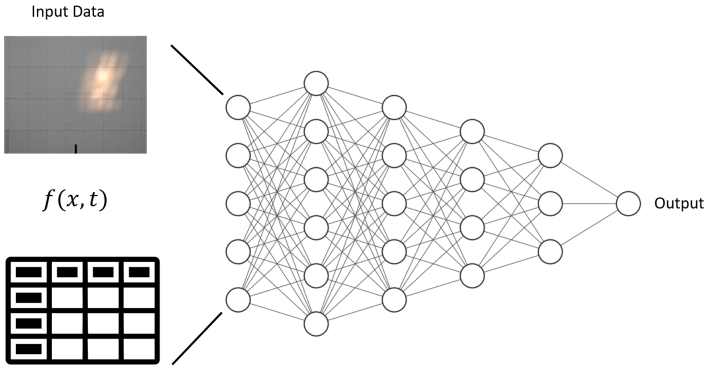


Figure 9: Schematic drawing of the MLP. It can map every input vector (e.g. functions, images, tabular data) on any output vector.

2.7. Neural Networks and Multi Layer Perceptrons

Black box models like neural networks are in direct competition with white box models already in use at solar towers. These purely data-driven algorithms dispense with physical traceability, but often achieve significantly higher accuracies. They are to be used on solar towers when the data sets have already grown considerably.

Their high accuracies can be explained by their ability to map any input vector to every output vector[87]. For example, this includes images, function values as well as tabular or time dependent data. It is mathematically proven

that a neural network with more than one hidden layer can act as a universal function approximator [87]. However, this mapping function has to be learned from data within the so called training process. This stores the relationship between input and output within the network. After this computational expensive (but one-time) training process many advantages can be archived. This ranges from speed advantages of several orders of magnitude compared to classical algorithms[88], over the use of input parameters that are unsuitable for physics-based models. An illustrative example is that they can use images or videos directly as an input. Physical information does not have to be extracted first [89].
Modern NNs received remarkable results in many different areas, from classification [41], over regression [42], to imaging tasks [43]. By using modern deep network structures and the availability of a sufficiently large data set, even the most complicated tasks can be performed. The quality of the NNs output continues to improve as the amount of data increases, even when classical algorithms have long since ceased to derive any value from it [90].

Among all Deep Learning network structures, MLPs are the oldest and simplest. Here, all neurons of one layer are connected to all neurons of the previous and next layer. They were first used successfully to identify handwritten letters[91]. They also work well on tabular data (like the heliostat calibration provides) and are still widely used for this today [92]. Overall, however, they are rarely ap-

plied. They count as hard to train due to vanishing/-exploding gradients [93]. They often converge to poor local optimums [94] and nowadays there are a lot of more advanced, stable alternatives, for example Convolutional Neural Networks (CNNs).
Nevertheless, these networks have also made astonishing progress in recent years. With the use of self normalizing neural networks (SNNs) [95] they became deeper than ever before, which allows them to perform more complicated tasks. Due to more intelligent pretraining methods [42] and data preprocessing [55] high precision can be achieved with just a few hundred measurement points, which is why these networks have celebrated a comeback in recent years.
This property also makes them particularly interesting for use in the solar field. Because many neural network architectures are not suitable for use in the heliostat field, since the uncorrelated individual errors of the heliostats do not allow superordinate training with data from the entire field. Instead, only data sets of a maximum of a few hundred measurement points per heliostat can be used. Two network architectures, which have already shown in other domains that they can manage with very little data, will be presented in the following.

2.8. Neural Radiance Fields

To predict the individual focal spots of a heliostat, ray tracing is not necessarily required. Instead of physically

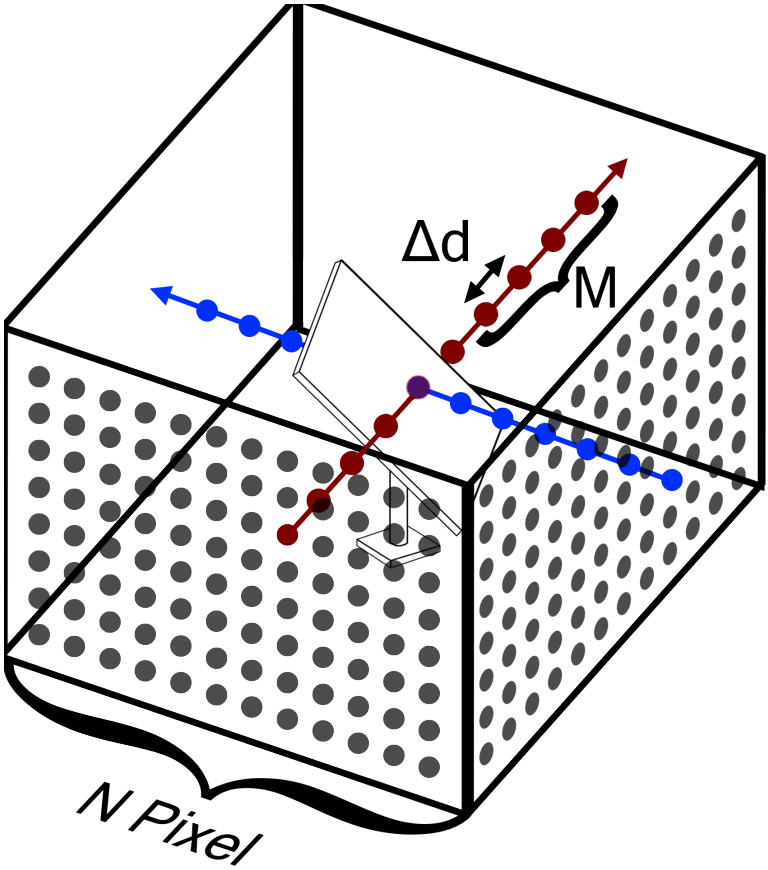


Figure 10: Schematic drawing of the ray pathing inside NeRF. Starting from each pixel a ray is emanated through the virtual volume and discretized in distance Δd. In this way the whole volume is discretized.

determining the shape of the focal spot, the information about the convolution function (sun distribution-heliostat-focal spot) can also be learned via data and stored within a neural network. One network type that has stood out on a very limited data set are NeRFs.
NeRFs are very new to the game and are the next big player when it comes to novel view generation. They belong to the upper class of differentiable rendering.
First presented by *Mildenhall et al.* [55] at the end of 2020, it was possible to reconstruct near photo realistic 360-degree views of any object from just 25 to 100 images and their camera position. NeRF uses very simple network structures. Two 10-layer deep MLPs form its core. The first (coarse) network is used to find the structure of the object, the second (fine) network determines the required color of the image's pixel.
The ingenious idea of NeRF is a type of preprocessing. Instead of giving the network (only) 100 images (for example, Image Numer x Channel x Height x Width = 100 x 3 x 128 x 128 = N) as pixels to the network, the images and their viewing direction are used for a preceding pathtracing algorithm. Starting from the camera position, one ray per pixel (Fig. 10 red and blue line) is shot into a virtual volume. Within that volume's boundaries these ray paths are discretized ($M \cdot \Delta d$). The sum of the points along the ray must then correspond to the color, predicted by an approximation of the rendering equation[96]. By looking from different viewing angles, these voxels are used within the volume along multiple ray paths and a loss is formed

from the cross correlations. In this way, the formerly N pixels become $N \cdot M$ spatial points, which multiplies the amount of data by far. Also instead of predicting a whole image at once, the network only has 4 outputs, namly RGBA (Red, Green, Blue, Alpha the opacity), which correspond to one voxel of the virtual volume. This reduces the complexity of the task by far. This together with the physically motivated regularization makes NeRF an incredibly competent tool for image generation.

In its original publication (to which this work refers to), NeRF required static scenes and lighting conditions as well as hours to days of computing time. But all these conditions were softened or completely eliminated within a few months by an avalanche of subsequent publications. Very quickly, dynamic lighting conditions became possible [97]. Dynamic scenes [98][99] followed shortly after. A little later results were even generated with less than 10 images [100]. Almost one year later NeRF can now render new scenes in realtime [101][102].

Nevertheless, the data and also the situation at the solar tower are not directly applicable to NeRF. Therefore, the data produced by the heliostat calibration must still be adjusted, the process of which is described in Section 3.2.2.

2.9. Pretrained Neural Networks

NeRF achieves its impressive results mainly through intelligent data pre-processing and physical regularizations. However, networks can also be regularized in a purely data-driven way. If the nodes of a network are not randomly initialized, but some training has been done beforehand, it is called *pretraining*. By preactivating the nodes needed to solve the task in advance, the local parameter space becomes more complex, which renders it more difficult to travel significant distances via the backpropagation method [103] and therefore restricts (regularizes) the process. After the pretraining, the actual training process starts, using the pretrained neuron weights and biases. In order to preserve the pretraining progress, the *learning rate* is, in general, considerably lower.

Pretraining can be distinguished in two different types: *supervised* as well as *unsupervised* pretraining.
In the case of supervised pretraining, the neural network is pretrained with a *source data* available in sufficiently large quality. Afterwards the network is trained with the so called *target data set*, which can be considerably smaller. This is also called the *transfer* step. The two data sets can be very similar. For example, simulation data can be used as the source and the real data as target data set. However, this is not a necessary criterion. Often, a similarity in the problem task, e.g. in image recognition/-classification, is sufficient.
For example, a network can be pretrained with image data

of airplanes, houses, lions etc. and then be trained to analyze images for cancer cells [104].

In the *unsupervised* case, no extra data set is necessary (but can be used). Instead, the network tries to capture the structure of the source data set's input distribution. Particularly noteworthy in the context of this work are the results of *Feng et al.*[42]. Their pretraining takes place in a so-called greedy layer wise process. For this every layer of the neural network is trained individually inside a shallow Stacked Autoencoder (SAE), a symmetrical neural network where input and output layer are the same (compare Fig. 11). The deep neural network is then initialized with the trained hidden layers of the autoencoders for the actual training process. Under the assumption that it is easier to train several flat networks than one deep network, this approach helps to bring the network close to a local optimum even before the actual training.
Feng et al. [42] were able to predict highly correlated material defects by means of an MLP and this with a data set consisting of only 500 tabular data points.
This work will look at similar network types and test for their suitability for heliostat calibration. For this purpose, beside the unsupervised pretraining by *Feng et al.* different supervised pretraining attempts will also be presented in Section 4.2.4.

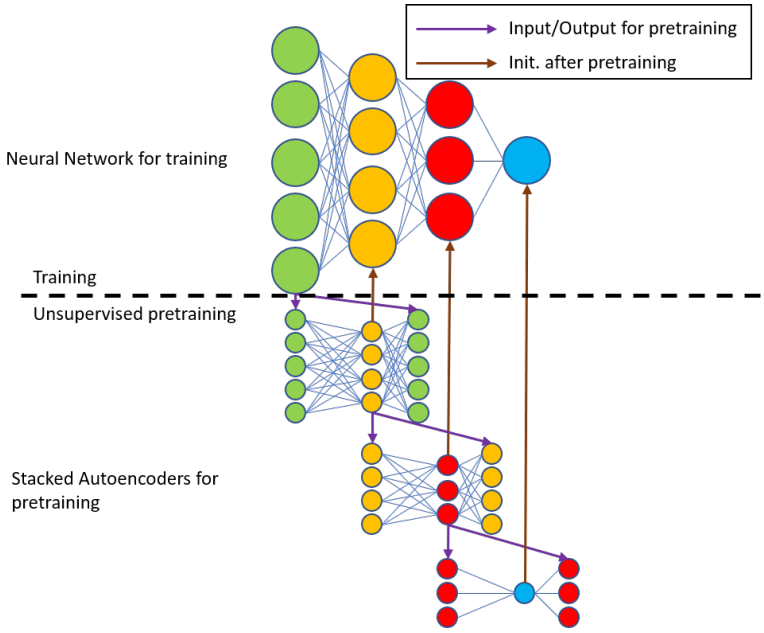


Figure 11: First unsupervised pretraining step. In this "greedy layer-wise" pretraining several autoencoders are created. The first takes the original training data as input and output layer and is trained until no enhancement is recognizable anymore. After this, for every layer of the original neural network, a new autoencoder is created and trained using the hidden layer trained on step earlier as the new input and output layers. After all autoencoders are fully trained the original neural network is initialized by the trained hidden layers of each. After this the real training takes place.

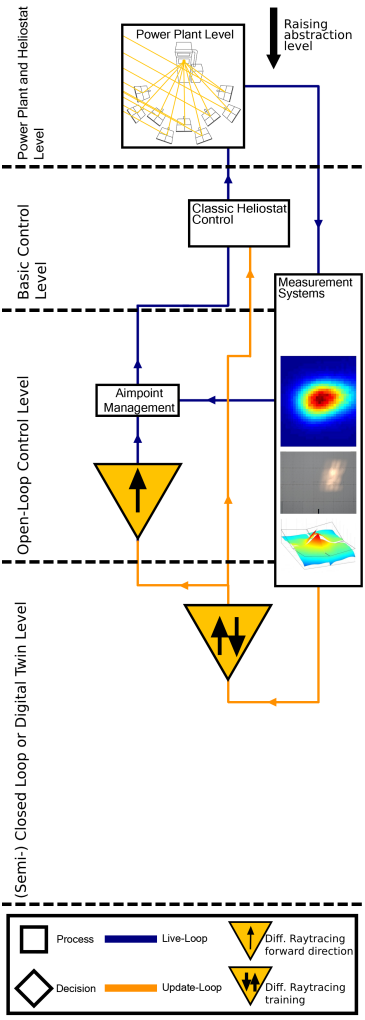
3. Methods

Based on the algorithms and measurement methods presented before, the methods that have been developed and applied in the course of this thesis will now be discussed. In the following the algorithms which were developed and examined within this work will be presented. First and foremost differentiable raytracing is presented. This method acts at the solar tower as the Digital Twin Environment. It can create, update and inherit other Digital Twin Instances like physical modeled digital heliostats or neural networks.

How the DT environment is integrated into the existing process levels will be shown in this context (Section 3.1). Then the DT environment itself and the therefore necessary differentiable description of the solar area will be discussed (Section 3.1.2). In this thesis, the environment will be used to optimize both the heliostat calibration and the surface error of heliostats. Also, regularizations of the heliostat surface are presented, which are useful for the convergence of the new algorithms (Section 3.1.3). Furthermore, the theoretical limitations of surface reconstruction will be derived from geometric relations (Section 3.1.4). Afterwards, the code implementation will be discussed 3.1.5. At the end of this chapter, the black box models competing with diff. ray tracing and their adaptations to the solar tower are presented (Section 3.2-3.2.2).

3.1. A Digital Twin Environment for Solar Tower Power Plants by Use of Differentiable Raytracing

For the implementation of Digital Twins in the solar tower power plant, a new level of abstraction is introduced, the *Digital Twin* or *Semi-Closed Loop* level. It aims to close the control loop by providing high-accuracy models for lower-level applications. For example, classical ray tracing calculations can now be updated with data-driven methods. the subsequent integration of digital twins ensures smooth and uninterrupted operation of the remaining process levels while training the new instances. Additionally, the derived models can be used autonomously, independent of the power plant process.



At the digital twin level, the differentiable ray tracer is trained with new data. Upon completing the training process, the trained heliostat digital twin instances or their parameters are transmitted to other level processes, such as heliostat control or the forward ray tracing module. By integrating digital twin models into state-of-the-art modeling and simulation tools, the Digital Twin level enhances the operational efficiency and performance of the Solar Tower Power Plant.
The ray tracer developed at the Digital Twin level can also be used for forward ray tracing at the Open-Loop Control Level. Parallelizing ray calculations and optimizing GPU usage enable rapid and efficient calculations. However, heliostat instances can also be transferred to established ray tracers like STRAL and SolarPilot.
In this work, the DT environment is utilized to derive heliostat-specific tracking errors, mirror deformations, and predict the flux density map from images of the fully automated heliostat calibration. The heliostat calibration is completely handled by the differentiable ray tracing, eliminating the need for the previous location of the process in the Open-Loop Control Level. Differentiable ray tracing also allows heliostat-specific surface diagnosis from focal spots for the first time. The found surface and corrected motion profile can then be used in the same environment for flux density prediction, encompassing all heliostat errors.

3.1.1. Differentiable Raytracing for Solar Tower Optimization

Differentiable ray tracing allows the classical ray tracing process to be performed in reverse. This means that the 3-dimensional scene no longer has to be completely defined, but conclusions can be drawn about the scene from the 2D image from measurement data. From this, quantities which would otherwise be very difficult to measure, can be obtained. For this purpose the ray tracer is integrated into a neural network training pipeline.

Diff. ray tracing starts like classical (forward) ray tracing by defining environment para-meters (material, light, geometry etc.) as well as camera ex- and intrinsics (camera angle, lenses, etc.). However, the object property to be examined only needs to be defined approximately. Often an matching topology is sufficient[29]. Using this coarse approximation the ray tracer generates different images. While training, the resulting image is compared with the *ground truth* (also called *target image*[4]). The comparison is done by an objective/loss function e.g. *L1* or *L2* pixel-wise losses. In fact other optimization algorithms

[4]The target on the solar tower which is used for calibration and the target image have two different meanings. The technical vocabulary of the two research topics solar towers and machine learning overlaps here. The target in machine learning means the ground truth to evaluate the prediction. The target at solar towers refers to the white area under the receiver, which is used for calibration. To avoid confusion, this work explicitly distinguishes between the *cal. target* and the *target*.

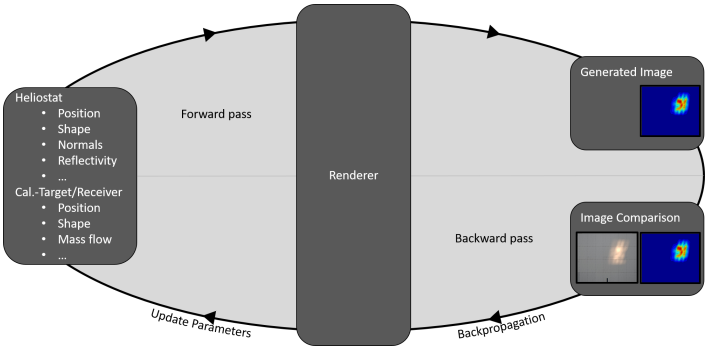


Figure 12: Differentiable ray tracing pipeline. The ray tracer can generate images from a given scene and also update the scene according to an evaluation/loss function.

like Adam and *schedulers* can be used in diff. ray tracing as well. The information about the deviation is then returned to the ray tracer, which updates the input parameters accordingly. This is done by means of *backpropagation*[105][106] as well as *automatic differentiation*.
The diff. ray tracing pipeline adapted to the solar tower is shown in Fig. 12. The optimization process at the Solar Tower Power Plant involves several steps to ensure accurate and efficient operation. Firstly, the scene is initialized with all available information about the environment, and the parameters to be optimized are determined, for example the receiver position and heliostats' surfaces and reflectivity in the field. Additionally, downstream processes in the power plant cycle can be optimized, provided that

they have a differentiable description.
The ray tracer uses the given scene to calculate a target/receiver image. This image is then compared to a measurement or simulation image and evaluated using different loss functions, depending on the optimization task at hand. After the evaluation, the input scene parameters are updated accordingly.
The diff. ray tracer is employed in this thesis for three distinct tasks at the Solar Tower Power Plant: The heliostat calibration, the heliostat surface diagnosis and the flux density prediction.

When using diff. ray tracing at solar towers, two aspects in particular must be taken into account. In order to use the backwards direction, it must be ensured that the scene under investigation is differentiable. How the differentiability in the solar field is guaranteed will be explained in the following Section 3.1.2.
However, the biggest difference between the application at the solar power plant and the common literature is probably that the objects under investigation are not directly spectated, but only their blurred reflections. Thus, for some optimizations the problem is strongly underdetermined. This is the case e.g. for the reconstruction of the heliostat surface from calibration images. The difficulty of the reconstruction is comparable with a passive non line of sight problem [107] rather than with a classic diff. ray tracing task. To counteract the underdetermination regularizations are needed, which will be discussed after-

wards (3.1.3). The quality of these regularizations can be measured by how far beyond the boundary to underdeterminacy a reconstruction of the surface is still possible. Where this boundary is will be discussed in Section 3.1.4. Finally, the implementation of the previously described theory on a code basis is presented in Section 3.1.5.

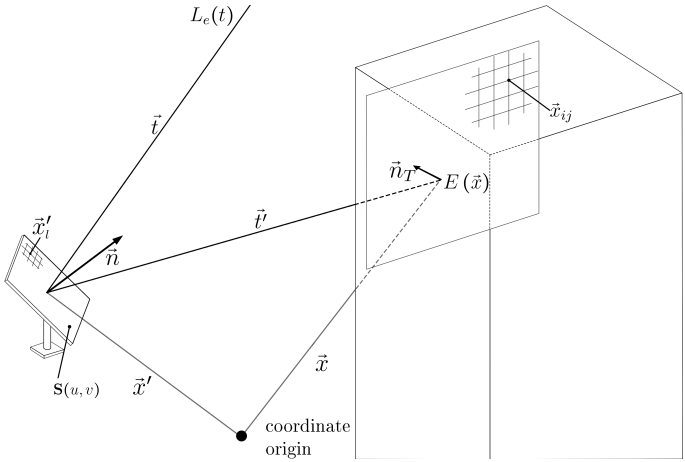


Figure 13: Representation of the used coordinate systems. The coordinate origin shown is for the differentiable ray tracer in the current implementation at the base of the tower. However, the formulation is independent of it.

3.1.2. Differentiable Description of Ray Tracing at Solar Towers

The flux density map at the receiver (irradiance) is composed of the superposition of the individual focal spots. The area of the receiver (or the calibration target) thus corresponds to our projection area in ray tracing. To obtain the flux density via ray tracing, we formulate the problem starting from the radiance. The radiance, a quantity that describes the radiation field in terms of power per

area and solid angle is given by L, which depends on the position $\vec{x}$ the direction $\vec{t}$. For non-absorbing media it is constant along lines.

$$L\left(\vec{x},\vec{t}\right) = L\left(\vec{x}+\lambda\vec{t},\vec{t}\right) \qquad \forall\,\lambda$$

The choice of light source is arbitrary within the ray tracing environment. A solar distribution function (e.g. Pillbox, Gauss, Circum Solar Ratio) is just as possible as the strip modulated projector of the deflectometry. Within this work a Gaussian solar distribution function is assumed.
The radiance is measured in Watt per area and solid angle ($\mathrm{W/m^2sr}$). The approximate size of the solar disc is $0.5°$[5] or 0.01 rad. In the following, the global illumination of the receiver will be neglected and only the radiance reflected from a heliostat will be considered. In order to obtain the irradiance at position x it is required to integrate over the solid angle. This includes the cos factor that is well known from e.g. the render equation[84, 96]. The irradiance E_x is considered at a position $\vec{x}$ on the calibration target or the receiver. In the following only the calibration target will be mentioned. However the derivation is true for both.

$$E\left(\vec{x}\right) = \int_{\Omega} L_r\left(\vec{x},\vec{t'}\right)\ \vec{n}_T\cdot\vec{t'} \quad \mathrm{d}\Omega. \tag{8}$$

with $\vec{n}_T$ the normal vector of the cal. target and t' the

[5]It varies over the year between 0.527°–0.545°.

distance between $\vec{x}$ and an arbitrary point $\vec{x'}$ in space, defined as

$$\vec{t'} = \vec{x} - \vec{x'} \tag{9}$$

Ignoring the direct sun illumination the integral in 8 only provides a contribution over the relative heliostat area, showing towards the cal. target. Thus, the integral can be transformed into an integral over the reflecting body

$$E(x) = \int_{A'} \frac{1}{|\vec{t'}|^2} L_r(\vec{x}, \vec{t'})\ \vec{n}_T \cdot \vec{t'} \quad \mathrm{d}A' \tag{10}$$

Assuming an ideal reflection, the reflected radiance L_r can be constructed as follows

$$L_r(\vec{x}, \vec{t'}) = \rho(\vec{t}) L_e(\vec{x}, \vec{t}) \tag{11}$$

In a first approximation it can be applied that the radiance coming from the sun is identical on every point on the power plant's area, so that applies:

$$L_r(\vec{x}, \vec{t'}) = \rho(\vec{t}) L_e(\vec{t}) \tag{12}$$

For this the reflection conditions must be met, otherwise the contribution is 0. Furthermore applies:

$$\vec{t} = M \cdot \vec{t'} \tag{13}$$

with

$$M(n) = \mathbb{1} - 2\vec{n}^T\vec{n} = \begin{pmatrix} 1-2n_1^2 & -2n_1n_2 & -2n_1n_3 \\ -2n_1n_2 & 1-2n_2^2 & -2n_2n_3 \\ -2n_1n_3 & -2n_2n_3 & 1-2n_3^2 \end{pmatrix} \tag{14}$$

with $\vec{n}$ the normal vector of the heliostat. Inserting 12 into 10 yields

$$E(x) = \int_{A'} \frac{1}{|\vec{t'}|^2} \rho(\vec{t}) L_e(\vec{t})\ \vec{n}_T \cdot \vec{t'} \quad \mathrm{d}A' \tag{15}$$

and with 13

$$E(x) = \int_{A} \frac{1}{|\vec{t'}|^2} \rho(M\vec{t'}) L_e(M\vec{t'})\ \vec{n}_T \cdot \vec{t'} \quad \mathrm{d}A \tag{16}$$

The point $\vec{x}$ on the target is therefore composed of all reflections of the surface of the mirror which fulfill the reflection conditions. This integral can now be solved numerically, leading to a ray tracing formulation of the problem. Except for a constant prefactor this leads to:

$$E\left(\vec{x}\right) = \sum_{\mathrm{pos}\, l} \frac{1}{|\vec{t_l}|^2} \rho(M_l\vec{t_l}) L_e\left(M_l\vec{t_l}\right)\ \vec{n}_T \cdot \vec{t_l} \tag{17}$$

Until now, the mirror was assumed to be ideal. All normal vectors used were therefore parallel. In reality, of course, this is not the case. Instead, the rotation matrix M is re-evaluated at each point l of the heliostat. To solve

an area integral by Montecarlo integration, it needs two sums. Both sums are contained in the sum over l and the positions are passed through as one list.
Typically, the irradiance on a single point $\vec{x}$ is considered to be on a square lattice $\vec{x}_{ij} = \vec{x}_0 + i\lambda\vec{e}_1 + j\lambda\vec{e}_2$, where λ is the lattice constant - or resolution of the generated images and the vectors $\vec{e}_1$ and $\vec{e}_2$ span the directions. This can be expressed by an additional discretization factor, or weights w. resulting in:

$$E\left(\vec{x}_{ij}\right) = \sum_{\text{rays } k} \sum_{\text{pos } l} \omega_{ijk} \frac{1}{|\vec{t_l}|^2} \rho(M_l \vec{t_l}) L_e\left(M_l \vec{t_l}\right) \ \vec{n}_T \cdot \vec{t_l} \quad (18)$$

With this the problem is fully described. For the equation to be fully differentiable, this must be true for all function terms.

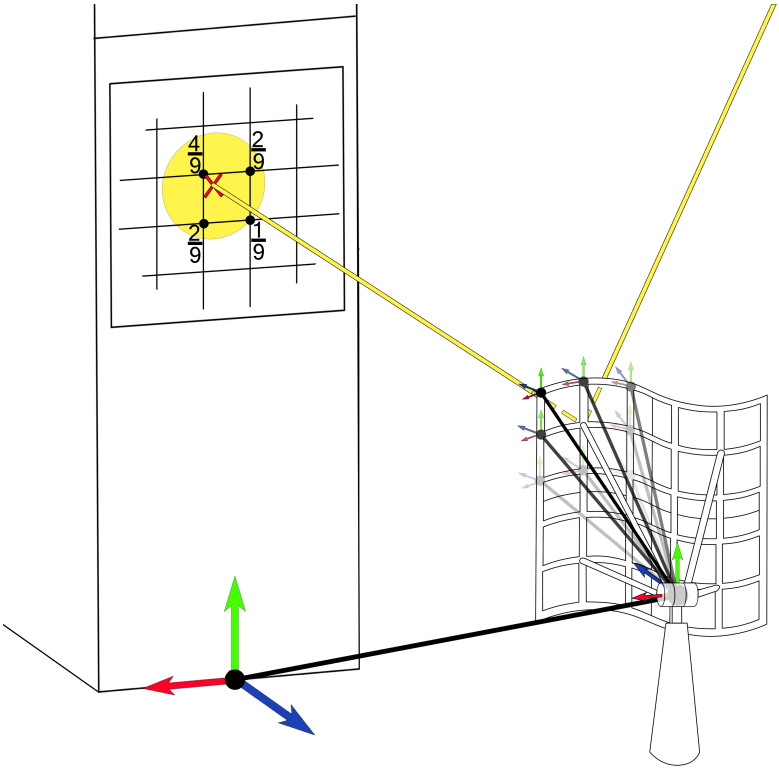


Figure 14: Schematic drawing of the ray tracing process including the binning function. The calculation starts at a heliostat lattice point. There, the incoming ray is reflected and traced to the target. Then the incoming ray's intensity is linearly distributed to the next nearest lattice points. It also shows how the different coordinate systems (tower $\rightarrow$ heliostat $\rightarrow$ facet point) are linked together. A rotation of the heliostat automatically influences all downstream coordinate systems.

For w that is usually not the case. Instead it is defined as follows

$$1 = \omega_{total} = \sum_{\text{rays } k} \sum_{0<n<N} 1 - \left(\frac{\vec{x_k} - \vec{x_n}}{\sum(\vec{x_k} - \vec{x_n})} \right) \tag{19}$$

where N is the number of considered nearest neighbors and x_n is the nearest neigbor position of the defined lattice. To get the contribution at the point i, j, we evaluate the function at exactly this point

$$\omega_{ij} = \sum_{\text{rays } k} \sum_{0<n<N} \left(1 - \left(\frac{\vec{x_k} - \vec{x_n}}{\sum(\vec{x_k} - \vec{x_n})} \right) \right) \delta(x_n - x_{ij}) \tag{20}$$

The binning procedure is schematically drawn in Fig. 14. The idea behind this comes from the solution of the Lattice Boltzman equations [108], where the collision directions of the particles are discretized in a similar way.

The parameter which has been neglected in the discussion so far is M. The rotation matrix M is itself part of the group SO(n) and thus continuously differentiable, but it is implicitly dependent on the normal vector $\vec{n}$. The result $E(x_{ij})$ is therefore highly sensitive to changes of this vector. $\vec{n}$ itself depends on the heliostat orientation (also an operation of the SO(n) group) and the position on the surface. In principle, the orientation of the normal vectors of each surface element can also be described as a rotation.
It is important to note that the regime is highly under-

determined. Obtaining a relistic deformed mirror by optimizing every surface is very unrealistic, due to the associated high number of optimizable parameters.
To address this issue, regularization techniques are necessary to effectively describe the surface and prevent overfitting by constraining the optimization process and reducing the complexity of the model.

3.1.3. Non Uniform Rational B-Splines for Heliostat Description

In order to regularize the surface of the heliostat, it is physical properties are taken into account. A heliostat consists of $X \in \mathbb{N}_0$ facets. Each of these facets is a continuous surface with defects described by dents and bulges. Rotational fields or branching on the surface do not occur. Taking advantage of these properties, the heliostat can be described by a continuous surface function for each facet. Arbitrary non-branching continuous surfaces can be represented by so-called Non Uniform Rational B Splines (NURBS). A NURBS surface is composed of different *B-spline* functions and their weighted control points. Each point of the NURBS surface is thereby uniquely defined by a set of points $\mathbf{P}$ (control points),$\mathbf{W}$ (weights) $\mathbf{U}$,$\mathbf{V}$ (surface vectors), with which applies[109]:

$$\mathbf{S} = f(\mathbf{P}, \mathbf{W}, \mathbf{U}, \mathbf{V}) \tag{21}$$

Evaluated at a (arbitrary) point the surface (also compare Fig. 13) is defined as follows

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u) N_j^q(v) \omega_{ij} \mathbf{P}_{ij}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u) N_j^q(v) \omega_{ij}} \tag{22}$$

with

$$\begin{aligned} N_i^p(u) =& \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u) \\ N_i^0(u) =& \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{23}$$

Here N_i, N_j represent the B-spline basis functions in the representation of *Curry and Schoenberg*[110]. The degree of the polynomial is freely selectable and is a strong regularization. The NURBS degree determines how many nodes are affected by a node's changes. The smaller the degree the more local the modification can be. For the mirror facet, a higher degree therefore also means higher stiffness.

The points u and v can query the function S at any (continuous) position. If the discretized heliostat is described by such a continuous function it can be represented exactly at the discrete positions l from formula 3.1.2 (compare Fig. 13), so that applies:

$$S(u,v) = S(x_l') \tag{24}$$

At the discretized heliostat grid points i, j applies:

$$\vec{n_{i,j}} = \vec{n}(S(x_l')) \tag{25}$$

Thus, each point and normal vector on the surface of the heliostat is uniquely defined by the NURBS function. So we can derive equation 18 directly according to the surface changes on the heliostat by means of automatic differentiation. Within the ray tracing environment the initial NURBS surface is chosen so, that N control points are evenly distributed over the heliostat's surface. Schematically this is visualized in Fig. 15 by the red dots. How close the control points are, can be freely selected and even adjusted depending on the training progress. If all weights are set to 0, the two surfaces are identical as long as enough control points are chosen to describe the surface completely. For the ray tracing process, any number of points M are sampled along the surface (blue points), where $N \ll M$. The position of these M points can be identical with e.g. those of the measured deflectometry surface data. At these points, rays are reflected and transmitted to the receiver/cal. target. The resulting focal spot can be compared with a measured focal spot and the heliostat surface adjusted accordingly.
The further the heliostat is away from the tower the more rays overlap. So the surface can not be reconstructed unambiguously. How far such a reconstruction is possible depends on the regularization and the distance.

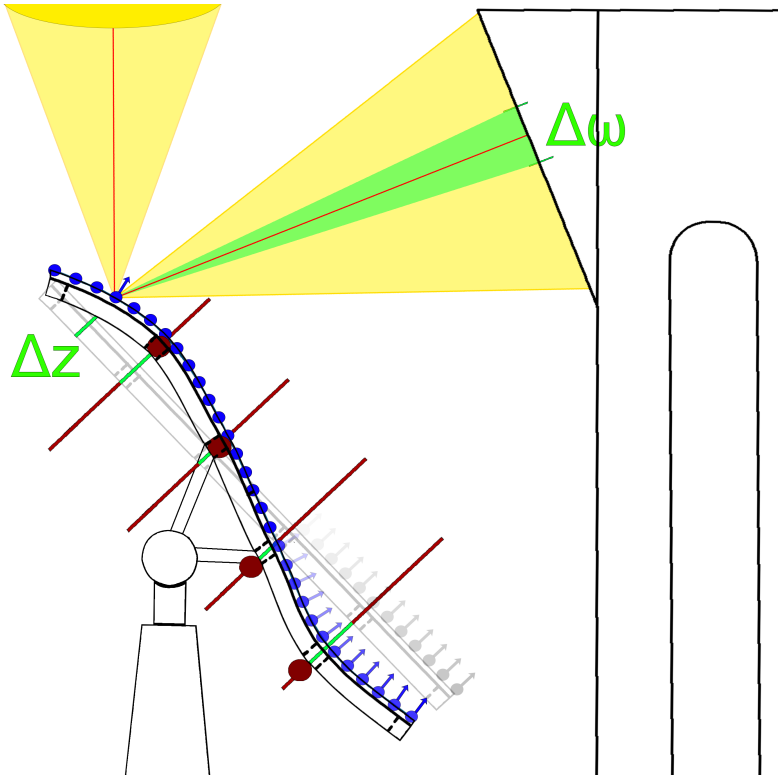


Figure 15: Schematic drawing of a heliostat NURBS surface. The controlpoints **P** (red dots) are shifted by their weights **W**(green section on red line) away from the ideal surface (grey). The deformation effects all neighboring discrete points and normal vectors (blue). One discrete point is moved by Δz which is influencing the ray direction by $\Delta\omega$. The information about the infinitesimal change $\Delta\omega$ can be traced back via auto differentiation to the change of the NURBS control point.

3.1.4. Resolution of Flux Density Based Surface Diagnosis

The resolving power is defined as the smallest distance between two points on the surface of a heliostat which can still uniquely be distinguished from each other at a distance of d from the projection surface. In the case of parallel incident rays and parallel normal vectors this would always be given. However, due to the widening of the rays caused by the aperture angle of the sun and the non-parallel normal vectors of the heliostat, the rays increasingly diverge and overlap. The stronger this superposition is, the more difficult the reconstruction becomes. The ambiguity of the problem increases from the point where the diverging rays cross.

In Fig. 16 different distances $(B - E)$ to point A are plotted. Starting from these points the rays widen (in relation to the sun by 0.5 degree). For the smallest distances AB_1 and AB_2 the rays coming from B_1/B_2 and A overlap at S_1 and S_2 for the first time. From S3, which is exactly at two times the distance of $S1/S2$ in the vertical direction of propagation, the overlap grows steadily (marked in black).
It can be assumed that the resolving power decreases continuously from $S3$. The point of overlap can be shifted according to different light source apertures as well as incident angles and the surface normals (see Fig. 17). In order to determine the intersection point S_3, S_1 is determined first. This results, as shown in figure 17, from a

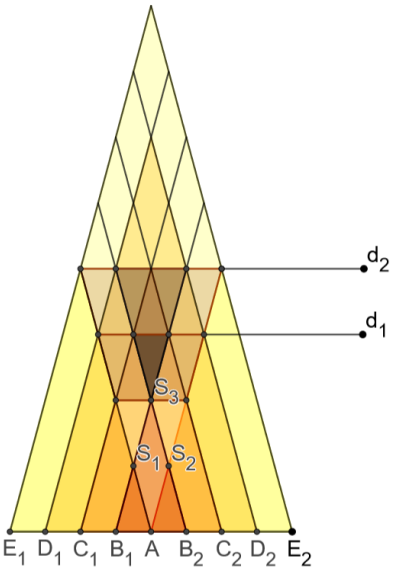


Figure 16: The mirror surface (E1-E2) divided into smaller distances. (A-E). The given (constant) aperture angle of the sun causes a beam expansion, which overlaps more and more starting from S1. The superposition between d_1 and d_2 grows overproportionally.

simple triangular consideration of the problem. Let the elevation of the sun be given as:

$$Ele = 90 - \theta_e \tag{26}$$

The incident ray (emitted from the sun) t_e is reflected at the surface normal n_i.
The angle $\theta_{a_1} - \theta_e$ thus results from

$$\theta_{ea} = \theta_{a_1} - \theta_e = \arccos\left(\frac{\vec{t_e} \cdot \vec{n_1}}{|\vec{t_e}\vec{n_1}|}\right) \tag{27}$$

Seen from the perpendicular, the angle between n_1 and t_r can be written as

$$\theta_{out} = \theta_{a_1} + \theta_{ea} \tag{28}$$

t_e is outgoing from the center of the sun. Due to the real expansion of the sun, the ray is expanded by 0.5 degrees ($2\theta_S$) and results in the 2 rays t_{exp} emanating per point. Thus, the inner side of the triangle b is described by the exterior angle:

$$\theta_b = \theta_{out} + \theta_S \tag{29}$$

The heliostat and the surface sections considered (in the range of centimeters) are much larger than the surface defects (millimeters and smaller). With the assumption that $d >> a$, it follows that $d' = d$.
Thus the interior angle of the triangle γ can be expressed

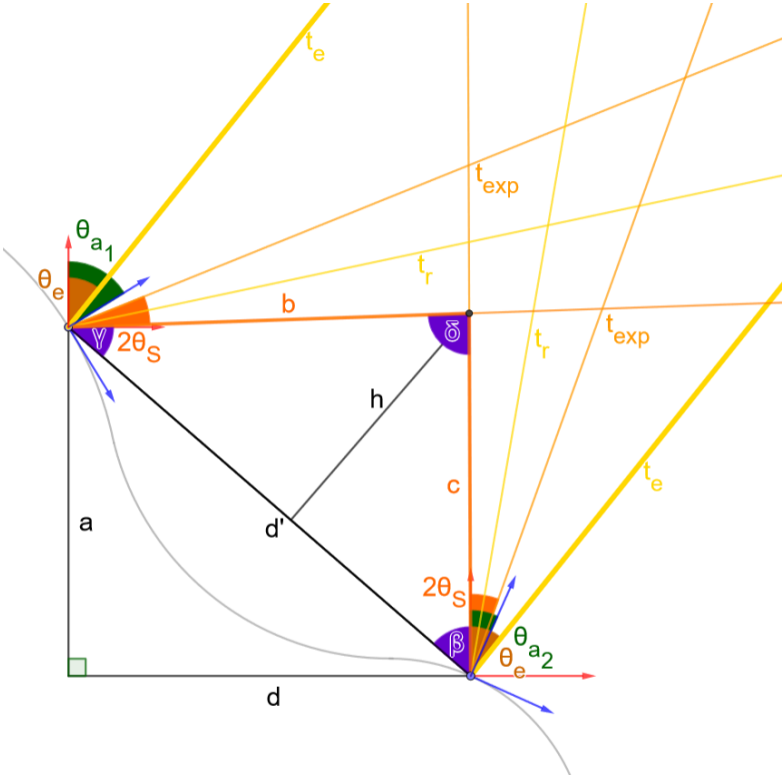


Figure 17: Trigonometric observation to determine the height h. The heliostat (gray) is strongly curved, however, only the features at distance d' are considered.

by

$$\gamma = 90° - (\theta_{a_1} + \theta_{ea} + \theta_S + \theta_{R_1}) \tag{30}$$

Where θ_{R_i} represents the microscopic roughness of the mirror and is only determined statistically. The same follows for β:

$$\beta = 180 - (90° - (\theta_{a_1} + \theta_{ea} - \theta_S) + \theta_{R_2}) \tag{31}$$

Where the sun angle is subtracted in this case to consider the smallest possible triangle. δ now follows from the sum of angles in the triangle. Since all angles and d' are known it follows from the sine theorem

$$\frac{d'}{sin(\delta)} = \frac{b}{sin(\beta)} \tag{32}$$

$$b = d\frac{sin(\beta)}{sin(\delta)} \tag{33}$$

and from Pythagoras

$$sin(\gamma) = h/b \tag{34}$$

and thus finally

$$S_3 = 2h = 2 \cdot d\frac{sin(\beta) \cdot sin(\gamma)}{\sin(180 - (\beta + \gamma))} \tag{35}$$

From this, first estimates for the procedure can be derived.

When considering an ideal, flat mirror lying in the xy-plane, the following applies

$$\beta = Ele - \theta_S \tag{36}$$

$$\gamma = Ele + \theta_S \tag{37}$$

In reality, however, the heliostat is always aligned so that it reflects the sun's rays onto the tower. In general, the mirror must be aligned in such a way that the normal of the mirror surface is in the bisector between the target point and the sun. Assuming that the target point is also in the xy-plane, it can be assumed that the heliostat is in the angle $Ele/2$. In figure 18 an estimation for S_3 (in meters) was made with these assumptions. The angle of solar incidence relative to the heliostat surface is considered. The more perpendicular the sun is to the heliostat, the later the rays overlap for the first time. The larger the distance of the considered error, the more the S_3 focal point shifts depending on the angle. For example, considering an error of about 1 m, the position of the sun can change the focus by up to 80 m. In other words, the more perpendicular the sun angle is to the tower, the better the resolution of the procedure.
The graph shows error sizes up to two meters. In Jülich, this would mean that a bump or dent spans the entire heliostat. Defects that are less than 20cm apart are hardly resolvable according to the figure. In order to understand these sizes correctly, it must be pointed out again that these numbers represent a limit exclusively for an unreg-

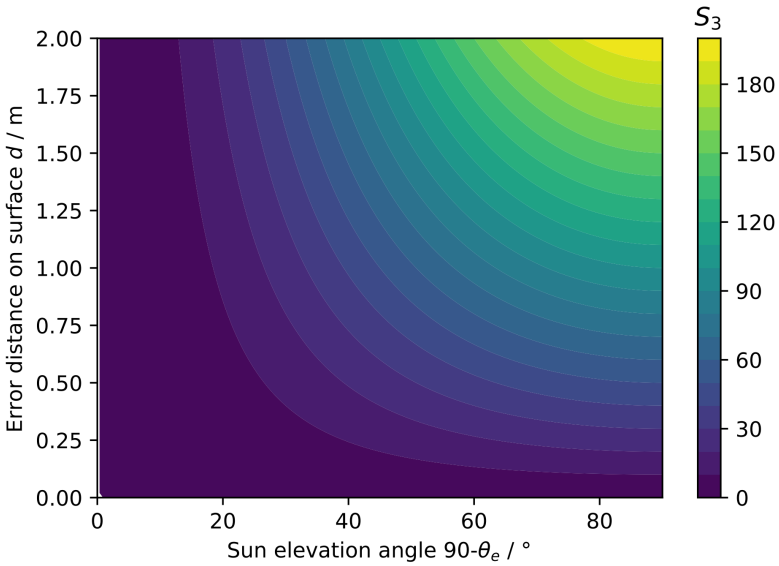


Figure 18: Estimation of the resolving power for different incident angles and mirror defect sizes assuming an ideal flat mirror. Regularizations can enlarge the distance.

ularized system. Where this limit actually lies depends on the selected regularization. Significantly higher limits can be achieved with a proper regularization.
The error quantities θ_a, θ_R, which are not considered here, depend on the surface and can shift the point S_3 forward (beam widening) or backward (beam focusing). However, both quantities are usually one dimension smaller than the sun aperture angle and shall therefore not be estimated here.

Another situation applies if the mirror surfaces are additionally deformed (focused) or the individual facets are arranged at an angle to each other (canting). In these cases S_3 can actually wander arbitrarily. In a heliostat field the focus point is usually set to a range that all rays are focused at one point on the receiver. In this case the problem is not only highly underdetermined but also freely permutable between the facets. It is therefore not possible to locate an error to a specific facet. As has already been shown, the angle of the sun also influences the overlapping of the rays. Thus the focal point can be shifted artificially by observations at different times (sun positions). Shifting the focal point also changes the overlap of the facets, which slightly reduces the free permutability. How much information the additional images bring will be discussed in Section 5.5. If a power plant operator is concerned about the physically correct reconstruction of the heliostat surface, a calibration target should be chosen which is not directly in the focal plane of the heliostat. Other

solutions would be the use of other light sources with a higher local resolution, for example, as in the stripe pattern deflectometry. In contrast to the latter, ray tracing only relies on a single image, which has some advantages. This possibility will be discussed in more detail in the outlook Section 6.3.

3.1.5. Code Implementation

Ray tracing techniques that are used in solar tower power plants are optimized to fit the specific (reduced) complexity of solar towers, and are designed to perform efficiently and faster then real time. In contrast, more general currently available differentiable ray tracers (e.g. *Li et al.*[27]) are predominantly designed to represent mostly static scenes with high complexity hampering their calculation speed. The implementation of differentiable ray tracing in solar tower power plants requires reconciling these opposing approaches.

On the one hand, in order to maintain performance for the forward direction, it is necessary to preserve the features of a solar tower power plant oriented ray tracer.
On the other hand, to enable the backward direction, it is required that the entire scene, including the heliostat kinematics, is fully differentiable and that gradients are provided for each calculation.
The optimization process itself is computationally intensive, and the ray tracing code must be designed to compute the rays in parallel to enable fast computation of

Algorithm 2: Differential Ray Tracer for Surface/Position/Material Properties

```
Function OptimizeObject(Epochs, Images, LightPostions,
  LightDistribution NumRays, CameraPlane):
    ReflectingObject ⟵ Load ideal shaped, discretized object;
    if Fit Surface then
      | ReflectingObject ⟵ FitNurbsSurface(ReflectingObjeckt)
    end
    Bitmap ⟵ Create array with same height, width, resolution as
      CameraPlane;
    Optimizer ⟵ ChooseOptimizer ;   // like Adam, Adamax
      etc.
    Scheduler ⟵ ChooseScheduler ;   // like Exponential,
      Cyclic etc.
    for Epochs do
      for (Image, LightPosition) in (Images, LightPositions)
        parallel do
        for (NormalVector, DiscretePoint) on ReflectingObject
          parallel do
          Rays ⟵ GenerateRays(NumRays,
            LightPostion, DiscretePoint, NormalVector,
            LightDistribustion);
          for RayDirection in Rays parallel do
            Intersection =
              CalculateIntersection(DiscretePoint,
              RayDirection, CameraPlane);
            if intersection is on CameraPlane then
              | Bitmap ← UpdateBitmap(intersection)
            end
          end
        end
        Loss ⟵ Criterion(Image, Bitmap) ;  // Can also
          be calculated batch-wise
        Gradient ⟵ Differentiate(Loss);
        ReflectingObject ⟵
          UpdateObjectParameters(ReflectingObject,
          Scheduler, Optimizer, Gradient)
      end
    end
```

Algorithm 3: Diff. Ray Tracer - Intensity Binning

Sample Bitmap `Binning`$(\vec{x}_{intensity})$**:**

for x,y *in* $\vec{x}_{intensity}$ ***parallel*** **do**

$index_{low} \longleftarrow \lfloor x \rfloor, \lfloor y \rfloor$; `// floor operation for indexing`

$index_{high} \longleftarrow index_{low} + 1$; `// get upper index`

$distance \longleftarrow$ calculate distances$(x_{intensity}, x_{high}, x_{low})$;

`/* calculate distance between continuous and index. For example`

$x_{intensity_{low}} \longleftarrow x_{index_{high}} - x$ `*/`

$binned_{intensity} \longleftarrow distance_x \cdot x, distance_y \cdot y;$

end

$bitmap \longleftarrow (indices, binned_{intensity})$; `// place binned intensities in bitmap`

individual heliostats. Furthermore, processing millions of rays and multiple images requires powerful graphics cards, and thus the ray tracer should be designed to take advantage of modern hardware. A (differentiable) ray tracer that meets all these requirements did not exist at the time of this work. Moreover, the retrofitting of existing (differentiable) ray tracers proved to be ineffective, which is why a new code basis was created for this work.

The entire code has been implemented in *Pytorch* from scratch and will be made available open source under `https://github.com/DLR-SF/holisticDIRC`. By using Pytorch, the ray tracer is natively GPU compatible, capable of computing gradients and making use of highly optimized algebra. No other packages or libraries are needed. Nevertheless the diff. ray tracer is supported by different plotting tools (*Tensorboard* and *Matplotlib*) and uses *yaml* configs files (*Yacs*) for a better reproducibility of the various simulations. Even though the whole ray tracing environment was developed from scratch, some numerical calculations e.g. of the flux density via Monte Carlo sampling and the alignment function of the heliostats were inspired by STRAL.

The newly implemented main routine to update the object parameters in dependence of the calculated gradients is shown in Algorithm 2. The basic procedure is strongly oriented to neural network pipelines. In Section 4.1.3 the routine will be shown in a float chart (Fig. 26).

When it comes to the calculation of the resulting image, the binning, meaning the assignment of the rays to a respective pixel, must also be considered. The basic procedure is shown in Algorithm 3. In the current implementation, the nearest 4 neighbors were chosen for this. This is a compromise between the speed of the binning function and the smoothing of the images. The number of neighbors can be increased. How this affects the image or the required number of rays was not tested.

The NURBS were also implemented completely from scratch by focusing on reusing available machine learning frame work functions (also *PyTorch*) in order to avoid reimplementation on a lower level. This has the advantage of not having to compile the NURBS code, while profiting of the highly optimized differentiable linear algebra routines.

The NURBS were designed in such a way that they can also be initialized with surfaces that are not (simply) analytically describable. The weights and control points of the NURBS are then set so that the 0 position corresponds to the loaded surface. The most important NURBS parameters, e.g. control points number, NURBS order can be chosen freely. To further reduce the degrees of freedom, the movement axis of the NURBS can be restricted to one direction (for example along the z axis, as shown in Fig. 15). During training, the number of control points can also be increased dynamically (linearly and quadratically). Heliostat surfaces that have already been pre-

trained can also be reloaded and further trained with any number of control points. In both cases, the algorithm benefits from better starting conditions. Thus, a higher surface resolution can be achieved during training.

In general, the differentiable ray tracer was designed to be used for general purposes at solar towers. A great effort was made to generate a complete heliostat representation. Each heliostat model contains all surface data (ideal and real/artificial errors), its own motion profile and aim points. Different heliostat shapes can be created functionally and easily. The heliostats are faceted (with any number of facets) which can be individually aligned or automatically focused. All these sizes are natively optimizable.
All objects are defined within differentiable coordinate systems, which are implemented similarly to a Robotic Operating System (ROS)[111] (using *Pytorch3d*) for traceability. This means that all other objects can be optimized as well. Parameters to be optimized can easily be selected and deselected. Objects can be defined inside the environment or be loaded as CAD models into the ray tracer. Environmental data can be adjusted in global coordinates (latitude-longitude) to the respective location of the power plant and realistic sun positions can be calculated at any time or day of the year.

Although this work discusses the Digital Twin of the heliostat model as the only aggregate of the DT environment,

it is by no means limited to it. Assuming a problem description which is almost everywhere differentiable, any downstream processes, such as receiver or thermal storage simulations, can be directly integrated into the environment. If the problem description can not be differentiable, these processes can still be integrated indirectly through their own terms in the objective functions.

3.1.6. Comparision to State of the Art

Particularly within the domain of solar tower power plants, numerous ray tracers exist, each serving its specific niche. Examples include Soltrace[18]Raytrace3D[22], Tonatiuh[112] and others[113, 114]. The differentiable ray tracer shares the closest resemblance to STRAL[86], which was also developed in the author's research group.
The basic functions of STRAL have been taken and extended by a differentiable formulation, which were inspired by the differentiable ray tracer REDNER[27]. By adopting the fundamental functions from STRAL, a ray tracer specialized for tower power plants, the differentiable ray tracer can compute results more rapidly than a general scene ray tracer would allow.

In the context of a solar towers, employing gradient-based optimization inside a ray tracing environment is a novel approach. However, when considering the forward direction, the speed and quality of predictions can be likened to those achieved by existing ray tracers. However, a comprehensive analysis falls outside the scope of this thesis.

In the following section, the results of the forward direction of the differentiable ray tracer are compared to those of STRAL. On the one hand, the differentiable ray tracer should produce the same images as STRAL under the same conditions. On the other, this should be archived in a comparable time. This can be treated as a sanity check. For a in depth comparison to other ray tracers then existing literature can be used with STRAL as a reference point[113, 114].

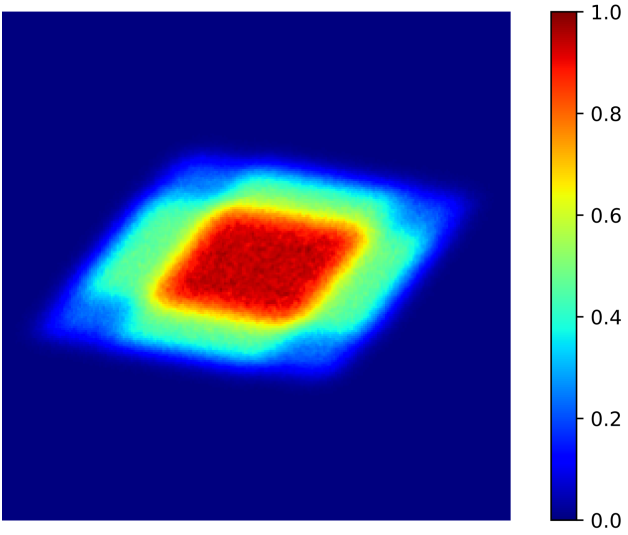
In all ray tracing environments within solar tower applications, the process involves reflecting light at the heliostats and determining the position on the receiver or camera plane. Subsequently, the ray is mapped to a specific pixel in the resulting image on the receiver using a binning function. The primary contrast in the forward direction between the differentiable ray tracer and STRAL (as well as other ray tracers designed for solar power plants) lies in this particular function. Its significance becomes even more pronounced when considering the ray tracer's speed, given that this function is the most frequently invoked one. In STRAL, binning works via an integer operation (cf. Alg. 1), which is mathematically identical to the floor function.

$$E\left(\vec{x}_{ij}\right) = \lfloor E(\vec{x}) \rfloor \tag{38}$$

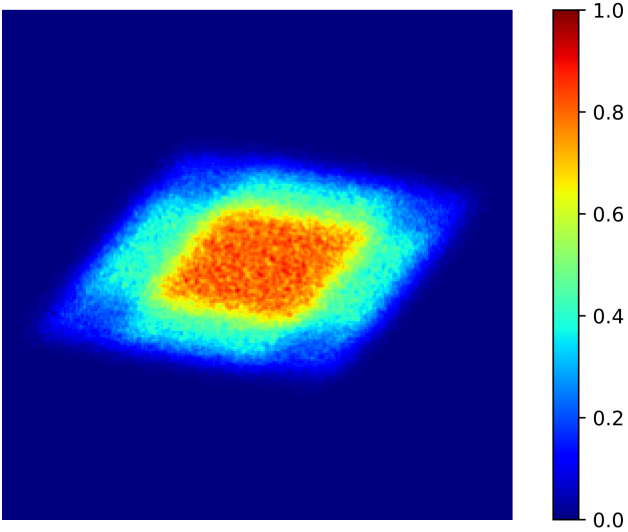
In principle, this is a differentiable function, but its derivative always evaluates to zero, thereby rendering its gradients unusable. The binning introduced in Section 3.1.2

brings another advantage besides differentiability. The assignment of the rays to the bitmap (for creating the image) is one of the most frequently called functions within the ray tracers. Although the binning in STRAL is less computationally expensive than the new binning function, the linear distribution of the incoming rays has an exciting side effect. The new binning works as (simple) anti-aliasing in computer graphics [115]. Thus, the image is rendered with a smaller granulation effect using fewer rays. This effect is shown by the single heliostat flux density images in Fig. 19. For better comparability, both images are generated using the diff. ray tracer but with different binning. Fig. 19b converges to Fig. 19a by adding more rays (approx. 3-5 times, but this was not investigated further quantitatively). On the one hand, this speeds up the calculation and thus more than compensates the more computationally expensive binning. On the other hand, the gradient based methods benefit from the gained smoothness.

Furthermore, the new environment receives a small additional speed benefit from parallelizing the rays (compare Alg. 2) instead of the heliostats (as in STRAL). Thus the diff. ray tracer has an advantage e.g. considering single or few heliostats. However, the advantage decreases when considering larger fields, but never becomes a disadvantage.
Due to the different parallelization, it is very difficult (and outside the scope of this work) to make a fair speed com-



(a) New differentiable binning



(b) Old binning using the floor function as used in STRAL

Figure 19: Relative flux density maps of a single ideal heliostat with canting calculated inside the diff. ray tracer using different binning functions.

Table 1: Speed Comparison of Diff. Ray Tracing and STRAL

Generated rays	Runtime STRAL [ms]	Runtime diff. tay tracing [ms]
4,096,000	652	41
40,096,000	6,342	449

parison. Nevertheless, a first estimation shall be made. For the comparison, care was taken to create conditions that are as similar as possible in the different programs, but this is not completely possible. The parameters used can be found in the appendix Chapter C. The version of STRAL 2.0. used for this test is not runnable on a graphics card, so the test was done on a desktop CPU (Intel i7 4790K). Only 1 heliostat was simulated 100 times from which the mean was calculated. The results can be found in table 1. The times per generated ray are significantly larger than those given by *Bellhomme*[86], which can probably be attributed to the ineffective parallelization in STRAL using only one heliostat. Nevertheless, more than 10 times better run times could be achieved. But as mentioned earlier, the runtime should converge with larger fields. Although, certain speed advantages can still be expected due to the highly optimized algebra in Pytorch and the reduced number of rays required throughout the new binning function.

Optimized NURBS by means of autodifferentiation are not known to the author from the literature. Neverthe-

less, they have already been used with a differentiable representation. For example by *Prasad et al.*[109]. They could even explicitly derive the analytic derivatives of the NURBS functions, allowing e.g. a direct integration of the NURBS as an NN layer. However, since the code was not yet published at the time of this work, a custom implementation of diff. NURBS was developed. Since NURBS are exclusively composed of fully differentiable polynomials, they are thus themselves differentiable and therefore an explicit representation of the differentials as shown by *Prasad et al* can be omitted, by means of automatic differentiation. It is possible that forming the explicit derivatives has a speed or memory advantage over the implementation with auto-differentiation. However, no information for a comparison could be taken from the publication.

A special comparison should be made between the Stone method[38], the standard calibration by means of a geometry model and diff. ray tracing applied on this task. The evaluation of the quality of a prediction is done within the ray tracer via the loss function, usually the pixel-wise Mean Absolute Error also called L1 (MAE) or Mean Squared Error also called L2 (MSE) is calculated here (which loss term is used depends on the task and will be discussed in Section 4.1.3). However, other loss terms motivated by the physics of the problem can (and will) be used as well. Thus, the angle between n_{is} and n_{model} as

in equation 1, for recap

$$F = \min_{\alpha,\beta,\gamma,\delta,\theta_k,\tau_k,\mathrm{GR1},\mathrm{GR2}} \sum_{i=1}^{N} \arccos(\vec{n}_{\mathrm{is},i} \cdot \vec{n}_{\mathrm{model},i})$$

can also be used as a loss function. Assuming that the geometry model within the ray tracing environment and the classic approach are identical, the same dependencies exist for n_{is} and n_{model}. Now the special case will be considered that not a single light ray is generated during ray tracing. Furthermore, the previously mentioned loss term will be the only evaluation function. In this case the optimization of the alignment is done via the current orientation of the heliostat and its difference to the measured normal vector determined from the focal spot. Within this example the only difference between the standard calibration and the diff. ray tracer is the calculation of the gradients and the update routine. Using backpropagation only the Jacobian matrix is calculated, while the LM algorithm also forms the Hessian matrix. The backpropagation can be modified accordingly[116][117][118]. adapting this, the two algorithms would be identical, except for the calculating methods of the derivatives. While the diff. ray tracer uses automatic differentiation, the standard method uses numerical derivatives. considering infinitely small step sizes, the truncation error of the numerical variant disappears and both algorithms are identical.

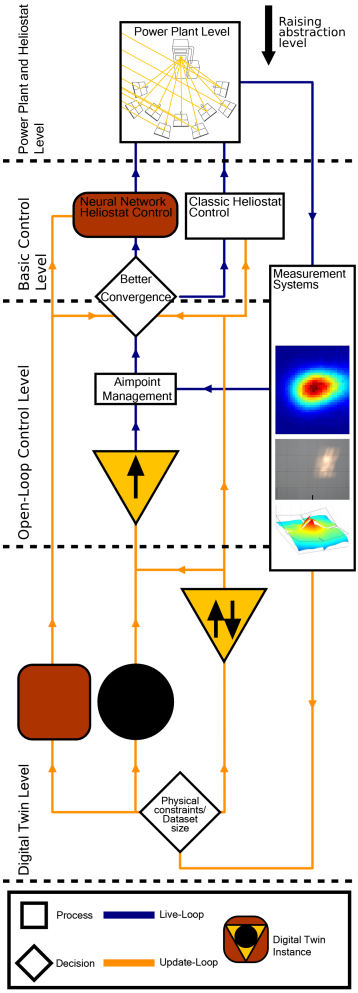
From this absolute minimal example becomes evident what

information gain the calibration can draw from the use of the diff. ray tracer.
Different loss terms can work together here or even replace each other in the course of training. For example, the loss from equation 1 can be a preliminary alignment in the first training iterations (epochs), which is then replaced gradually by the L1 or L2 loss. This way the error in the centroid determination can be eliminated and information about edges, corners, and rotations of the focal spot can be included in the calculation. Additionally, if the heliostat's facets are mounted with a rotational distortion around the heliostat normal, the image loss can also detect this. This would be impossible using only the distance as the evaluation function, as will be demonstrated in Section 5.2.

There are some other possibilities for ray tracing and other algorithms used at the solar tower in gernal to be made more efficient, which have been noticed during the analysis of the existing methods but could not be investigated any further due to the scope of this work. An overview is given in the appendix Chapter B.

3.2. Neural Network Instances for Solar Tower Powerplants

Differentiable ray tracing is a white-box model that provides results directly from the initial measurement. However, the solution space can be constrained by physical models, leading to potentially inaccurate results. In contrast, neural networks can handle more complex and unconstrained relationships between input and output, albeit requiring a larger amount of training data. In the following, two neural networks are introduced. The first architecture is based on a MLP represented by a red square, which has the capability to take over the entire control function of the heliostats once it successfully converges. It uses the same tabular data as



the classic gradient-based optimization method. The second architecture is called NeRF, represented by a black circle, which enables the determination of heliostat-specific focal spots without using ray tracing.
It should be noted that the training data used for the neural networks is limited to a few hundred data points per heliostat in order to ensure applicability at solar towers. This limited data set poses a significant challenge, and various techniques are required to enable neural networks to work effectively with such a small amount of data.

3.2.1. Pretrained Multi Layer Perceptrons for Heliostat Calibration

Neural networks for heliostat calibration bring several advantages compared to the already discussed white box models. A network that can work with the (little) existing data sets makes it possible to completely dispense with pre-modeling of the geometry model. Due to the non-linearity of these algorithms, more complex errors, which would be very complicated to describe analytically, can also be mapped. After retraining with sufficient data, such an already working network structure should directly be able to be used at other power plants and thus speeds up an overarching development. However, this also means that it is no longer possible to exclude individual parameters from the analysis, for example through field studies, and thus reduce the amount of data required.
Therefore, a network is needed that gets by with as little data as possible. In addition, the calibration images can-

not be used directly as input for the network, since they are no longer available after the training process. Many more complex network structures and standard AI methods, such as CNNs, are therefore not an option. This means, the network must be able to work with the same tabular data as the classical heliostat calibration. A data type with which neural networks could achieve only few successes until today.
The only really successful example known to the author from the literature using tabular data on really small data sets is by *Feng et al.*[42]. Using transfer learning [119],[120] and SAE pretraining, *Feng et al.* were able to analyze highly correlated material components with fewer than 500 data points. There are mainly two ideas behind this: First, the data set is really small, so too many neurons will most likley overfit the problem. Second, instead of assigning random values to each node at the beginning of the training, the network is initialized with the node values calculated in the pretraining, which are already close to the global optimum. This is the starting point of this work.

Based on the network structure designed by *Feng et al.* the networks for heliostat calibration were also kept as small as possible. During this work various networks up to a maximum of 50 nodes were tested. The network structure with the best results presented later is shown in figure 20. All networks used the sun and the focal spot position as their input, and either the azimuth or eleva-

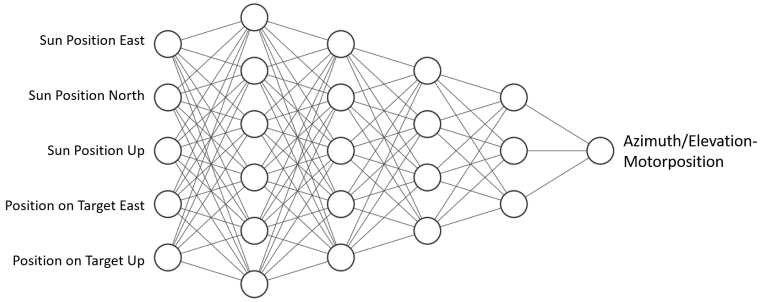


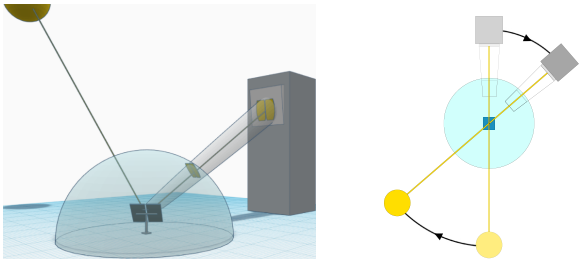
Figure 20: Schematic drawing of the MLP used in this work for heliostat calibration. The network determines the motor position of either the first or second movement axis of the heliostat from the given sun position and the measured focal spot offset on the calibration target.

tion motor position as the output. Because the network is always trained on one heliostat, it's position can be omitted. Due to recent publications [121] which achieved better results with larger input vectors, it was decided to use the vector representation (3 nodes) of the sun as input instead of its Euler representation (2 nodes). However, the results did not show any significant differences. The measured position of the focal spot on the target is additionally given to the network. This can be used for aim point control after training. Together these are 5 input nodes and one output node.

In addition to the unsupervised pretraining used by *Feng et al.* 2.9, also supervised pretraining is discussed in this work. In this case the network is pretrained with artificial data from simulations. Two different pretraining data sets are investigated. One is based on an ideal GM without any errors. The other is data driven and uses the same function template but already fitted with different amounts of data from the solar tower Jülich. Both types of supervised pretraining are discussed in more detail in Sections 4.2.4 and 4.2.5, referring to the data set recorded for evaluation and training.

3.2.2. NeRF for Flux Density Prediction

NeRF networks represent the data-driven counterpart in this work for heliostat surface reconstruction or flux density prediction using differentiable ray tracing. The goal here is to increase the degrees of freedom and to improve



(a) 3D view of the solar tower NeRF setting. The cal. target is projected onto a virtual sphere.

(b) Top view of the environment after the coord. transformation to a resting heliostat and a moving tower.

Figure 21: 2D and 3D representation of the NeRF solar tower setting.

the flux density prediction when the amount of data is so huge, that diff. ray tracing does not take any further benefits from it.

The neural network in NeRF is used to scan a virtual volume and provide feedback on whether the respective voxel in this volume section is occupied by the heliostat or not. A heliostat surface can then be reconstructed from the individual voxels.

In this work the original publication by *Mildenhall et al.*[55] is implemented for use at solar towers. The network remains largely untouched, but the input data must be slightly adapted. However, at least for this original release, not all conditions can be met. Which conditions these are and which adaptations are needed for the imple-

mentation will be discussed in the following.

NeRF expects photos of a non-moving object which is photographed from different angles under constant illumination.
For the solar tower setting the calibration target represents the camera, which obviously does not move around the examined heliostat (compare Fig 21a).
For this to work, the coordinates of the environment have to be transformed into a resting heliostat and a moving target coordinate system. Using these coordinates the cal. target corresponds to the camera and its viewing angle is therefore passed to the network. The position of the tower corresponds to the transformed sun position mirrored at the heliostat axis. Lastly, NeRF still requires a defined volume. For this, the entire calibration target is projected onto a sphere.
All transformations described so far are reversible. However, with the used implementation of NeRF, two conditions remain violated.

First, because the sun is both the light source and (after some transformations) the viewing direction, the illumination of the heliostat changes, which violates the condition of constant illumination.
Second, and probably the most important difference, no direct photos are taken of the examined object, but only the (blurred) focal spots are observed. The physics in NeRF, namely ray pathing, are not able to assign the ray

expansion accurately. A reconstruction of the surface is not possible with the physics inside of NeRF. Ray tracing would be required for this. In the used implementation of NeRF, it is only able to create likewise blurred, not really existing objects. Nevertheless, these replicas behave and act like the heliostats focal spot accordingly to the corresponding sun position.

It should be mentioned that the work on NeRFs in the course of this thesis was started shortly after the first NeRF code releases appeared [55]. Many of the violated boundary conditions, such as the constant illumination [97], have already been corrected by subsequent publications. Also the coordinate transformations can nowadays be neglected and taken over by the network itself [98]. However, these were not available at the time of work on NeRF for solar tower application. This has a corresponding negative impact on the results shown later and they must therefore be evaluated accordingly with the follow up publications in mind.

NeRF has emerged for use on the solar tower as a preliminary to the differentiable ray tracing approach. The lack of this crucial algorithmic backbone inside NeRF (among other reasons discussed in Section 5.8) has led to the development of an algorithm that correctly represents the physics of solar towers. Differentiable ray tracing can therefore be seen as a consequent successor to NeRF. The NeRF results presented later must also be assessed in this

context.

Nevertheless, also the original publication of NeRF used in this work is unique. Even though it has its downsides for the use at solar towers, it remains one of the best networks for new view synthesis using only 30 to 100 (and in follow up publications significantly less) data points per training, and it has achieved astonishing image quality. Even with a reduced image quality due to the violated conditions at the tower, the achieved results could still be good enough for use at the solar tower. How well NeRF performs on solar towers will be presented in Section 5.7.

4. Data Set Analysis and Pipelines

Working with AI always means working with data. It is important to understand which and how much data is available, how it is (pre-)processed and how the models interact with the data, for example which evaluation functions are used. For the work with the models described above, both simulative data and data measured at the solar tower in Jülich are available. The different data sets, their acquisition, and how the AI routines interact with this data will be discussed in the following sections. First, the data sets and workflows of the diff. ray tracer are pre-

sented in Section 4.1. Here the data sets used for the different tasks are presented (Section 4.1.1-4.1.2). Then the workflow is described, how the data is processed within the ray tracing environment (Section 4.1.3). Afterwards, the neural networks data sets and then their pipelines are presented (Section 4.2). As before, first for NeRF (Section 4.2.1-4.2.2), then for the MLP for heliostat calibration. Here, a distinction is made between the training (Section 4.2.3) and pretraining (Section 4.2.4) data set. The chapter is closed with the MLP data pipeline (Section 4.2.5) and general recommendations for the use of pretrained MLPs at solar towers (Section 4.2.6).

4.1. Data Set Acquisition and Pipelines for Differentiable Ray Tracing

To evaluate the differentiable ray tracer, simulation data enhanced with measurement data is used. For this purpose, a heliostat was measured deflectometrically at the solar tower in Jülich. The measured surface is shown in Figure 22. The peak in the center was artificially added via set screws to have a prominent feature for later reconstruction. However, the feature is within the range of common surface deformations[122]. Deflectometry measurement provides discrete points and corresponding normal vectors. The displayed surface was calculated from these using geometric relations. Later, when it comes to the reconstruction of surfaces, only the normal vectors will be compared to each other to reduce the mathematical error.

In order to evaluate the diff. Raytracer, a heliostat in 44.2 m distance to the tower with 4 individual focused facets was deflectometrically measured at the solar tower in Jülich. With this surface on the one hand the accuracy of the reconstruction can be verified, on the other hand focal spot images can be generated in any quantity and under conditions, which would not be possible at the solar tower, e.g. images outside the focal plane of the heliostat. To increase the validity of the simulated data set, an additional calibration was performed on a cloudless day and compared with the ray tracing data. The comparison is

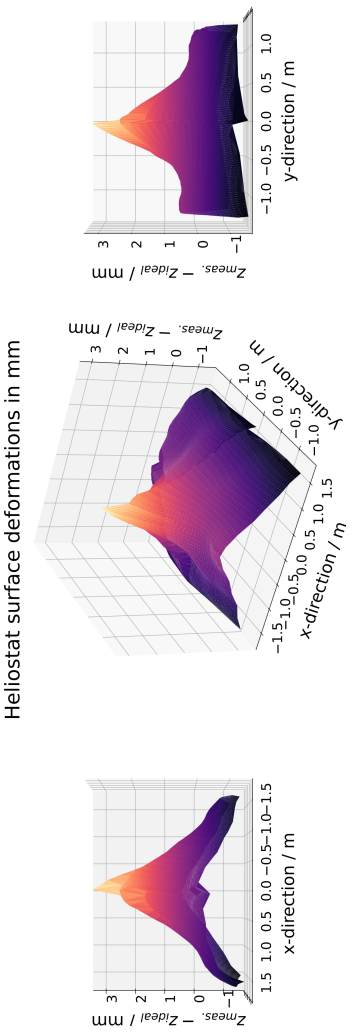


Figure 22: The surface deformation of the measured heliostat in Jülich. The deviations were calculated from the deflectometrically measured normal vectors. The peak in the middle was artificially added via set screws to get a concise feature.

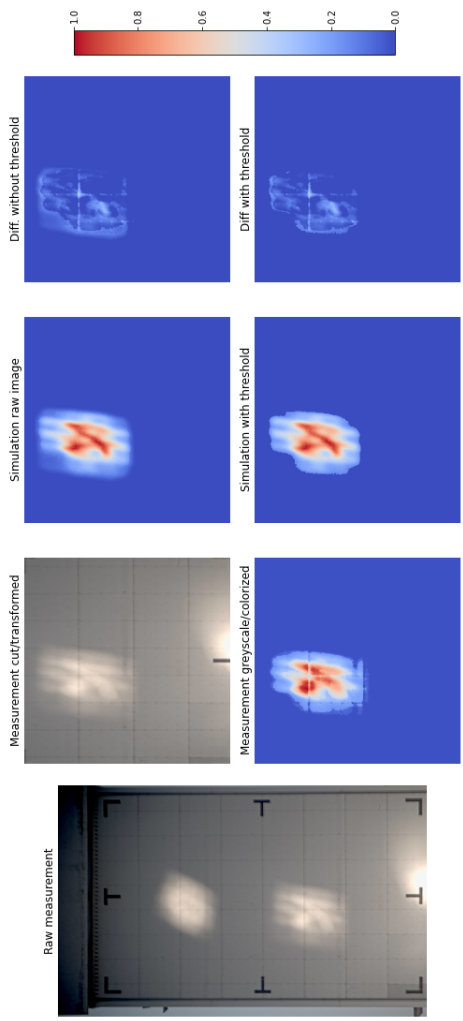


Figure 23: Callibration measurement at the solar tower in Jülich compared with a simulation supported by deflectometry data. The focal spot at the bottom of the image included in the measurement was removed for the comparison by hand.

shown in Fig. 23, the simulation supported with deflectometry has extremely high correspondence with the real measurement, both qualitatively (Fig. 23) and quantitatively (Tab. 2). This agreement can even be increased by a simple threshold operation.
The largest deviations result from the measurement setup, which contains square plates which are not simulated. As

	With threshold	No threshold
Hausdorff-Distance	30cm	33.5cm
Cross-Correlation	98.3%	97.2%
Mean Absolute Error	7.5e-5	7.65e-5

Table 2: Different evaluation measures to check the similarity of simulation and measurement

usual with the heliostat calibration, the image shown here is not centered. The position and rotation of the heliostat in the simulation was adjusted fully automated by the differentiable ray tracer by optimizing the heliostat's alignment parameters depending on the measurement image, which will be discussed in more detail in Section 5.2.

As will be presented in later results, reconstruction of the heliostat surface, as well as the focal spot shape in exact focal length, is especially challenging. For this, in addition to the physical regularization, a data-driven one is needed. An appropriate large data set (>15 images) fulfilling the clear sky conditions, as well as not being over- or underexposed could not be recorded due to reconstruction work

in the field and weather conditions (as well as other interfering factors like other focal spots in the image). That is why in the later training runs only the raytracing images generated with deflectometry data are used.

In the following, the different data sets, created with deflectometric enhanced ray tracing will be discussed.

4.1.1. Data Sets for Heliostat Surface Diagnosis and Flux Density Prediction

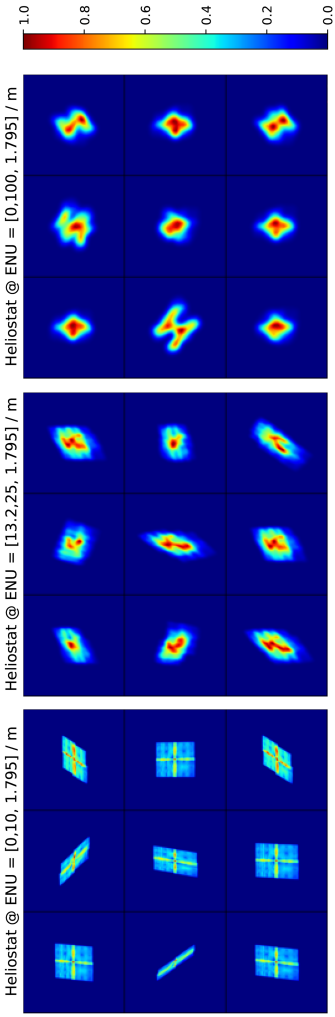


Figure 24: Focal spots of the same heliostat in different distances.

The use of ray tracing data has several advantages. On the one hand, different parameters can be varied within the virtual environment. Thus, e.g., the distance- and focal length-dependent resolving power can be determined. On the other hand, it allows to make forecasts for the whole year, for which very complex measurement campaigns would be necessary. For the reconstruction of heliostat surfaces, 1-50 focal spots of randomly drawn solar positions are used. Figure 24 shows 9 of each of these focal spots at different distances (but same focal length). All focal spots used for surface reconstruction are centered.

4.1.2. Data Sets for Heliostat Calibration

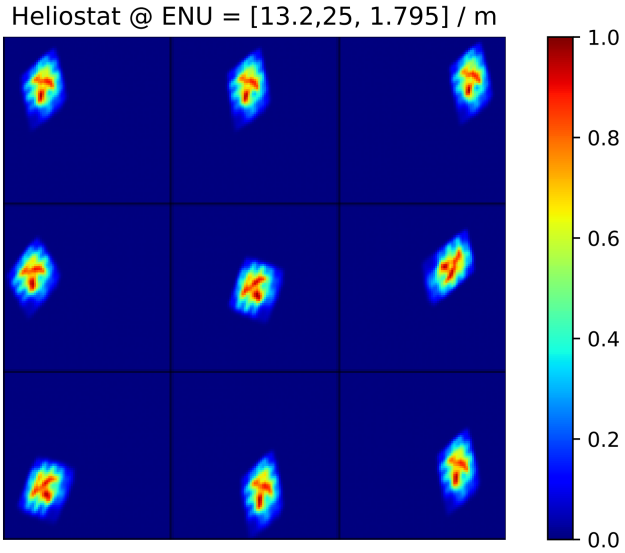


Figure 25: Focal spots of the same heliostat as in the previous section but augmented with different alignment errors under the same sun inclination. The errors are randomly generated but for the image were selected to cover all the positionings on the cal. target.

For heliostat calibration, a larger image section is selected. In this case, the target is 10 m x10 m in height and width. The geometry model in the diff. ray tracer, is in its current state of development, simpler than the one used in Jülich. The heliostat has no offset at the rotation axes and thus aligns the heliostat at its origin (the top of the pylon or $\vec{O}_1$ compare Fig. 5).

The error model is thus reduced to three errors. α, β which are identical to the geometry model in Jülich and which tilt the orientation of the heliostat to the horizontally or vertically (compare Fig. 4), and the ξ error which is not considered respectively can not be considered by the geometry model in Jülich. ξ describes a rotation of the mirror facets around the normal vector of the heliostat ($\vec{n}$ in Fig. 13). Since this rotation does not result in a translation of the target image, it cannot be detected by the existing distance determination of the regression used in Jülich. Such an error can occur if the two alignment motors of the heliostat are not attached to the center and rotate the heliostat (also intentionally) around this axis. This is e.g. the case with the newly retrofitted heliostats at the solar tower in Jülich.

Figure 25 shows images with various errors that shift the target image over the entire calibration target. The center image contains only the rotation error ξ. For the test later α, β, ξ are chosen to be between 0.1 mrad and 15 mrad.

4.1.3. Data Pipeline to Predict Heliostat Properties by Differentiable Ray Tracing

To predict the different heliostat properties, the calibration images are given to the diff. ray tracing environment which updates the corresponding heliostat properties accordingly. In principle, the main workflow was already illustrated in Fig. 12. However, since the procedure differs slightly depending on the task, it will be discussed in

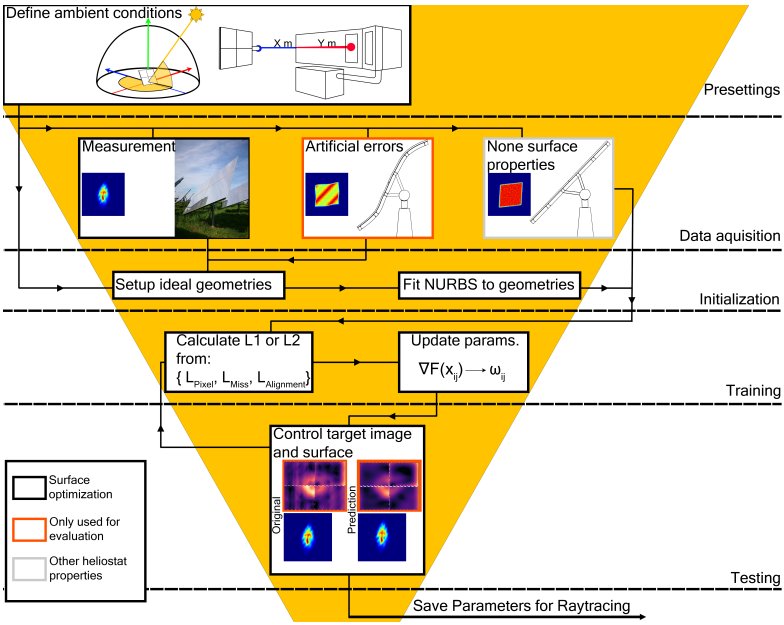


Figure 26: Pipeline to determine heliostat properties from calibration images. In principle, drone photos, flux density or calibration measurements can also be used here. The loss terms shown are explained in the text.

more detail and with focus on the existing program sections here.

The detailed workflow for diff. ray tracing is shown in Figure 26.
In the *Presettings* the environmental conditions used for the successive simulations are defined. For example, the position of the solar tower and the sun, the orientation of the heliostat, or which of these variables should be optimized in the following.
The *Data acquisition* corresponds to a forward ray tracing step, which is separated from the diff. ray tracing loop. Here the *target* data set including training and testing images are created or loaded. Within this work the three already mentioned data sets are considered.
A distinction is made here between two cases. If surfaces are to be reconstructed, an ideal image of the heliostat is first created within the ray tracing environment during the *initialization* step. The NURBS are then adapted to this ideal mirror surface. It needs a NURBS surface for each facet (resp. for each discontinuity of the mirror). The ideal mirror surface may also be arbitrarily curved.
Otherwise, if no surfaces are to be reconstructed, e.g. for calibration or for the determination of canting errors, the NURBS are not needed and the heliostat is defined by discrete points and normal vectors either by an ideal representation or by a preloaded surface.
In *training*, the actual loop begins as already shown in 12. The initially ideal heliostat generates images at the same

sun positions as they occur in the target data set. The images are then compared to eachother.
In general this happens using the L1 (MAE),

$$L1(x, y) = \Delta E = \sum_{n=1}^{N} |x_n - y_n| \tag{39}$$

or L2 (MSE) loss

$$L2(x, y) = \Delta E^2 = \sum_{n=1}^{N} |x_n - y_n|^2 \tag{40}$$

where x_n is the prediction and y_n the target value. However, which values are to be predicted depends on the task.

The heliostat calibration uses L2 because it provides non-sparse, unambiguous solutions and penalizes outliners more strongly than L1 does[123]. For this task three different loss terms are used:

$$\begin{aligned} L_{total} =& a \cdot (L2_{pixel} + WD_{pixel}) + & (41)\\ & b \cdot (L2_{alignment} + WD_{alignment}) + & (42)\\ & c \cdot (L2_{miss}) & (43) \end{aligned}$$

with $L2_{pixel}$ the pixel wise image loss and

$$L2_{alignment} = L2(\vec{n}_{pred} - \vec{n}_{target}) \tag{44}$$

whereby $\vec{n}_{pred}$ is the current alignment vector of the heliostat and $\vec{n}_{target}$ is the normed vector between the target image's centroid of area and the heliostat position. $L2_{miss}$ measures the distance between those rays which missed the target and the cal. targets edge. This term is zero if all rays are hitting the cal. target. WD are the corresponding *weight decay* terms a, b, c are scaling factors. Unless otherwise specified, they have the value:

$$a = 1 \cdot 10^{-5} \tag{45}$$

$$b = 1 \tag{46}$$

$$c = 1 \cdot 10^{5} \tag{47}$$

For the surface reconstruction L1 is used, because it provides a sparse (but ambiguous) solution. For one reason this is due to sun blur and ray broadening the surface reconstruction is in an underdetermined regime. It is therefore desirable to have a function that allows multiple alternatives. The other reason and most important property of L1 is the sparsity[123]. The NURBS were initialized so that their original shape is identical to that of an ideal heliostat. The zero position of the control points is within the surface. Instead of displacing all nodes while training, the L1 loss should converge to a sparse matrix. With excitations only where the heliostat deviates from the ideal surface. The loss for surface reconstruction is composed as follows:

$$L_{total} = a \cdot (L1_{pixel} + WD_{pixel}) + c \cdot (L2_{miss}) \tag{48}$$

Considering the $L1_{pixel}$ loss alone, the difference between the pixels, can be regarded as a power difference ΔE and thus as a physical quantity. The loss for every evaluation and test data sets is therefore the $L1_{pixel}$-loss without any additional terms.

Every independent loss term has additionally to the constant scaling factors a, b, c also an adaptive learning rate. Unless otherwise specified, the values are:

$$LR_{pixel} = 5 \cdot 10^{-5} \tag{49}$$

$$LR_{alignment} = 1 \cdot 10^{-2} \tag{50}$$

$$LR_{miss} = 1 \cdot 10^{-2} \tag{51}$$

the learning rate decreases exponentially with a decay factor of 0.9995 for surface reconstruction and 0.995 for the heliostat calibration. *Adam* is used as the optimizer using the standard values as preprogrammed by *Pytorch*.
Depending on the available amount of data, 6x6 (for 1 image) to 12x12 (15-50 images) NURBS control points per facet are used.

Depending on the selected loss, the desired parameters are updated using *backpropagation*.
The last step in the workflow is the *testing phase*. Here, images are generated and compared with a test data set that is different from the training data set. In addition, the generated surfaces or alignment parameters are compared with the predicted ones. This comparison serves

the evaluation within this work and is not available in daily operation. If the (image) test loss stagnates over multiple epochs, the training is stopped and the updated parameters can be used to update the DT Environment or any other ray tracer/CAD program.

The entire routine takes about 5-10 minutes on an *NVIDIA 2080TI* for high image resolutions, one heliostat, and a data set of one image using about 800k rays per iteration.

4.2. Data Set Acquisition and Pipelines for Determining Heliostat Properties with Neural Networks

The two neural networks used in this work, NeRF and pretrained MLPs, basically use the same data as the differentiable ray tracing, namely the images of the heliostat calibration. However, they need much more data in order to function properly. In addition, artificially generated data and real measurements work together this time.

4.2.1. Data Sets for Neural Radiance Fields

NeRF tries to infer the target image directly from the (previously transformed) sun positions. However, unlike differentiable ray tracing, NeRF requires significantly more data. The original publication used between 25 and 100 images. For the first tests at the solar tower, two different training data sets with 100 and 200 images each were created. On top of that, there are 2 test data sets with 100 additional images each. All images were created using the ray tracing software STRAL.

The first new data set will be called *Sanity Check* and uses a simple square heliostat with one facet and is placed 100 m away from the target. The sun has a fixed elevation angle of 10° and an azimuth that is scanned linear in 200 steps over 180°. The sun profile is a Gaussian. The corresponding test data set has a slightly tilted elevation (2°) and the azimuth steps are in between the training

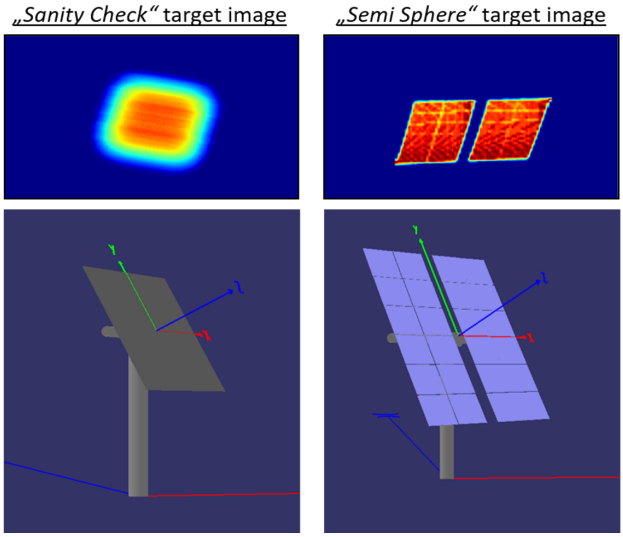


Figure 27: Representation of the heliostats used and corresponding target images of the two data sets presented. The intensity of the displayed images is normalized to 1.

set steps. As the name of the data set suggests, it is used to check the basic functionality of the network. The steps between the individual images are very small and the extrapolation in the test data set is very low. Also the images do not contain very dense details (compare Fig. 27 on the left).

The second data set will be called *Semi Sphere*. The heliostat consists of 24 small facets and on the same position as the heliostat in the *Sanity Check* data set. In contrast this data set is created by simulating with a point sun (avoiding sun blur), resulting in a lot of sharp details in the image (compare Fig. 27 on the right). The amount of data is also reduced to 100 images, which were randomly drawn from the entire semi sphere above the heliostat. A summary of the data sets used is given in Table 3. Unlike differentiable ray tracing, NeRF is not a physically motivated representation of the system (even though supported by physics). While the ray expansion in diff. ray tracing causes an underdetermination and the reconstruction therefore becomes more difficult, neural networks tend to have problems with higher detail depths. If NeRF is able to represent the facet details, it should also be able to represent real errors. Thus, especially from the *Semi Sphere* data set, conclusions can be drawn about the behavior with real data.

Data set Name	Elevation [°]	Azimuth [°]	Data set size	Distr.
Sanity Check (Train)	10	0-180	200	linear
Sanity Check (Test)	12	0-180	100	linear
Semi Sphere (Train)	0-90	0-180	100	random
Semi Sphere (Test)	0-90	0-180	100	random

Table 3: Training and test sets used for the evaluation of NeRF.

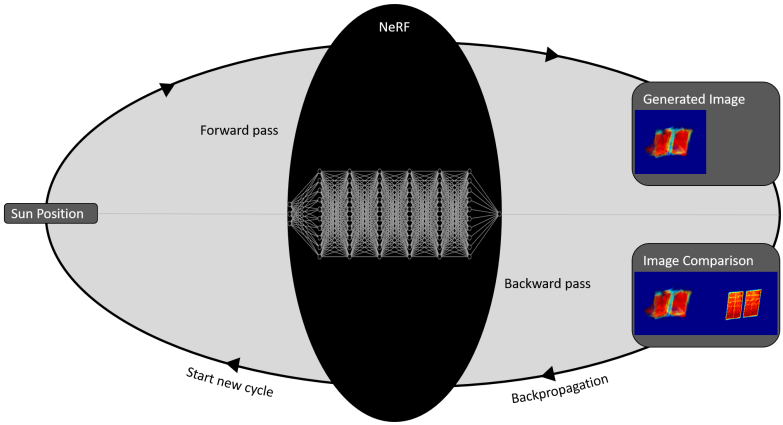


Figure 28: Update Routine of the NeRF Network architecture.

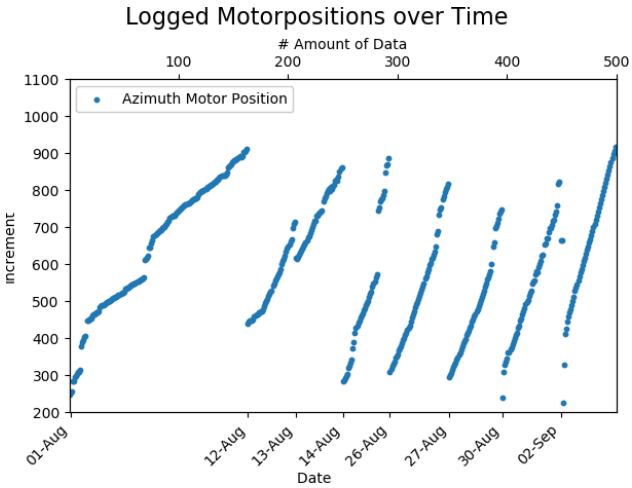
4.2.2. Data Pipeline for Neural Radiance Fields

Basically, the NeRF data pipeline is directly comparable to that of differentiable ray tracing (compare Fig. 28 and 12). From any given sun position an image is calculated. As in the original publication NeRF has no direct image output. As described in Section 2.8, NeRF generates a full 3D volume within the network structure. The images are then rendered from the given viewing direction using this volume. which is compared with a target image. Then the parameters are updated. But there are 2 main differences. First, NeRF can only do this one task. Other heliostat parameters cannot be optimized with it.
And secondly, not the physical environment parameters are adjusted, but the weights and biases within the neural network. For each image, the entire virtual volume

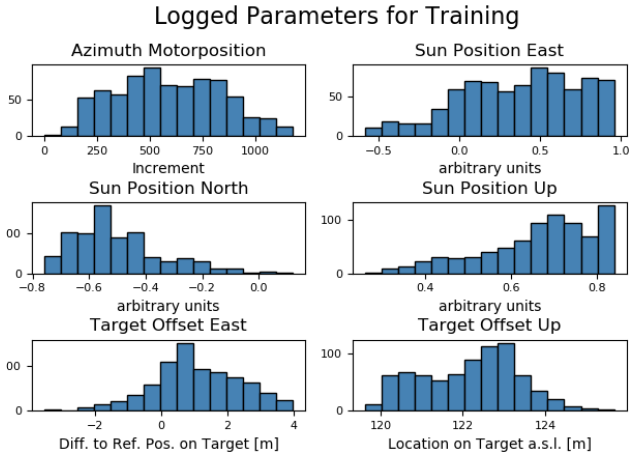
(each voxel) must be rendered by NeRF. This has a significant impact on the runtime. On an NVIDIA 2080TI, one training cycle takes 3-12 hours, which has been a time bottleneck during this work. The image generation after the finished training took several seconds per image, which would be a bottleneck in daily powerplant operation. However, today's implementations of NeRF [102] are capable of training in a few seconds and rendering in real time.

4.2.3. Data Sets to Train Multi Layer Perceptrons

Neural networks require a large amount of data to produce meaningful results, which can be a challenge given the limited data set available for each heliostat. Therefore, the goal is to get the nodes of the network as close as possible to the global optimum solution by using a suitable (large) data set in the pretraining. Afterwards, the network is finetuned with (little) real data. Within this work 3 different pretraining data sets are discussed as well as the training data set. These are presented in the following. The training data set for training and evaluation (as well as the unsupervised pretraining as done by *Feng et al.*[42]) of the network was recorded in Jülich for a heliostat far back of the field. The heliostat position is at 57.2 m east, 243.5 m north and 1.795 m over of the base of the solar tower. A few motor increments can therefore already cause the tower to be missed. It was recorded on 8 individual days between August 1st and September 2nd in 2019, of which a large part of the data falling on the



(a) Logged azimuth motor positions as a function of time for the examined heliostat.



(b) All input and output parameters used to train the network and their distributions.

Figure 29: Different visualizations of the collected data at the solar tower Jülich.

first measurement day. The data was recorded at different times between 9:00 a.m and 3:30 p.m. The amount of data varies between 1 and nearly 200 measurement points per day. No conditions were placed on weather conditions, sun position, or direct normal irradicance (DNI) of the sun, as it would be the case in daily operation.

Unlike the algorithms presented so far, the MLP for heliostat calibration does not work with image data. Image data cannot be used here, because it would not be available after training. Tests with augmented data did not produce useful results. Instead, tabular data is processed, as it is also used for the classical regression at the tower. The data set consists of a total of 500 measurements (Fig. 29a). This includes the *sun position*, the focal spots calculated centroid in *east* and *up* direction (*target offset east* and *target offset up*) as well as the current heliostat alignment given in *motor position* increments at the time of measurement. In a first approximation, the accuracy of the examined heliostat is given by 0.35 mrad per increment. Fig. 29a shows only the azimuth motor position of the heliostat during calibration, which will be, without loss of generality, the main value of the following discussion.

The histogram (Fig. 29b) shows the logged values required for NN calibration. The values *Target Offset Up* and *Target Offset East* are showing the focal spots position on the target. The focal point of this particular heliostat is often located in the center of the target but tends to drift eastward and downwards. It is rarely above

or west of the target center, which will be discussed later in Section 5.4.

Since this is a data set created by measurements, all values are affected by uncertainties. Due to a delay in the transmission signal caused by the technical equipment on the tower, the recorded measurement data is only accurate to one minute. Two measurements within one minute can result in 2 motor positions being logged at the same time. Other known errors that affects the measurement accuracy are that the heliostats sometimes miscount by a few increments. There are also some uncertainties in the calibration process e.g. due to different weather conditions, which affects the focal spot shape. Except for an algorithm that tries to detect systematical dropouts, the data has not been filtered or preprocessed in any way. For the required high accuracy of heliostat alignment, the algorithms used must be able to handle these inaccuracies.

4.2.4. Data Sets to Pretrain Multi Layer Perceptrons

For a supervised pretraining a larger data set similar to the original task is needed. Simulation data is ideal for this purpose. In total, two of these data sets are generated. The data is created by means of the geometry model presented in Section 2.4. For both data sets, 4000 randomly distributed sun positions between 1. August and 1. October in 2019 are simulated. This data is given to the network in a 90:10 training to test ratio using a temporal split.
The first data set uses the geometry model as ideal (all error parameters are 0). The data generated in this way was iteratively adjusted to the first measurement day of the training data set using an offset and a scaling factor. This was done due to of a huge discrepancy between the real and the simulated data. It could also be done with just a few measurement points. In Fig. 30 results of the adapted simulation compared to real measured calibration data is shown. Even if many peaks are not reproduced by this model and the difference to the real data increases continuously, at least a similarity can be observed.
The second data set is created with the same geometry model but errors already fitted via the LM Algorithm. Due to the data dependency of this fit an analysis of it takes place in Section 5.1.

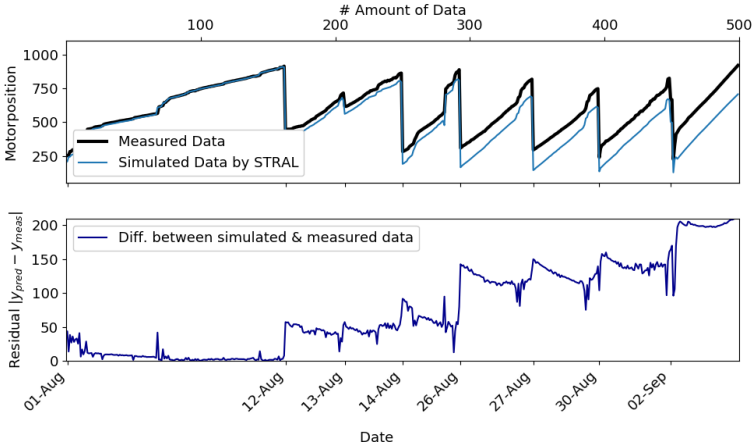


Figure 30: The upper graph shows the measured calibration data (black line) compared to the data simulated by the ray tracer STRAL (light blue line) for the same time stamps. The lower graph shows the same data but relative to each other. While the shape of the curve looks similar to the measured data, there is clearly a linear offset, which grows over time.

4.2.5. Data Pipeline for Pretrained Neural Networks

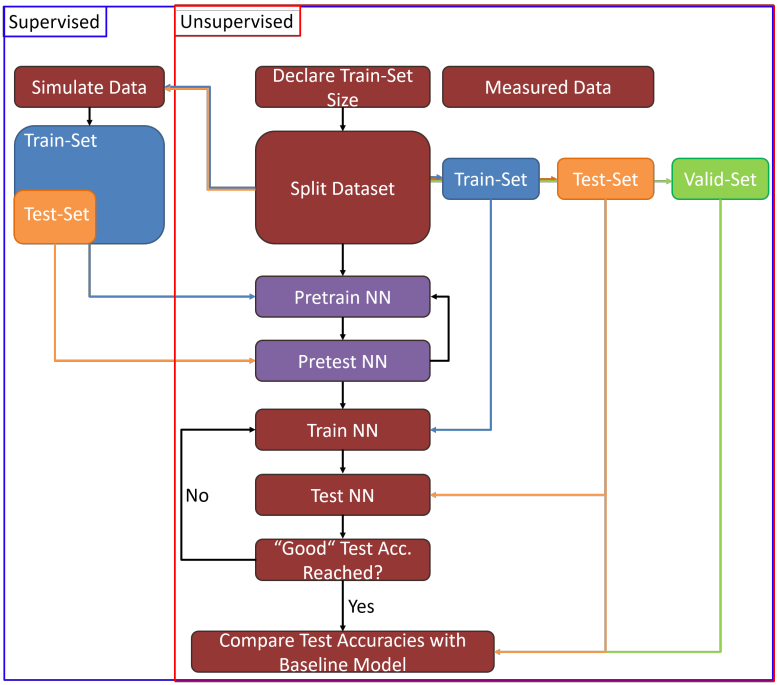


Figure 31: Sketch of the workflow. Depending on the pretraining routine, the workflow is slightly different.

One of the most important factors for using neural networks on the tower is the amount of data required. Therefore, the data set accessible to the network is increased after the end of each training cycle. For data-driven pretraining methods, the same applies. The whole workflow

is shown in Fig. 31. Depending on the kind of pretraining (supervised or unsupervised) a different routine is chosen. In the unsupervised case the same data as for the training process is given to the network using the stacked autoencoder setup as used by *Feng et al.*

For the supervised pretraining exactly the same amount of training data is given to the GM fitted by the LM as at the later neural network training step. This includes the same train/test split. Using this training data set the Levenberg-Marquardt algorithm is used to fit the GM parameters. Only if the test loss is lower than a previous one, the newly calculated parameters are used to generate the pretraining data set. Otherwise the pretraining data set of the best test loss up to this point is used. An example procedure is sketched in Fig. 32. While the training data for the neural network is increased every training cycle the LM train/test split stays the same after the third iteration because the test loss does not improve any further after the third cycle. In this case, all following training cycles the same pretraining data set would be used. The ratio between training and test data set depends on the available amount of data. For smaller data sets, a test/-training ratio of up to 50:50 is used. All remaining data is in a third validation set. When the training set becomes larger than 50% of the available data the test data set is created from all remaining data except for 5% which is reserved for the validation set.

It is important to separate the training and test data set properly. In contrast to the usual procedure the data sets must not be shuffled (between training and test data set). The networks are incredibly good at interpolation. A test data set lying in between the training data points would be fitted very well but would have no expressiveness on future data, respectively its ability to extrapolate. As it was done for the pretraining data set a temporal split is chosen here. The training data set is created from the older measurements, the test data set by the newer ones. Other splits, like by the position of the sun (below and above a given elevation/azimuth), are also possible here. For the later discussed results the network as shown in Fig. 20 is used. For the activation function Scaled Exponential Linear Units (SELUs) is chosen. The nodes of the network are initialized according to the publication of *Klambauer et al.*[95]. SELUs are chosen because they provide better gradients on deep MLP. Also other activation functions were tested, but with much worse results. The output layer is linear, which is helpful for regression tasks. To regularize the prediction error the L1-Huber loss [124] is used. This is a mixture of the L1 and L2 loss, which are selected depending on a threshold. The threshold is set to 1. Thus, values not included in the range of the training data set are penalized more strongly. To enhance the learning process, an adaptive learning rate, which means a factor that controls the amount of change inside the nodes, is selected. The used learning rate becomes smaller every time the test loss does not reduce over

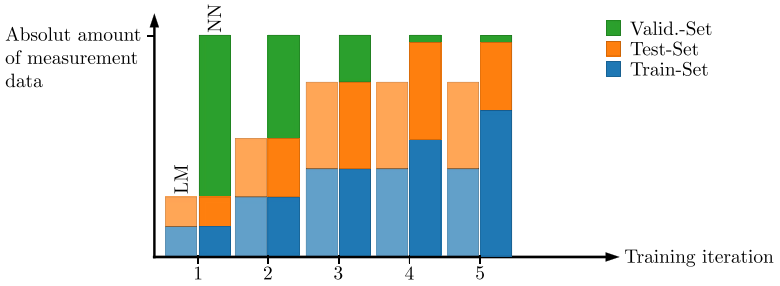


Figure 32: Example of the training cycle routine including the amount of data for the training and pretraining data sets.

a certain amount of *epochs*. The pretraining starts with a learning rate (lr) of 0.1 and is reduced steadily. The learning rate while training is considerably smaller (lr = $1 \cdot 10^{-5}$) in order to preserve the pretraining progress. This is enhanced by using AdaMax [1] as an optimizer.

As the differentiable ray tracer, the whole method was implemented using *Pytorch*. However, an older version as applied in the diff. ray tracer is used, which is why *Numpy* and some minor tweaks from *Scikit-learn* are mandatory too. For loading data the program uses *Pandas* and for plotting *Matplotlib*.
The whole process on the full dataset takes around 15-30 minutes on an Intel i7 4770k processor depending on the data set size. A graphics card is not used.

4.2.6. General Remarks Concerning the Training of Neural Networks for Heliostat Calibration

The network shown in Fig. 20 is of course not the only network structure tested for this work. The network, i.e. the nodes and the number of layers, were varied iteratively. Furthermore different input embeddings, data augmentations, activation functions, layer freezing and more were tested. The results of very similar networks to the depicted one differed only in nuances. Significantly deeper or shallower networks, on the other hand, produced much worse results on the limited data set. These abortive attempts, and the hyper parameter tuning of the networks will not be addressed further in this thesis, as they go beyond the scope of the intended research questions and would shift the focus away from its actual objectives. Nevertheless, in summary, under the described conditions (e.g. SELU, AdaMax) and the corresponding pretrainings as well as the upper end of the available data, a very stable convergence could be observed even with deviating network architectures.

Nevertheless, it must be noted, that the entire training routine, including the training cycles with growing data sets was also performed for the networks not mentioned here. Beacuse of that some compromises had to be made due to time constraints. Especially considering the fact that no suitable graphics card was available to train the

networks at that time. The biggest compromise concerns the pretraining. Because this was the largest bottleneck, it was decided to limit it to 4000 data points in the vicinity of the measurement dates. The author recommends to enlarge this data set considerably and include simulated data points over the whole year for future use at solar towers. In this case, even slightly larger networks should be considered again.

5. Evaluation

After laying out the theoretical foundations, explaining the methods, and collecting the data, the next step is to present the results. In this section, the focus is on presenting the different tasks, namely heliostat calibration, flux density prediction, and surface diagnosis, rather than the different approaches used such as diff. ray tracing and

neural networks.
The first task presented is the heliostat calibration, which is accomplished by means of regression and is applied at the solar tower in Jülich. This is discussed in detail in Section 5.1. Subsequently, the results of the new AI models are analyzed. The diff. ray tracing approach for heliostat calibration is presented on a simple toy example, which demonstrates the potential of this method (Section 5.3). The NN models are also examined, first without pretraining in Section 5.2.1, and then with different pretraining variants in Section 5.2.2. Finally, the different approaches are discussed together in Section 5.4.
The same structure is followed for the surface diagnosis (Section 5.5) and the flux density prediction (Section 5.6) tasks, which are accomplished using diff. ray tracing and NeRF (Section 5.7). These tasks are also discussed together in Section 5.8. Overall, by presenting the different tasks and their respective approaches, this section provides a comprehensive overview of the results achieved in this study.

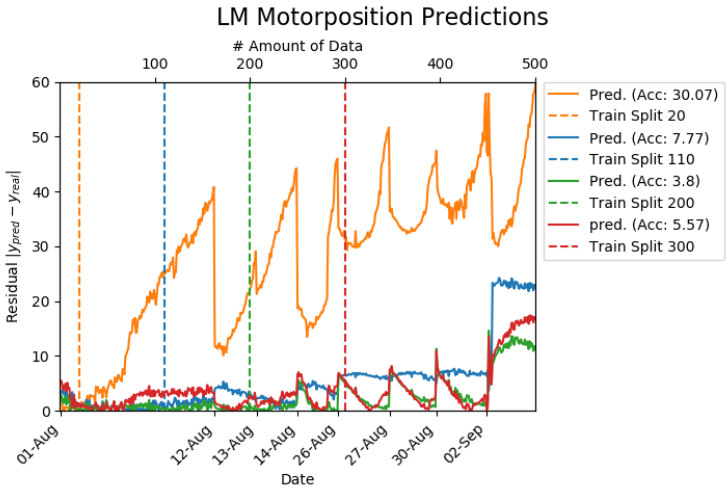


Figure 33: Motorposition predictions of the GM given in increments (y_{pred}), relative to the logged (y_{real}) increments, estimated by the LM algorithm for different amounts of data. The different training-sets are located on the left of the dotted lines.

5.1. Results on Heliostat Calibration Using the Classic Regression Applied in Jülich

This section aims to test the best possible accuracy of the GM used in Jülich, which will be the baseline model for the evaluation of the later used neural networks. For this purpose, all available parameters (compare Section 2.4) are optimized. Also if a restriction of the parameters could achieve better accuracies on smaller data sets, the

global optimum would most likely be missed (unless the corresponding parameter is also 0 in reality). Such a reduced selection will be omitted here.
Fig. 33 shows how the GM in Jülich behaves with different amounts of the recorded data set given to the Levenberg-Marquardt algorithm. The dashed line indicates the training/test split of its color-correlated counterpart. Left of the dashed lines the available training set and on the right the test and validation data set are shown together. For the smallest data set (orange line), it is easy to see that the algorithm is overfitting. Even if the training data set could be fitted very well, the prediction accuracy in the test data set decreases steadily. However, the accuracy improves fast and continuously with an increasing amount of data up to about 200 data points, where it reaches an accuracy of 3.8 increments (ca. 1.5 mrad).
Beyond 200 data points, however, the LM algorithm stagnates at a roughly constant level. The slight drop in accuracy for a higher amount of data could be due to an imbalance of the data set. The peaks appearing in the red and green line as well as the plateau in the rear part could indicate two things. Either only local optimums were found and the peaks are regression artifacts or the results are close to the global optimum but the GM cannot map the real heliostat completely due to missing modeling parameters. The best accuracy achieved by the GM is 1.5 mrad. The neural network must be measured against this in the coming sections.

5.2. Results on Heliostat Calibration with Pretrained Neural Networks

After showing (within a certain convergence band) the absolute accuracy of the geometry model in Jülich, the neural networks will be tested. In order to be able to better classify the results of the later used pre-training variants, first a look at the NN behavior without pre-training will be taken.

5.2.1. Neural Networks Trained from Scratch

For this reason the results of the LM and a randomly initialized neural network will be made. Figure 34 shows the measured data (black) and the respective predictions of the GM (green) as well as the predictions of the randomly initialized neural network. In the left plot the network was supplied with only half a day (morning to noon) of data. Since it has not received any pretraining, it has to extrapolate for the second half of the day and is becoming less accurate. This effect becomes stronger in the evening hours of the following days. Apart from the peaks of more than 75 residuals, there are nevertheless also predictions that are very accurate. However, this is not due to a learned tracking function, but the particularly good predictions are always at times (sun position) which are similar to the training data set.
In the center plot the network already uses 200 measurements for training. Seeing a full day could successfully teach the network to predict the rough course of the helio-

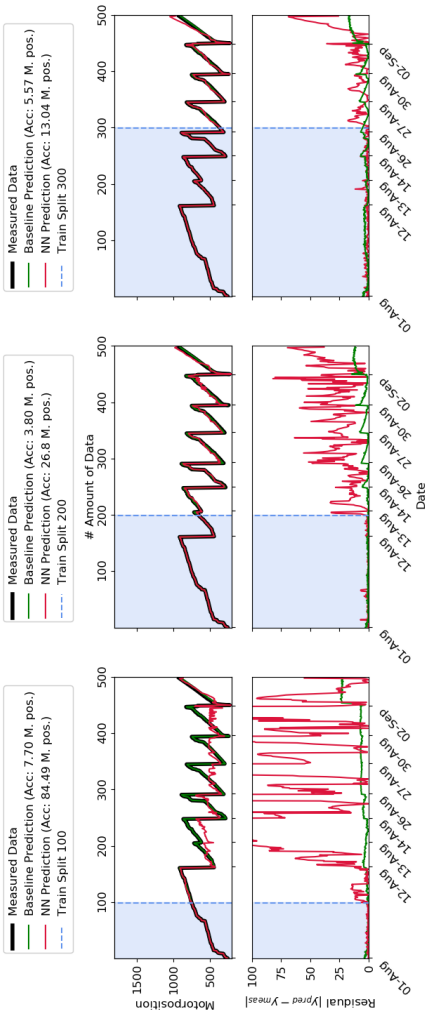


Figure 34: Predictions of the baseline model and the not pre-trained neural network. The accuracy of the neural network rises with the amount of data, while the baseline model stagnates at some point. Nethertheless the NN never reaches a competitive accuracy.

stat's movement (top plot). In this view, the measurement curve and the prediction are almost congruent. However, when viewed more closely (lower plot), it is noticeable that the prediction is still very poor with an average deviation of 26.8 motor positions. Most importantly, the prediction accuracy between the training and test data sets differs significantly. The network is overfitting. It is very likely that the predictions with significantly different sun positions will also be even less accurate.
In the right plot the network has already seen 3 full measurement days. The difference of 100 data points nearly doubles the accuracy. Nevertheless, an acceptable accuracy could not be achieved. In addition, the peak in the validation set indicates that the accuracy will still drop in future predictions. From the development of the prediction in dependency of the amount of training data, it can be extrapolated very well that neuronal networks can achieve good results for much larger data sets even without pretraining. However, this would require new measurement techniques that can produce such a large amount of data. With 300 data points, the current measurement method is already at the upper limit of what is technically feasible in daily operation.

5.2.2. Pretrained Neural Networks

The pretraining aims to render the nodes of the network closer to the global optimum. The process is designed to help improve the accuracy of the network and reduce data usage to a minimum. The measurement data is pro-

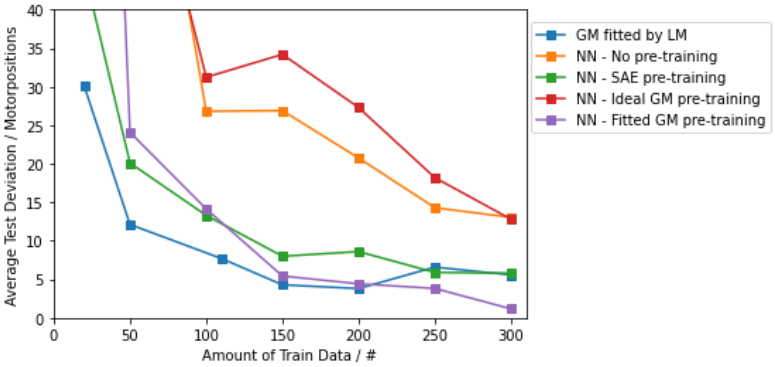


Figure 35: Summary of all tested algorithms. For all algorithms, the accuracy dependence on the amount of data is clearly visible. The highest accuracy is achieved by the neural network pretrained by the fitted GM.

cessed as described in Section 4.2.5. Either the network is trained with the data unsupervised using a SAE or the data is used to generate a data set with which the network is pretrained with. Since the data set only contains 500 data points, no more than 300 were used for training, as the test data set would otherwise have become too small to make reliable statements. Fig. 35 shows the average prediction accuracy of the conventional LM-method (blue) and the neural networks, with and without pretraining, depending on the amount of data.

The behavior of Levenberg-Marquardt (blue line) and the neural network trained from scratch (orange line) as described before, can be seen more clearly here. The accuracy of the GM fitted by the LM is increasing up until about 200 data points, but then it stagnates at about the same or slightly less accurate level. While the predictions made by the neural network trained from scratch are improving steadily, but never achieving a competitive (or even usable) accuracy. Surprisingly, the results of the neural network pretrained with the ideal GM turn out less precise than those of the network trained from scratch.
Since a considerably smaller LR is used in training than in pretraining, it is possible that a poor local optimium was found through this pretraining, which could not be overcome in the subsequent training process.
Unsupervised pretraining with SAE ends up at about the same level as the GM fitted by the LM algorithm with a mean accuracy of 5.8 motor positions. It performs among

the best and significantly better than the non-pretrained network.
The highest accuracy overall is achieved at 300 data points by the neural network with the pretraining by the fitted GM, reaching an accuracy of 1.21 motor positions. As described in section 4.2.5, the network was pretrained using a data set generated with the GM Model from the 200 datapoints training cycle, since the LM did not improve its test loss with more data points.

The best results of the neural networks have been summarized and plotted against the recorded data set in figure 36 . The blue area marks the training data set of the neural networks. Gray is the overall best result of the LM algorithm.
In this plot it is particularly clear that the pretraining by means of the ideal GM did not work. Especially in the back part of the plot the outliers become extremely large and would probably diverge further for later measurements. It is also interesting that (even if exaggerated and more noisy) the neural network shows the same features as the LM algorithm (three peaks and a plateau). This suggests that the behavior of the pretraining was transferred to the network and not corrected by the actual training. This hypothesis is supported by the fact that the GM pretraining (violet line) also includes these peaks while the SAE pretrained network (green line) has no clearly visible salient features. Since the SAE pretraining never saw a GM, but the other two networks did, it is

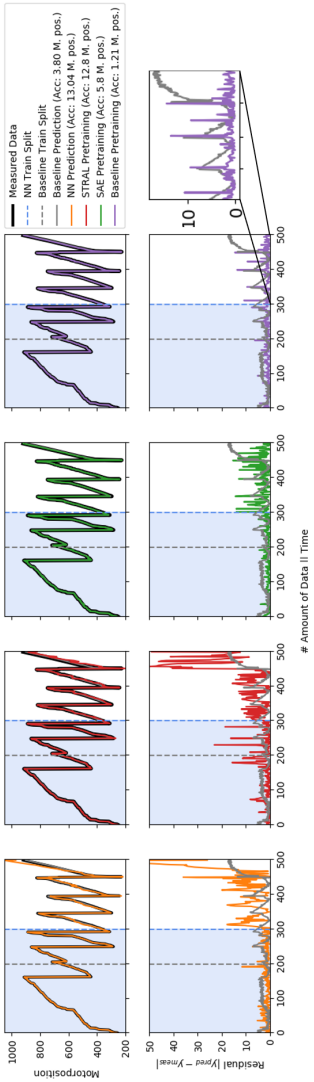
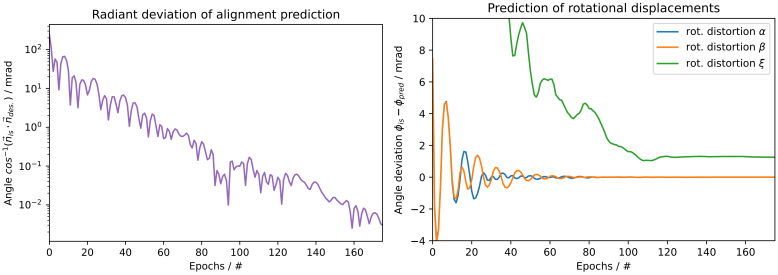


Figure 36: The best predictions of all tested methods. For a better comparability the best result of the baseline model, using 200 calibration points, is also drawn inside every graph.

very likely that the peaks are artifacts from the pretraining and that they will disappear with larger data sets. This also indicate that the peak behavior demonstrated by GM in the previous chapter can be attributed to the low complexity of the geometry model.
While the overall prediction by SAE pretraining is not better than the LM, the behavior in the test data set suggests that the basic mapping function was indeed learned. The prediction is noisy, but very steady over time. Here, some more data could already be sufficient.
As already mentioned, the violet line still includes the pretraining peaks, but instead of wide curves, they have been reduced to sharp spikes. Despite these peaks, the network achieves a mean prediction accuracy 1.21 motor-positions. In a first approximation this corresponds to an uncertainty of 0.42 mrad. This is 5 times better than the LM results with 300 data points and 3 times better than the overall best result in this comparison.



(a) Alignment deviation of the trained heliostat to the target data set heliostat.

(b) Angular deviation of the three error parameters inside the GM

Figure 37: Test losses for heliostat calibration using differentiable ray tracing over several epochs.

5.3. Results on Heliostat Calibration Using Differentiable Ray Tracing

A simple Toy Example using only 3 error parameters is used to evaluate the heliostat calibration. Although this cannot completely describe a real heliostat, it has another decisive advantage for method comparison. The 3 free variables have only a limited moving space and only one local optimum, which coincides with the global optimum. The results of the regression are therefore not on a convergence band and must not interpreted but give a clear statement about which approach is better. Therefore, a heliostat model with a deflectometric measured surface was loaded into the ray tracer and assigned with random alignment errors on α, β and ξ to create the target data set (compare 4.1.2) containing one image. The calibration

process itself starts with an ideal heliostat (including an ideal surface) and optimizes the heliostat's alignment by varying the error parameters.

As shown in Fig. 37a the angular deviation of the initially ideal and the erroneous heliostat is constantly falling. Due to the simplicity of the GM, the function has only one local optimum which is identically to the global optimum, which is why the deviation will fall down to machine accuracy. This is not surprising. Other regression algorithms, such as the LM, would perform similarly on such a simple GM. Thus, the plot verifies only the basic functionality of the methodology and proofs the existence of the global optimum.
More exciting is a look at the individual error parameters in Figure 37b. Here the difference between the alignment errors α, β, γ and the predicted angles are shown. A difference of 0 corresponds to an exact prediction. It shows very clearly how α and β oscillate continuously towards zero. ξ, on the other hand, behaves completely differently. This is due to the fact that the gradients are formed in another way. ξ would remain completely constant only using the $L2_{Alignment}$ loss (or equation 1). ξ is only optimized over the $L2_{Pixel}$ loss. In Fig. 37b ξ was set to 17.5 rad and ends up around 1.5 mrad, a deviation which is not visible for the naked eye. This is especially impressive because ξ does not scale with distance. No matter how far the heliostat is from the tower, it does not make the rotation more visible.

5.4. Discussion of AI Methods for Heliostat Calibration

Altogether, three different methods for heliostat calibration were considered for this work. (1) the GM fitted with tabular data by the LM algorithm, which is already established in Jülich, (2) the pretrained neural network, which uses the same data, but is purely data driven mostly without physical premodeling, and (3) the differentiable ray tracing which also relies on a GM, but works directly with image data.

Using a GM fitted by a classical regression works at the lower end of the scale in terms of data quantity. Even with fewer than 100 data points, higher prediction accuracies are achieved than by the randomly initialized neural network trained on the entire data set (compare Fig. 34). It would be possible to achieve even better results on small datasets by using fewer free parameters in the GM. In this case, even on extremely small data sets (starting from the first measurement), results can be achieved which are suitable for heliostats close to the tower and simple control strategies. Since parameters cannot be easily removed from NNs, such small data ranges are impossible to achieve for purely data-driven NNs. Thus, the results shown using all 8 error parameters do not represent the best results using the minimum required amount of data points but the overall maximum accuracy (within a certain convergence band) of the GM.

Furthermore, for the comparison at hand, the upper data range is more interesting, because the GM stops improving at 100 data points. With a best result of 3.8 motor positions on 200 data points, the GM achieves a good average accuracy. However, this value alone is not significant. Graph 33 shows how much the data actually scatters. With a deviation up to 15 motor positions the GM is unsuitable for modern aiming point or closed-loop control strategies. Especially for the investigated heliostat at a distance of 200 meters, it can happen that the tower is missed even in conservative operation.
In addition, it should be noted that the GM does not make any appreciable progress above 100 data points. It even becomes less accurate above 200 measurements. This most likely has to do with measurement inaccuracies and errors, which are both not covered by the GM.

The neural networks behave exactly in the opposite manner. None of the networks could achieve any meaningful results using only (or less than) 50 data points. Instead, all accuracies rise in every new training cycle using more data (compare Fig.35). It is obvious that no network has reached its global optimum inside the available data range. In fact, also the curve of the NN trained from scratch (orange line) allows an extrapolation that provides good accuracies in a few hundred data points. However, the utilized 300 training data points already represent an unusually large data set. A further increase of data is not possible, at least not with the measuring method used.

This makes neural networks trained from scratch ineffectual for use at solar towers so far.

The pretraining with an ideal GM even has a negative effect on the predictions of the network. As shown previously in Fig. 30, the ideal GM and the real values differ strongly in their motor positions. Probably due to this the network converged to a local minimum, which it could not leave while training. Such a (nearly) non-data driven method is not recommended for pretraining.
Only the data-driven pretraining methods make a realistic prediction possible within the available data range. The SAE pretraining reached its minimum at 250 data points, with a mean accuracy of 5.8 increments. Thus, the results are about as accurate as those of the classical regression. Even if the average prediction is slightly less precise than the classical regression, the SAE results should not be underestimated. On the one hand, the errors of the prediction are not broad, but sharp peaks. An error control that checks the prediction of the network, and cuts off outliers can further increase the accuracy of this method. On the other hand further improvements are conceivable with increasing data sets on this method, while the GM will stagnate at the same level. This type of pretraining does not require any modeling of the heliostat and can therefore be used directly on other solar towers. With this in mind, the neural network using SAE pretraining is superior to the classical regression starting at around 150 data points.

The pretaining using a data set created by the fitted GM already achieves competitive results starting at 150 data points and continues to improve beyond that. At 300 data points the best result on the available data range is reached, achieving a mean accuracy of 1.2 increments (approx. 0.42 mrad). This is a threefold increase in accuracy compared to the best result of classic GM. This accuracy is extremely promising. It means that on the one hand large parts of the heliostat errors, which are obviously not covered by the GM, are of a systematic nature and can be covered by a neural network. On the other hand, the pretraining can help the neural network to learn motor positions which are rarely accessed.

Compared to the theoretical accuracy of the camera target method of 0.1 mrad [40], the predicted 0.42 mrad are very close to this minimum. Other publications [54] [125] [126] using data measured in a solar field specify an accuracy of 2 mrad-6 mrad, which could be exceeded dramatically. *Prahl et al.* [126] is particularly interesting for the comparison, since it uses the same heliostats in Jülich and must therefore be able to deal with the same errors.
In literature only certain closed-loop methods, achieve similarly high accuracies [127], but these were achieved under controlled laboratory conditions. Taking into account that the data set for evaluation was recorded under real circumstances inside an existing heliostat field, the 1.2 increments or the 0.42 mrad are the highest accuracy ever measured under comparable conditions [40].

Of course, none of the values (from any regression) are absolute. All of them lie on a convergence band, which can behave differently depending on the data set and the starting conditions. Nevertheless, the stability of the NN methods is supported by the fact that both the SAE and the fitted GM model pretraining converge to similar values. And while classical regression has long since reached its limit within the available data range, the NN methods are still at the lower end of the scale and can still improve. To avoid the stagnation of the classical regression the LM could also be extended by an enlarged GM. Currently, the GM takes into account 8 free parameters. The whole model could be extended to e.g. 3 rotations and directional strains/compressions, which would result in 20 parameters (3 rotations and deformations each around the pylon as well as the 1st and 2nd axis of motion and 2 gear backlashes). Angle-dependencies, time-dependencies, nonlinearities, (systematic) stepper motor errors or even external influences such as wind would then not been taken into account. The network proposed here currently has exactly 18 nodes (including weights and biases) and has already proven to be able to cover most errors. The high entanglement of the nodes among each other also allows the mapping of nonlinear correlations. If e.g. local wind speeds should be included in the forecast, it is sufficient to add an input node. The biggest advantage (besides the high accuracy) of this approach is that a modeling of these effects and other disturbances can be omitted. This especially accelerates an overarching development to other

power plants.

Whether there are any nonlinearities in the investigated data or not cannot be said with certainty. The black box behavior of the neural networks does not allow to draw conclusions about the properties of the heliostat. Information about the torsion of the heliostat, for example, as provided by the GM, is lost. However, since such information is not used in daily operation, this is acceptable. However, the black box behavior must be given more attention compared to the classic control function. On the one hand, there is no clearly traceable control function after completion of the training with a neural network. This can already be seen from the peaks in Fig. 36. Unlike the GM, the error of the NN behaves abruptly. So, the heliostat might be tracking almost flawlessly one moment and miss the tower in the next. This requires a control function that smoothes the output of the network and detects outliers. In addition, care must be taken that at least the pretraining data set is balanced and includes the respective extreme values (sun position, deflection, receiver area). Because, although the networks are capable of abstraction to a certain extent, they work best for values between known extrema included in the (pre-)training. Aiming at a designated position that goes beyond the min-max values contained in the (pre-)training set, or a position that was rarely steered to could lead to higher errors. The heliostat examined here, for example, tends to drift to the lower east of the target center as described in Sec-

tion 4.2.3. Targeting a top west position could cause a larger error. A similar situation applies to the other input parameters. The sun positions can become more extreme depending on the season. For measurements taken in this work this is not noticeable, because the days became shorter during the measurement campaign. A different picture may appear from spring to summer. To avoid this the pretraining data set should include values from the entire year.
Probably the most uncertain point of this method is the prediction accuracy on the receiver. The network was trained exclusively with data pointing at the calibration target. As described before, the networks are very susceptible to predictions outside the previously known range. Pretraining can help, but giving the network an aiming point on the receiver when it has never seen these values in training will inevitably lead to a larger error. Here it must still be evaluated whether a linear offset on the calculated values or the network prediction on receiver coordinates results in the smaller error. Due to the 3 times higher accuracy of the network, the method should still have an advantage even with this error. This holds especially true since the ability of extrapolation should also increase with the data set.

The biggest disadvantage of the NN methods is the data set size. The used data set could never have been recorded in regular operation, especially for larger fields.
Therefore, either a measurement campaign is needed to

get enough data for every heliostat, e.g. this can be done during construction, or the data set has to be reduced further. For the latter, there are still some open possibilities. First, if the data set is not recorded on just a few days but over the whole year, the data variance rises. This can have a slightly positive effect on the data usage, but a measurement campaign would also be more difficult to perform. A second alternative would be the use of surrounding heliostats. These errors are not correlated, but as long as they are close to each other, the behavior is at least similar. As it can be seen in Fig.35, the closer the pretraining is to reality, the fewer data points are needed. Therefore, either the pretraining can be done with data from one (or more) neighboring heliostat, or the training process can be initialized with an already trained network with high accuracies to reduce the amount of data needed. In both cases only one to a few heliostats have to be measured extensively.

However, a reduction to just a few data points remains unlikely, neither because of the ability of the networks nor the GM, but due to the complexity of the problem. Without prior knowledge it is most likely impossible to describe such a complex system, including dozens of (linear) degrees of freedom and additional dependencies on alignment, stiffness, external disturbance factors as well as possibly aging processes, with just a few tabular data points down to an accuracy bordering an error free description.

In order to reduce the required data set while maintaining the same accuracy, there are 3 possible solutions.
The first is to use parameter studies to enhance the GM. However, such studies are highly specific. Their results are only applicable to one heliostat model and depend on the position of the heliostat in the field and the height of the tower. Such a solution is contrary to the goals of this work to promote the overarching development process of solar tower power plants and thus is not feasible. In addition, such parameter studies have already been made for the geometry model in Jülich. Nevertheless, a correspondingly high accuracy could not be determined in the course of this work.
The second option is a change of the measurement method. For example, the heliostat could be moved over the target in a neander shape and be filmed at the same time. This would drastically increase the amount of data. However, it also reduces the variance of the input distribution.
A third solution (without any drawbacks) is to obtain more information from the existing data sets.

This is the point at which calibration by means of differentiable ray tracing comes into play. Here, the information does not have to be extracted (e.g. by means of computer vision), but the image can be processed directly. This allows properties such as direction of incidence, rotation and inclination to be extracted from the images and taken into account. The additional information can then help with faster convergence or higher accuracy on the same

count of measurements. This also prevents errors of the preprocessing algorithms, such as when the centroid of the focal spot is incorrectly determined due to surface defects.

The investigated heliostat in the differentiable ray tracer (Section5.3) did not have an elaborated multi-axis GM. Nevertheless, the advantages of the method are already recognizable. On the one hand, it could be shown that the algorithms are identical under certain conditions (e.g. that not a single ray of light is generated within the ray tracer) and ray tracing thus has more access to information. On the other hand this information contained in L_{Pixel} can be used to determine ξ with an extremely high accuracy of ca. 2 mrad. Of course, direct image processing also reduces the data supply chain.

One of the most important advantages of diff. ray tracing is that information about the heliostat can easily be incorporated into the optimization. For example, surface information has a direct influence on the image loss. Having surface information changes the task to matching unique features of the focal spot. In this case loss terms for matching two fingerprints or unique faces [128][129] can be used. Similarity losses like the weighted hausdorff distance[130] could be used as well. Physical restrictions (e.g. an error parameter that cannot exceed certain values) can also be incorporated.

In summary, heliostat calibration can be improved with

the newly presented AI methods in-situ for both small and large data sets. On the lower range, more information can be extracted from the available data by handling images and advanced loss functions. On the other hand, non-linear errors can be represented completely without complex pre-modeling and with unprecedented accuracy.

5.5. Results of Heliostat Surface Diagnosis Using Differentiable Ray Tracing

The superordinate goal of this section is to enable the most accurate flux density prediction possible. For differentiable ray tracing, this process is closely related to the ability to reconstruct the surface of the heliostat. As derived in Section 3.1.4 this reconstruction has a theoretical limit but can be shifted to higher distances by adding regularizations. For the description of the surface NURBS functions were chosen within this work. Their ability for regularization will be tested in the following.

For this purpose, a total of 4 heliostats were created with artificial surface defects. The surface defects are sinusoidal and differ in their wavelength (compare 38, left column). The surface varies in its height by a total of 1.1 mm (slope deviation of 1.1 mrad). Within the ray tracing environment, the 4 heliostats were placed at distances ranging from 10 to 400 meters to create a corresponding target image (the sun position was randomly drawn, but identical for all training iterations). From the individual target image the respective surface should be inferred. Thus, each surface was reconstructed from exactly one target image.

The results of this are shown in Fig. 38 in the second to sixth columns. The reconstruction of rather coarse structures remains possible even for heliostats at a distance of 400 meters. At a range of 10m, significantly finer and

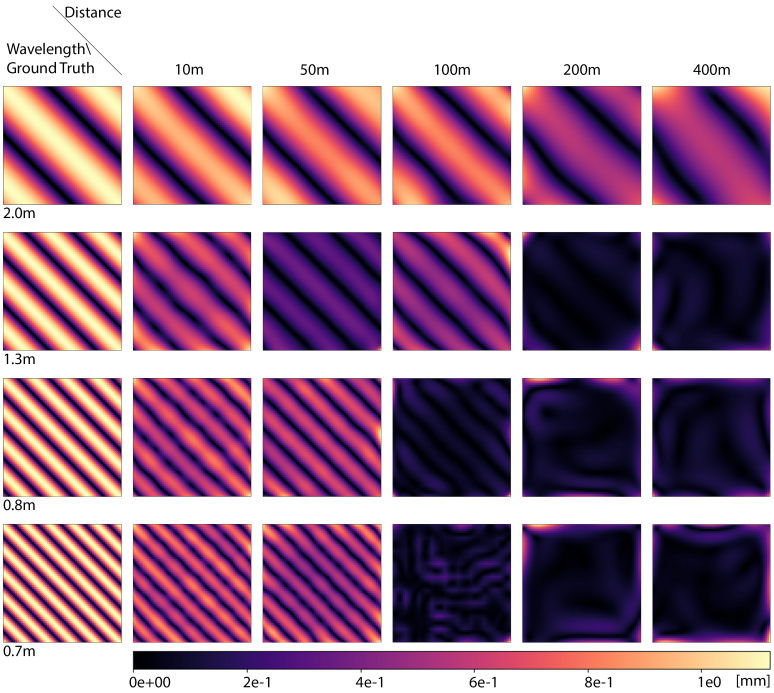


Figure 38: Artificially varied surfaces (left column) and their reconstruction from target images at different distances.

at 600m coarser sructures could still be resolved. However, the representation was omitted here. Depending on the detail frequency, the resolving power decreases as expected. All shown surfaces are completely unprocessed and use the same color range. This leads to the fact that explicitly the images of the surfacse at 1.3m/50m and 0.8m/100m appear worse than they actually are. Both surfaces contain an exaggerated regression artifact in the lower right corner, which affects the color representation. In fact, the average deviation of 1.3m/50m is lower than that at 1.3m/100m. This can also be seen in figure 39. In the graph, the average errors of the surfaces are plotted as a function of the training progress. The error is calculated by:

$$\Delta\overline{n} = \sum_{i}^{N} \frac{|\arccos(\vec{n}_{i,pred} \cdot \vec{n}_{i,groundtruth})|}{N} \tag{52}$$

where N is the absolute number of surface normals, $\vec{n}_{i,pred}$ is the predicted and $\vec{n}_{i,groundtruth}$ the predefined surface normal. The results cluster into 3 groups. Every surface below an average error of 0.4 mrad was reconstructed unambiguously. The range between 0.4 mrad and 1 mrad deviation is very mixed. There are very good reconstructions (50m/0.7m) next to those which failed partially (200m/1.3m) or completely (400m/1.3m). Maybe a stronger regularization or a better evaluation function is necessary for a successful convergence. Every error larger than 1 mrad is counted as a reconstruction failure.

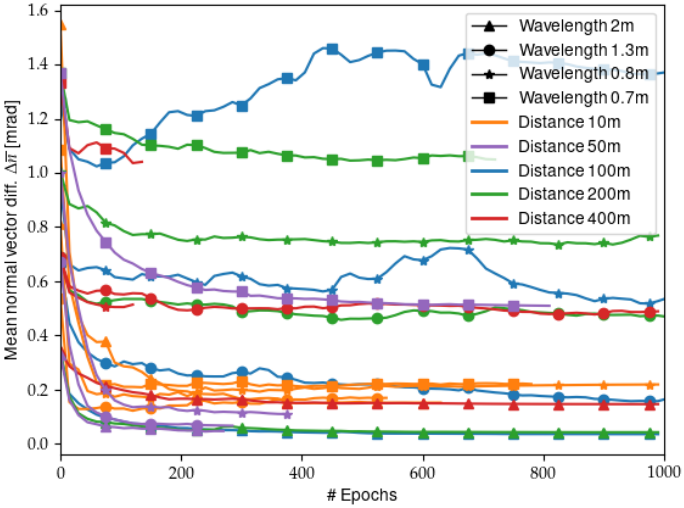


Figure 39: Learning curves of the shown surfaces in Fig. 38.

Comparing the reconstruction limits (where the surface reconstruction failed) with the theoretically derived limits in Section 3.1.4, a resolution was achieved, which covers about twice the distances as previously calculated, which is a strong argument for the introduced NURBS regularization of the surface.

The values derived from this theoretical analysis can only be transferred to a real heliostat in Jülich to a limited extent. There, most heliostats are faceted. These facets are in general canted and/or focused to a fixed focal length. To determine the resolving power for a real heliostat, one was deflectometrically measured in Jülich. The measured heliostat is located at East, North, Up (ENU) 13.2 m, 25 m ,1.795 m and is canted, but not focused. The calibration target is located at ENU 0 m, −3.23 m, 35.89 m. The measured surface was used to create target image data sets within the ray tracing environment as shown before in Fig. 24. Both the position of the heliostat and the focal length were varied to obtain reconstruction capabilities for different scenarios.

The results are shown in Fig. 40. It should me mentioned, when it is stated in the image that the surface was calculated from one target image, the sun position in Jülich on October 21 at 15:00 UTC was chosen (randomly). In case of more than one image the sun positions were drawn randomly, but identical for all multi-image surfaces.
The best reconstruction of the deflectometric measured

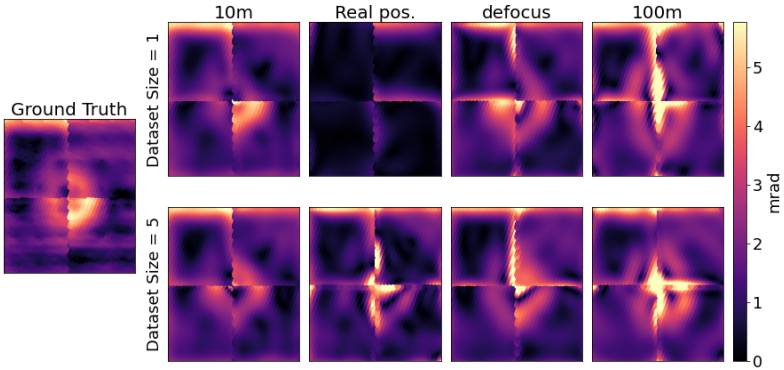


Figure 40: Reconstruction of a real measured surface from target images.

surface (labeled in Fig. 40 as the *Ground Truth*) is achieved at a distance of 10 m (first column). This is not surprising, due to the low underdetermination at such short distances. Although some details (mini wrinkles) are lost, the relevant features are reconstructed. In order to represent the mini wrinkles at short distances, the regularization (the NURBS order) can be further reduced and the number of degrees of freedom (NURBS control points) can be increased. In this case, however, some regression artifacts are formed, which worsen the representation of the surface. These can be filtered out by post-processing. However, only surfaces that were natively calculated by the ray tracer were used for this graphic. Thus, stronger regularizations were used than necessary for the degree of underdetermination. The bump in the center, the de-

formation in the upper left corner and even the facet on the upper right is slightly tilted as in the measurement. Adding more images does not change the quality noticeably.

In contrast, the reconstruction of the heliostat (Fig. 40, second column, first row) placed at its real position in the field fails. Since the heliostat is viewed at such a distance from the tower that it coincides with its focal length, all facets are completely superimposed and thus the surface defects are permutable along the facets. This not only intensifies the underdetermination, it makes a reconstruction impossible, at least for one point in time. The permutability is well shown by the deformation of the upper left facet in the ground truth, which can be found on the upper right in the reconstruction.
As discussed in Section 3.1.4, different sun positions can shift the focal plane slightly forwards and backwards. This can restore some information. Using more images (second column, second row), coarse details like the bump in the middle and the upper right facet canting become recognizable again. This will also be discussed in more detail later.
However, the canting and not the distance is responsible for the rather coarse reconstruction, which can be seen in the third column (defocus). There, the heliostat is placed at the exact same position but without canting, which results in a similar reconstruction quality as the heliostat placed at 10m distance. This shows very well how strong

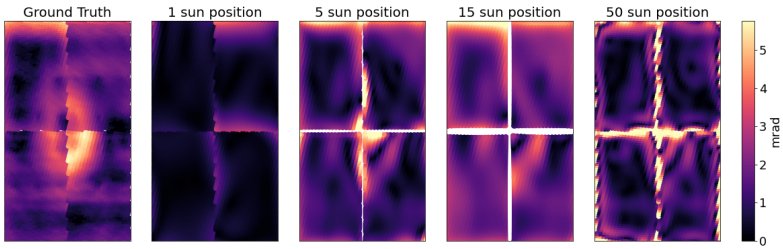


Figure 41: Surface reconstructions depending on the data set size. The surfaces using 1 and 5 images are the same as in Fig. 40

canting increases the underdetermination.

The reconstruction at 100 meters (fourth column) also suffers from a higher underdetermination due to both, the remaining canting overlap of the facets and the distance. Nevertheless, the reconstruction found all prominent features, even though slightly broadened.

To counteract the underdetermination given by the focusing, diff. ray tracing can use more data for training. In Figure 41, the reconstructions with one and 5 images are shown one more time. In addition, the reconstruction with 15 and 50 images is also displayed. From 1 to 15 images the reconstruction quality consistently rises. The heights turn out clearly smaller than with the Ground Truth, nevertheless the reconstruction accuracy is impressive if it is considered that the reconstruction with only one picture

Images	NURBS order	Mean surface deviation [mrad]
1	9	0.97
5	9	0.74
15	9	0.65
50	9	0.64
50	5	0.97

Table 4: Quantitative results of the surface reconstruction for different dataset sizes and regularization parameters. The surface reconstruction quality is given by the mean normal vector deviations from these of the deflectometric measured surface.

seems almost impossible. However, the quality also peaks at 15 images and remains at about a constant level for more images, using the same number of optimization and regularization parameters (compare Tab. 4). The shown image with 50 training images has a lower NURBS order than the images with less training data, thus the NURBS nodes are less interlocked. At first glance, the reconstruction quality suffers noticeably from this (see also Tab. 4), since for example the peak in the center is no longer displayed smoothly and flat but by local exaggerations. However, a closer look reveals that finer details, such as the wrinkles, are now displayed. In fact, the underdetermined nature of the problem, combined with the weaker regular-

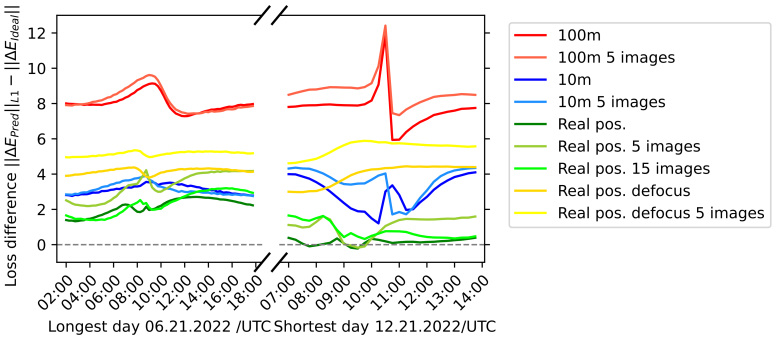


Figure 42: The graph compares the quality of the flux density prediction generated from two different surfaces by their difference. ΔE_{Pred} stands for the observed energy difference with the previously calculated surface and ΔE_{Ideal} for an ideal surface. The larger the positive difference to 0, the better the predicted surface performs.

ization, as well as the higher statistics given by the large dataset allow together to find a local minimum that gives better focal spot predictions (compare Fig. 44b) than all other reconstructed surfaces investigated here. This will presented in the following section.

5.6. Results of Flux Density Prediction Using Differentiable Ray Tracing

In order to test the flux density prediction of the previously calculated surfaces, flux density maps at different

sun positions were calculated using ray tracing and compared to those with an ideal surface.
In the course of a year, the sun has two extreme values: the longest and the shortest day. All other sun positions throughout the year fall between the values of these two days. Thus, the forecast quality for the rest of the year can be estimated at least approximately by interpolating between those days. This has been done in Fig. 42.
Here the times of the two extreme value days are plotted against the difference of $||\Delta E_{Pred}||_{L1}$ and $\Delta E_{Ideal}||_{L1}$. $||\Delta E_{Pred}||_{L1}$ and $||\Delta E_{Ideal}||_{L1}$ represent the $L1_{Pixel}$ deviation of the flux density prediction from those of the ground truth values. Where $||\Delta E_{Pred}||_{L1}$ uses the previously calculated surface and $||\Delta E_{Ideal}||_{L1}$ uses the ideal. The larger the positive difference to 0, the better the calculated surface performs and vice versa. The relative representation was chosen for two reasons. First, it can be seen directly at the null line whether the model leads to an higher or lower focal spot prediction accuracy. Second, the absolute improvement of the model is dependent on the selected resolution of the flux density map. By the chosen representation the relation remains independent of it.

In the previous section, the heliostat at 10 m distance performed best at the surface reconstruction. This however is not reflected in the prediction of flux density, as can be seen in Fig. 42 (blue lines). It performs less effectively than the source data collected through the defocused heliostat (yellow lines) with a similarly precise reconstructed

surface and also noticeably less precise than the prediction using data from the diagnosis of 100 meters (red lines). This is due to its short distance to the tower. Surface defects have a stronger impact at a higher range as they scatter more strongly. Thus, the deviation between the focal spots created by reconstructed and ideal surfaces also increases with the distance. Accordingly, even a very roughly reconstructed surface (as at 100m) is of a considerably greater benefit to the flux density prediction than a very precisely calculated surface at a shorter distance.
Furthermore, it can be seen that the imprecise reconstructed surface of the heliostat with canting at its real position (green lines) only results in a small improvement. For some minutes on the shortest day the prediction accuracy even drops (slightly) below zero. On the longest day of the year however, the improvement is clear.
Interpolated throughout the year, an improvement of the flux density prediction can be observed for all heliostat surfaces trained on one image. This increases when further images are added, even though the improvement is not consistent for all increments (blue and green lines).
The heliostat with canting at its real position forms another exception here. Although the use of 5 images improves the prediction (in the late evening hours of the longest day it even reaches the accuracy of the uncanted heliostat), the accuracy still drops below zero between 9 and 10 a.m. This could only be prevented by using 15 images. However, this also results in a reduction of the improvement throughout the year. Here it becomes espe-

cially exciting if a look back to the last section is taken. The reconstruction with 5 images seems less accurate with 15 images. However, the average prediction is better with only 5 images. Here it can be seen for the first time that surfaces do not necessarily match the real ones can lead to very good results. In the later discussion, this fact will become more evident.
Despite the moderate improvement of the green lines, an interpolation between the longest and the shortest day allows the assumption, that the overall forecast for the year will improve compared to the state of the art prediction. This will be evaluated in more depth in the following on the example of the green lines in Fig. 42. In Fig. 43, the positions of the sun during the year are plotted at half-hour intervals. The color coding in the top row indicates the relative improvement of the flux density prediction. The bottom row additionally shows whether a prediction is better ($||\Delta E_{Pred}||_{L1} - ||\Delta E_{Ideal}||_{L1} > 0$) or worse ($||\Delta E_{Pred}||_{L1} - ||\Delta E_{Ideal}||_{L1} < 0$) than without any surface reconstruction.
In contrast to the previous plot, it can be seen that not a single value throughout the year is negative (bottom row). This can be explained by the fact that the temporal resolution is significantly lower here. The previously very local dropouts are simply not included. This speaks very much for two facts. First, the dropouts only happen on the shortest days of the year, and second, they occur very rarely.

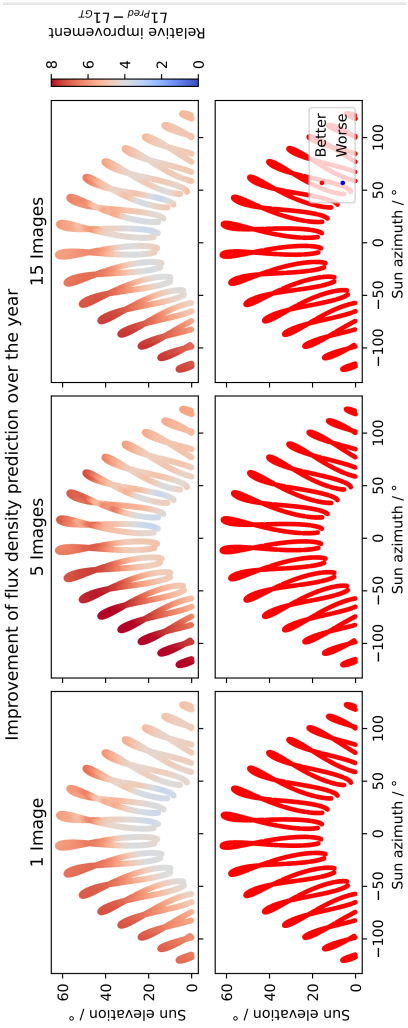
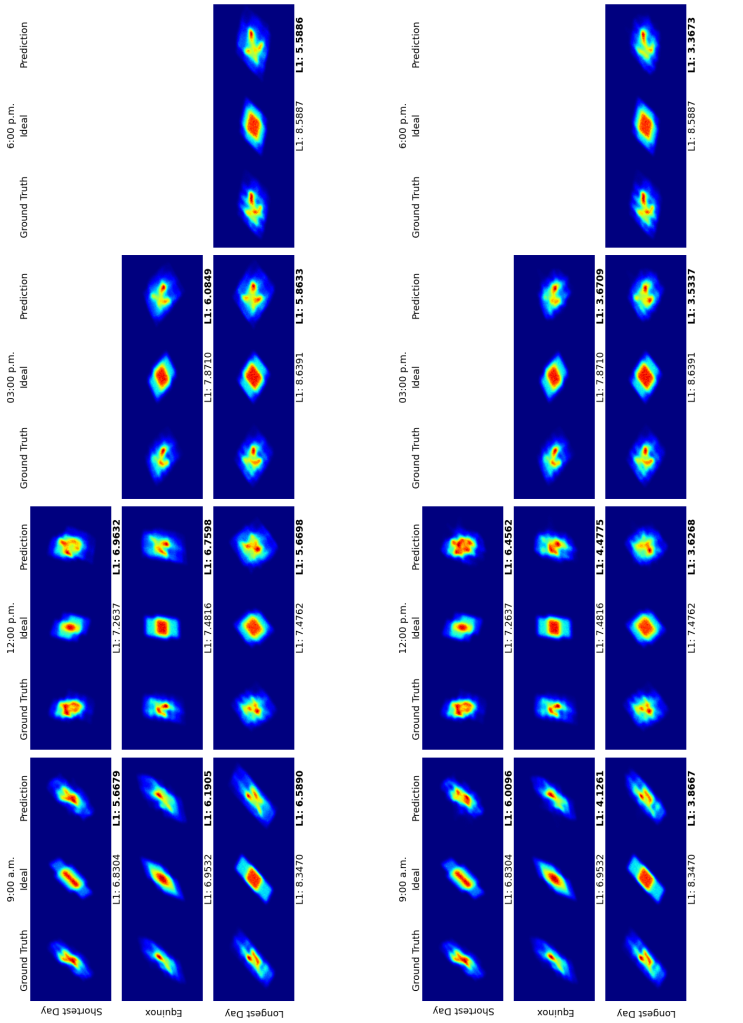


Figure 43: Relative improvement of the flux density prediction over the year compared to a non-optimized surface. For the bottom row it is additionally checked if the quality of the predictions is positively or negatively affected by the surface reconstruction.

This becomes more clear in the top row. There, the relative improvements of flux density prediction over the year are shown. Here the gradient between the longest (upper envelope function) and the shortest day (lower envelope function) of the year is more noticeable. The improvement is stronger throughout the summer months. This behavior can be explained by the position of the sun. On the longest days the sun is more perpendicular to the heliostat normals. This results in larger focal spot, than on the shortest days. In addition, the individual features are more concise. Both result in a larger L1 difference.
This can also be seen in Fig. 44a, where the focal spots of the measured (ground truth), the ideal and the predicted surface trained on 15 images are explicitly shown. The corresponding focal spots for 1 and 5 images are presented in the appendix Chapter D, because they are not necessary for the upcoming discussion. Moreover in this visualization the quality of the images can be inspected.



(a) Heliostat model trained on 15 images.

(b) Heliostat model trained on 50 images.

Figure 44: Comparison of deflectometrically measured flux density profiles with measured, ideal and trained surfaces throughout the year.

Once again, different times on the shortest day, the longest day and additionally the equinox in spring are visualized. Despite the very good quantitative improvements, the qualitative similarities of the prediction to the Ground Truth (GT) are astonishing.
Many of the prominent features are represented very accurately. Only mini wrinkles are not displayed correctly. This is due to the strong regularization. Because of the high underdetermination, (even with 15 images) it must be avoided to find only a local optimum or in other words to avoid overfitting. This is regulated by a stronger interlocking of the NURBS control points. The high NURBS order thus manages to map the shape of the surface but not its local disturbances.

In Figure 44b, this high regularization has been weakened (the NURBS order of 9 has been reduced to 5). Instead of physical regularizations, data-driven regularizations were used. Here, the surface was trained with a total of 50 focal points (the images are generated using the surface shown before in Fig. 41). Using the data-driven regularization the details in the images become sharper, and also the mini wrinkles are visible in the focal spots (e.g. equinox and longest day at 3p.m. and equinox at 12 p.m.). Nevertheless, data-driven regularization comes at a price. With 50 images, the optimization took over 3 hours. With the high image quality achieved, which comes close to that of deflectometry, it is worth the time.

Images	NURBS order	Mean focal spot impr. [%]	Max focal spot impr. [%]	Mean surface deviation [mrad]
1	9	25	52	0.97
5	9	42	84	0.74
15	9	39	47	0.65
50	9	40	46	0.64
50	5	77	155	0.97

Table 5: Quantitative improvements of the flux density prediction (second and third column) for different data set sizes. The improvement is calculated from the mean ratio of the focal spot prediction with and without surface reconstruction over the year. The fourth column shows the mean normal vector deviations from these of the deflectometric measured surface.

Finally, the quantitative results of the flux density prediction and the surface reconstruction will be summarized. Tab. 5 shows the average and maximum improvements over the entire year depending on the data set size. The improvement is calculated by

$$IMP = 1 - \frac{L^1(\text{Ideal, Ground truth})}{L^1(\text{Pred, Ground truth})} \tag{53}$$

Where L^1 is the mean absolute error. The result of the calculation depends on the size of the window in which the focal spot is viewed. The closer the window is drawn around the focal spot, the greater the improvement. The window for Tab. 5 was chosen in such a way that focal spots can be completely mapped at any time of the day or year. For the evaluation, sun positions were simulated in half-hour intervals for the entire year 2022. Especially for the data set with 50 images, the accuracy of the prediction (despite erroneous surface reconstruction) stands out. With 77% more accurate results over the entire year and 155% maximum improvement, the prediction almost reaches the accuracy of deflectometry. Even though the improvement is significantly lower when training with only one image, the effect is also clear here. Even if the real surface was not found in the calculation, an improvement of 25% can be achieved compared to the calculation without surface information. It is worthwhile to perform the surface optimization in any case.

5.7. Results of Flux Density Prediction Using NeRF

Lastly, the results of the NeRF network will be reviewed for its ability to predict the flux density map. NeRF was trained with different amounts of data, depending on the data set (*Sanity Check*, *Semi Sphere* see Section 4.2.1). The data sets differ in the level of detail and the difficulty of the extra- or interpolation. Reconstruction of the test set should be much more difficult in the Semi Sphere data set. Nevertheless, NeRF achieves nearly identical prediction accuracies with both data sets (see figure 45). On the entire test set, the Semi Sphere data set achieves a peak signal to noise ratio of 27.1 and the Sanity Check one of 27.8. This is a lot lower than the image quality in the original NeRF publication (the best reached a ratio of about 31). Although, there are individual predictions with high accuracy (Sanity check 8;9, Semi Sphere 12;13), the prediction is heavily blurred, inaccurate and dependent on the incident angle. Moreover, there are angles under which the reconstruction fails completely. Interestingly, these angles are independent of the examined data set. In Figure 45, the images 1 and 17 are particularly inaccurate for both heliostat types. Thus, at least one of the previously made approximations (resting heliostat, constant illumination) must fail in this region.
A further comparison with presented values throughout the year is not necessary at this point, because the image quality can already be found as insufficient by eye.

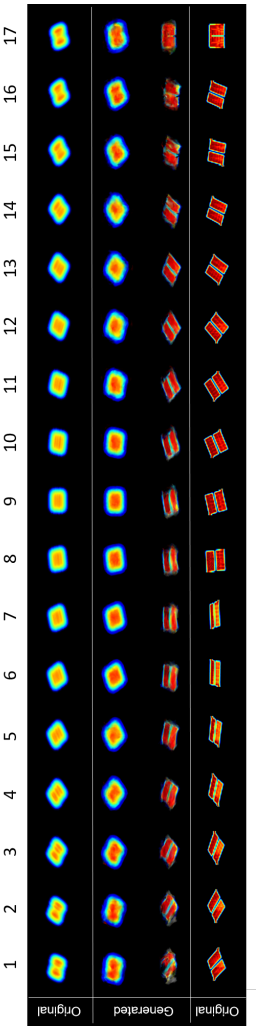


Figure 45: Predictions and ground truth of the flux density map of two different ideal heliostats predicted by NeRF. The upper two rows show the sanity check, the lower two rows the semi sphere data set.

5.8. Discussion of AI Methods for Heliostat Surface Diagnosis and Flux Density Prediction

Starting with the recently presented NeRF architecture, it can be said that the results were not very convincing. None of the data sets was able to provide sufficient accuracy for focal spot prediction. Especially for the *Sanity Check* data set this is a tough statement. The data set includes twice the amount of data as used in the original publication by *Mildenhall et al.*. In addition, the entire hemisphere is limited to a small fraction and the extrapolation for the test data set is very small. Despite these easy conditions, a good reconstruction could not be achieved. Furthermore it can be expected that the prediction breaks down further for angles with a higher deviation to the training data set.
While NeRF does not perform any better on the *Semi Sphere* data set, the similarity of the prediction to the *Sanity Check* is remarkable. With half as much (supposedly more complex) data to process, less accurate results would have been expected. However, the quantitative results are about the same level. This effect can assumable be explained by the larger angles between the images inside the set also increasing the amount of information available to the network.
The rather moderate image quality for both data sets can be explained by the many applied approximations. They not only negatively affect the overall image quality, but

are most likely also responsible for the collapse of the image quality in some angles.
In addition, the mediocre image quality is accompanied by an extremely long computing time. The *Sanity Check* data set took almost 8 hours to run on an *NVIDIA 2080TI*.

NeRF thus has many disadvantages. Where, runtime, static conditions of the scene, and constant illumination can be compensated by subsequent releases, this does not hold for the ray pathing used in NeRF. This simplification will not be able to model the physics at solar towers. In principle, ray tracing could be implemented in subsequent NeRF iterations (or the other way around, the NeRF networks can be integrated into the diff. ray tracing tool). However, especially against the background of the results using differentiable ray tracing alone, the approach using a neural network seems highly questionable.

It starts with the principle methodology. Both methods calculate their gradients physically motivated and update their parameters using the same AI methods (Backpropagation, Adam, LR Deacy etc.). However, differentiable ray tracing uses only the part of the rendering equation relevant for the use at solar towers (the reflection equation). In contrast NeRF uses the whole rendering equation, but in a simplified version. Diff. ray tracing is therefore more efficient and provides better gradients. In principle, this kind of gradient calculation could also be implemented in NeRF, but there are other arguments against

the use of neural networks.
The main argument are the NURBS for surface description. Using NURBS, the surface is modeled highly efficiently and complete. Mathematically, there is no heliostat surface that cannot be described by the NURBS. At the same time, it can be described with a minimum number of control points, and thus with the fewest degrees of freedom. This is really important for solving a task as underdetermined as the surface reconstruction at the solar tower. In addition to that position of the control points is dynamically adjusted during training. Local defects can be mapped as well as long-wave, heliostat-spanning deformations, by a dynamically and optimizable density of control points. Therefore, even if no neural network for surface description is integrated, the NURBS have the possibility to provide any number of degrees of freedom for a complete surface description and are able to strongly regulate the surface with efficient physics at the same time.

Both properties are necessary for the heliostat surface diagnosis. In Section 3.1.4 it was approximated how far a non-regularized system can uniquely resolve the surface (compare Fig. 18). Based on the artificially generated surfaces shown in Fig. 38, it can be deduced how well the NURBS regularization works. In this comparison the NURBS regularization allows a reconstruction of approx. twice the distance as theoretically derived. Without canting, even a reconstruction of coarsely resolved surface de-

fects up to 400 meters is possible. This is a strong argument for the use of the NURBS surface description.
In Fig. 40 it could also be shown how much influence the canting has on the reconstruction process. Nearly all surfaces could be reconstructed with low (at a distance of 100 m) to very high accuracy (in a distance of 10 m and the real position without canting). To obtain such clearly resolved surfaces (outside the focal length and with one image), a strong regularization by the NURBS was needed. However, this did not resolve the fine wrinkles of the heliostat structure.
In exact focal length a different picture appears. Using only one image, the high underdetermination (due to the permutability) of the system prevents a real surface from being found. More images (up to 15) increased the image qualty a lot and a reconstruction becomes possible.
What is interesting about this is that it could be shown that realistic surfaces are not necessary for a focal spot reconstruction. Even in a completely failed reconstruction with only one image in exact focal length, an average improvement of 25% throughout the year could be achieved (Fig. 41). Previously, ADAM was used as the optimizer for optimization with only one image. For this underdetermined problem, such a local optimizer is only partially qualified, since the training can quickly fall into a minimum that it can no longer leave. If only one image is available for use on the tower, a global optimizer such as Basin Hopping should be considered. With this, probably significantly higher improvements can be expected.

For a better focal spot prediction with finer image details, the physical regularization can also be replaced with a data-driven one. The NURBS can also be applied for this purpose. By using more images (Fig. 42), the average prediction accuracy could consistently be improved. The only exception is the transition from 5 to 15 images of the heliostat in exact focal length (Fig 42, green lines) to the tower. Here, the additional images prevented the accuracy of the prediction from becoming negative at some points in time, but the average accuracy was lower than the training with only 5 images. Despite the low average accuracy, distinct features of the focal spots are depicted very well in the individual images of the focal spots (Fig. 44a), which seems to contradict these results. The quantitative deviation can be explained by a larger scatter/blur in the edge areas of the focal spots, which form a slight corona and thus negatively influence the loss. The discrepancy between average accuracy and high image quality can probably be resolved by simple post-processing steps like threshold operations, as it was done in 4.1. Even without these steps, a consistent improvement of the annual forecast could be achieved (Fig. 43). Nevertheless, the images are lacking in fine details, which is due to the strong physical regularizations that are still needed using only 15 images. Physics and data-driven regularizations work against each other in this data range. This barrier is overcome through the use of even more images.

In Fig. 44 the flux density maps of the surfaces gener-

ated by 15 images (Fig. 44a) and 50 images (Fig. 44b) are compared. Using 50 images, the quality of the images is increased by far. A mean improvement of 77% and a maximum of 155% is stunning. The focal spots now include fine details e.g. like mini-wrinkles. However, this did not have a positive effect on the reconstruction of the surface
But, even if the reconstruction of the surface is negatively effected by the canting and the surfaces do not have a high agreement with reality, extremely accurate focal spot predictions can be made throughout the year.

The solar tower in Jülich is by far not the largest solar power plants. Distances to the tower of up to 1.5 km are quite possible. At such distances it becomes impossible to reconstruct the surface of a heliostat correctly using only its focal spot. Nevertheless, the NURBS surface is an ideal description of the heliostat mirror. With it, the mirror can be described efficiently with the smallest possible number of free variables. How well this works could already be observed at the heliostat in exact focal length. Although the reconstruction of the surface has not a high agreement with reality, focal spots can be generated which inherit an enormous depth of detail (compare Fig. 44b). A focal spot at an extremely long distance does not show such fine details at all. Which is why fewer free variables and thus less images can be expected to produce similar efficiency gains.

In summary, the results provided by NeRF were not very convincing. On the one hand, the approximations made could not be met by the physics at the tower. On the other hand, the use of neural networks for heliostat surface reconstruction is generally questionable. To describe such a strongly underdetermined system with an immense number of free parameters, but without strong physical regularizations seems to be the wrong approach, especially in comparison with the results of diff. ray tracing.
This, in turn, supported by NURBS, is capable of reconstructing heliostat surfaces from its focal spot at a distance of several hundred meters away from the solar tower. Outside the focal plane, a single calibration image is sufficient for this purpose. One image is also enough to improve the flux density forecast for the whole year with exceptions of only a few minutes. As the data set grows, these gaps can be closed and image quality can be improved both qualitatively and quantitatively.

6. Summary of Digital Twin Methods for In-Situ Solar Tower Plant Optimization

After the individual algorithms for the respective tasks (heliostat calibration, surface diagnosis, flux density prediction) have been discussed, the methods themselves, namely the diff. ray tracing and the neural networks, will now be discussed together.

6.1. Discussion

Differentiable ray tracing performed successfully in all areas tested and, is as far as the author knows, the first algorithm capable of extracting this extensive amount of information from heliostat focal spots.

The *heliostat surface diagnostics* delivered impressive results. In particular, the NURBS surface used for regularization was able to show its strengths here. The more data is available, the more control points and the lower the NURBS order can be set. With the strongest, purely physical regularization, heliostat surfaces could be reconstructed with only one cal. target image at a distance of up to 400 meters (Fig. 38). Also the high underdetermination, caused by the canting could be counteracted by the data-data driven approach (Fig. 40). Nevertheless, it could only be pointed out that a target image, which was not taken directly in the focal plane, leads to considerable improvements of the reconstruction, even if the heliostat is moved further away from the target for this purpose.

By (partially) dropping the physical regularization and using more data, the *flux density prediction* could be improved quantitatively and qualitatively (Fig. 44a to 44b). Even the mini wrinkles of the heliostat could be displayed in the target images. For all data set sizes, starting by one image, an improvement for the entire year could be achieved 43.

NeRF had considerably more images available for the same task and could not deliver acceptable results despite the additional degrees of freedom (Fig. 45). Both the physics within NeRF (raypathing) and the approximations made for the application at the solar tower do not seem to be valid. Due to the astonishing results of the NURBS supported diff. ray tracing, the work on this type of network was discontinued.

Although the *heliostat calibration* could not be compared quantitatively, it could be shown in Section 3.1.6 that the algorithm used in Jülich and the diff. ray tracing are identical under certain minimalistic conditions and that by abandoning these, only the ray tracer can benefit. Calibration by means of diff. ray tracing works with images and additional loss terms and therefore has more information available than the conventional heliostat calibration (which uses tabular data only). Thus, the required data set for the calibration can either be reduced or the accuracy can be increased by additional variables while keeping the amount of data constant.

Nevertheless, using the same variables as studied in Section 2.4, the algorithm will also converge to the same poor predictions as the LM in Fig. 33. This is where the white box model meets its limits. On the one hand, the premodeling of every single expected error is a very difficult, maybe even impossible task. On the other hand, the number of these errors and thus degrees of freedom would far

exceed the available amount of data. The presented black box models help here in two ways.
Firstly, the nodes of the network are strongly interconnected and can therefore also represent non-linearities with a more or less similar number of parameters as the GM (see Section 5.4).
Secondly, the pre-modeling of the parameters is completely omitted, at least in the unsupervised pretraining. A network trained in this way was able to achieve accuracies similar to the geometry model used in Jülich on the existing data set of 300 training points. The difference is that the accuracy of the GM started to stagnate after 150 data points, while the accuracy of the NN was falling for the whole range and thus further improvements are to be expected. In addition, since no GM has to be tailored to a specific heliostat class, the network is also transferable to other solar towers (after re-training).
For the supervised pretraining a GM still has to be premodeled. However, a rudimentary one like the one used in Jülich should generally be sufficient. With this, an unprecedented accuracy of 0.42 mrad could be achieved for a heliostat at a distance of 200 m away from the tower.
In both cases, the pretraining served as a data-driven regularization and prevents the network from overfitting despite the many parameters. Further improvements are still expected with additional data.

In contrast to the heliostat surface, which is excellently described by the NURBS, the heliostat's alignment model

don't reaches realities complexity. Neural networks are the better choice for this problem. As shown in Fig. 36 they are capable of catching almost any error if provided with enough data.

However, exactly the latter remains a challenge. The 250-300 training data points used are not only at the limit of what is feasible, for large fields this even goes beyond that scale. Here, either additional measurement campaigns or new measurement methods are necessary to generate enough data for the NN. But based on the unprecedented high accuracies of the new method, this additional effort seems worth it in any case.

An alternative approach to both possibilities is to integrate the MLP into the diff. ray tracing environment to further reduce the required amount of data. The differentiability allows the neural network to be placed directly at the location of the alignment function within the environment. Input and output parameters remain identical for the function, only the content changes from linear algebra to a neural network. Thus, the network can also benefit from image data and new loss types. However, such an implementation still has to be tested.

Following this, the use of differentiable ray tracing as a digital twin environment on the solar tower is highly recommended, even for tasks beyond those presented here. Excellent heliostat twin instances can be generated and

also be operated independently of the actual power plant operation, e.g. for research. In addition, the ray tracer offers the possibility to replace individual functions with black box models. This makes it possible to always get the best out of the highly varying data sets on the tower. In summaray, as long as a sufficiently large data set is available, arguably no algorithm is better suited for heliostat calibration than a neural network. As long as the data set is still being built, diff. ray tracing should be used. This may change if the neural network is used inside the ray tracing environment.

6.2. Conclusion

This work explored the potential of differentiable ray tracing and neural networks in the context of solar towers. The research demonstrates that differentiable rendering is particularly suitable for the in-situ application at solar towers. The differentiable ray tracer, which was developed from scratch in this work is capable to optimize every relevant parameter in the heliostat field by means of measurement data. This digital twin environment can centrally control heliostats, derive their individual errors and predict future behavior.

By using images from the fully automated heliostat calibration, which is already implemented in most solar power plants, the approach is capable of predicting the irradiance fully automated and with unprecedented accuracy. This leads to a 25% improvement in flux density prediction throughout the year with just one image, and up to a 77% improvement with 50 images compared to state-of-the-art ray tracing methods. Extensive in-silico experiments suggest that these findings can be generalized to the entire heliostat field. Notably, this method is the first to extract information from heliostat focal spots to reconstruct corresponding surface deformations. Due to its computational efficiency and minimal data requirements, it can be implemented at a relatively low cost, necessitating only software adjustments.
The direct image processing also adds additional information to the heliostat calibration. With a first proof of

concept it was shown, that the calibration is now able to correct previously neglected errors. The use of images should not only reduce the data set needed for optimization but also allow more variables without the risk of overfitting.

In supplementary experiments outside the differentiable ray tracing environment, purely data-driven neural networks without physical constraints were tested.
These include pretrained MLPs for heliostat calibration and NeRF networks for flux density prediction. In both cases, the physical pre-modeling is replaced by the respective network structure. The MLP for heliostat calibration outperformed state-of-the-art methods, achieving an unprecedented accuracy of 0.42 mrad for a heliostat 250 m from the tower. A new benchmark also in comparison with literature. This method also enables the easy integration of complex measurement variables, such as local wind speeds, to further reduce tracking errors. However, achieving such high accuracy required a large dataset of over 300 measurements, which may be impractical for large heliostat fields. This limitation can be mitigated by incorporating these networks into the differentiable ray tracing environment and providing both physical information about the heliostat and additional information from the directly processed image data.
The NeRF networks were not convincing in the prediction of the irradiance. On the one hand, some approximations made, which were needed to adapt them to the calibration

task, could not be fulfilled. On the other hand, the additional free variables did not bring any further advantage. For this task, the differentiable ray tracing method has notable strengths: without restrictions, it is a physically motivated model, which on the one hand has enough parameters to represent reality completely, and on the other hand offers flexible surface regularizations.

Nonetheless, some aspects remain unexplored in this thesis. Real data from solar towers has not yet been tested in the ray tracing environment, and the integration of neural networks for heliostat calibration, which has shown promise, remains pending. However, subsequent research has successfully addressed these aspects.
Interested readers are directed to the publications *Pargmann et al. (2023a)* [131] and *Pargmann et al. (2023b)* [132]. The first publication introduces the application of differentiable ray tracing at solar towers with real data, achieving even more accurate flux density predictions than those with deflectometric-supported ray tracing. The second publication combines rigid-body kinematics with a neural network for heliostat calibration, achieving unprecedented accuracy starting from the initial measurement.

Ray tracing has occupied a central place in solar tower research even before this thesis and with the introduction of differentiability, it is now possible to employ ray tracing in a data-driven manner at solar towers. This approach combines the strengths of a physically motivated model with

the flexibility of modern approaches like neural networks. It offers a comprehensive representation of reality and can be supported by flexible regularization techniques. In the future, neural network-enhanced differentiable ray tracing could replace existing algorithms and serve as a central digital twin for open- and closed-loop control, diagnostics, and predictions in heliostat fields. It represents the key element in efficient, autonomous, and intelligent solar power plants.

6.3. Possible Future Applications for Differentiable Ray Tracing

In summary, differentiable ray tracing could already deliver very good results, e.g. in the reconstruction of heliostat surfaces from focal spot images, but overall it remains behind the resolving power of deflectometry. This is mainly due to the better local resolution of the light source. Such a light source can also be built directly into the ray tracer. With this the deflectometry can be performed much more stably since the solver is no longer based on geometric optics. By using diff. ray tracing the required amount of images is reduced to one (the deflectometry needs at least 3), which reduces the exposure time drastically. Furthermore, a wrong alignment of the heliostats no longer leads to the breakdown of the procedure. The alignment could even be optimized directly. The method should also be more robust against errors from from dew and dirt. Diff. ray tracing thus not only creates a substitute for deflectometry in everyday operation, it also offers a way to automate the known procedure. This gained stability can also be used outside the solar tower industry, whenever the laboratory conditions usually required for deflectometry cannot be met. Such a ray tracing supported deflectometry for industrial processes, such as in the aircraft or automotive industry, was patented in the course of this work.

Even if this work depends mostly on images taken by the

heliostat calibration, images on the receiver are of course also a suitable input. The differentiable description works here as well. Diff. ray tracing can make a gradient based optimization of the aim point distribution on the receiver possible for the first time.

Until now, only the optimization of the power plant in its existing state was considered. However, optimization can also start with the designing of the plant. Given a premodeled ideal flux density on a receiver, the heliostats in the field, their number, position or even mirror shape can be dynamically optimized using the new gradient method. On the one hand, this would allow the power plant to be designed under more realistic conditions (e.g. unevenness of the landscape can be taken into account). On the other hand, more fundamental questions can be asked, e.g. about the ideal heliostat shape, size or distribution in the field. The wide variance of heliostat models in the industry suggests that these questions remain unresolved to this day. With the joint optimization of these quantities, for example, dynamic relationships that have not been considered so far can be opened up. Of course, there are also circumstances that cannot be simulated by diff. ray tracing, such as material costs. Nevertheless, the ray tracing environment forms a physically more exact description of reality than common tools, so that features like material abrasion etc. could also be simulated.
Furthermore, since completely independent sizes can be optimized, this also allows completely new questions to be

asked. A field which was designed for a special aim point control is just as possible as a receiver shape matching the heliostat shape.

The NURBS provided a very good physical regularization so far. However, the NURBS basically do not know what a real heliostat looks like. *Prasad et al.*[109] could already show that NURBS can also be used as NN layers. The NURBS in diff. ray tracer could be preceded by NNs which were pretrained on measured real surfaces. In this case the NNs would act as a memory function to further regularize the surface. Thus, surfaces could be reconstructed that are still significantly more underdetermined.

6.4. Possible Future Applications for Neural Networks

Differential ray tracing has its limitations5.5. Depending on the level of detail of the surface and the distance of the heliostat to the tower a correct reconstruction can become impossible. The NURBS surfaces could show that even with nonphysical surfaces good results can be achieved, but in this area the whole approach of surface reconstruction is questionable. Instead, neural networks can be used to bypass this intermediate step by mapping directly from the sun position to the resulting focal spot. Previous approaches showed a rather large data requirement for this, but a solid pretraining (e.g. with very realistic looking

NURBS surfaces) where the heliostat learns the general mapping function and just has to re-map to the heliostat specific defects could significantly reduce the data requirement of these generative models.

The neural networks for heliostat calibration could also shine with their results in this work, even if the amount of data was rather too large. Nevertheless, this offers new approaches. So far this work has discussed exclusively in-situ methods, but *Carballo et al.*[54] have already successfully shown how well neural networks work when conditions are created especially for them.

A calibration process takes up to one minute. For a field of many thousands of heliostats, the data set per heliostat remains very small. However, the method can be improved by moving more heliostats to the target at the same time. The allocation of the focal spots as well as their segmentation would then be taken over by a neural network. With each additional heliostat that can be calibrated at the same time, the amount of available data is doubled. This solution is already patented and the implementation is in progress.

The heliostat calibration also needed so much data, because the task requires an extrapolation of the learned data. For example, if a heliostat is measured for an entire day, the input (sun array) data on the next day will be a little higher or lower in the elevation than the day

before. The more distant the new day is to the training data set, the stronger the extrapolation (this applies similarly to randomly chosen measurement points). However, a heliostat only moves to a certain number of motor positions within a year. With a (well adjusted) moving lamp on the heliostat, or an installed spotlight and a moving drone with a target, these outer motor positions can be steered to. With these motion profiles, even with tabular data, the network should be able to learn the required control function much faster, since all values can be intrapolated. Possibly a calibration of the heliostats can be omitted if such an initial measurement is done, albeit very precisely.

Finally, when talking about AI at a power plant, the issue of safety must always be discussed. The black box behavior can cause essential components of the power plant to have spontaneous dropouts, since their behavior cannot be predicted. Here, safety functions are necessary to control the output of the AI without affecting its speed. Again, differentiable ray tracing offers an ideal environment for this. On the one hand, trained models can be tested for their behavior in this environment under conditions that are as real as possible. On the other hand, the black box models can be run in parallel to white box models within this environment and thus they can be compared to each other, which provides a save operation.

Because one thing is certain: No matter whether in pre-

diction, diagnosis, simulation or control, the use of neural networks and generally AI methods at the solar tower power plant will increase dramatically in the coming years. This is because the possibilities of these algorithms are greater today than ever before. The differentiable ray tracing offers an excellent simulation environment for all upcoming methods and also acts independently as a digital power plant twin and hopefully lays the foundation for an efficient, autonomous and intelligent solar power plant.

Appendix

A. Used Parameters and Abbreviations of the Geometry Model

In table 6 all used parameters and abbreviations are listed. The function template used at the solar tower in Jülich assumes a model heliostat with 2 axes. Thereby each axis can move the suspension point ($\vec{O}_1$) by a constant vector ($\vec{v}_{12}$,$\vec{v}_{2M}$). Without rotations the suspension point (the mirror center) would be: $\vec{O}_M = \vec{O}_1 + \vec{v}_{12} + \vec{v}_{2M}$. Each rotation R is described by a rotation matrix depending on the angles $\alpha, \beta, \delta, \gamma, \tau_M, \theta_M$ and is acting on either the first or on both translation vectors. Where $\alpha, \beta, \delta, \gamma$ are constant angles of misalignment (which have to be determined by the regression algorithm) and τ_M, θ_M are the actual controllable alignments of the heliostat, measured in motor positions. Beside these rotational misalignments, for each axis the gear ratio ($GR1, GR2$) and an offset (τ_k, θ_k) are considered for the control parameters τ_M and θ_M. A distrinction is made between the Fixed Coordinate System (FCS), describing the position and the orientation of the pedestal of the heliostat and the Rotational Coordinate Systems (RCS1, RCS2) belonging to the movement of the first and second rotational axes.

Symbol	**Explanation**	**Degrees of Freedom**	**Dependencies**	**Explanation**
R_{01}	Alignment 1. axis (incl. tilting)	2 (rotation)	α	Fixed rotational displacement in FCS
			β	Fixed rotational displacement in FCS
R_θ	Rotation 1. axis	1 (rotation) +1 (gear ratio)	θ_M	Tracking angle azimuth. variable rotational displacement in RCS1
			θ_k	Axis offset in RCS1
			GR1	Gear ratio
$\vec{v}_{12}$	Distance between suspension point and axis 1	1 (translation)	-	-
R_{12}	Axis tilt 1. axis	1 (rotation)	γ	Fixed rotational displacement in RCS2
R_τ	Rotation 2. axis	1 (rotation) +1 (gear ratio)	τ_M	Tracking angle elevation. variable rotational displacement in RCS2
			τ_k	Axis offset in RCS2
			GR2	Gear ratio
$\vec{v}_{2M}$	Displacement between 2. axis and mirror center	1 (translation)	-	-
R_{2M}	Mirror tilt 2. axis	1 (rotation)	δ	Fixed rotational displacement in RCS2
$\vec{O}_1$	Heliostat suspension point (without translation)	3 (translation)	-	-
$\vec{O}_M$	Heliostat suspension point (mirror center)	3 (translation)	All of the above	-

Table 6: Matrices and vectors used for the function template underlying the Levenberg-Marquardt regression algorithm and abbreviations used in the text.

B. Optimization Potential of Existing Software

During this work, several opportunities for improvement of existing processes and methods have been identified and will be addressed here. They are only indirectly related to the presented methods, but should be considered e.g. in the further development of diff. ray tracing.

The former has already been mentioned in a footnote of this work. MLP and the geometry model both work with tabular data. These contain, among others, the center of the displaced focal spot. At the solar tower in Jülich this calculation takes place via a mean, but not a centroid determination. This increases the error of the calculation. Since the conclusion of this work is not to use tabular data in the future, this is a negligible change.

More serious is the next problem (which has also been mentioned in a footnote). Currently, there is no separation of the training and test sets at the tower. But this is elementary, because the accuracy of the training data set has no significance at all. Thus a new measurement can lead to incorrect geometry parameters being adapted. For the use of neural networks this problem is even more aggravated. For very small data sets a train/test ratio of 50:50 should be used, for growing data sets the test ratio can be reduced to 10%.

The next problems are related to the effective running time of the ray tracers used. While, shading and blocking were excluded in this work, for future implementations this will not be the case. Thus, this still has to be implemented. STRAL currently uses a bounding box based method to calculate the blocked/shaded rays. A method that is usually used for complex structures like leaves in trees to save computing power. For simple structures like heliostats this method is very computationally expensive. Here the use of so called shadow rays could speed up the process significantly and is therefore a good choice for the implementation in the diff. ray tracer.

The last problem concerns ray sampling which the diff. ray tracer has taken over from STRAL. Here the heliostat is discretized first. Randomly sampled (sun distributed) rays are then emitted at each discrete point. The rays are then distributed again at the receiver to the discrete pixels of the image. This method, which is based on Monte Carlo sampling, is used to solve the irradiance integral and is very efficient for whole scenes where the ray scattering is unknown. However, this precondition does not apply to the application at the solar tower. A better approach would be to start the rays at the image pixel, reflect them at the continuous heliostat surface and check if the light source was hit. Since the spatial direction of the heliostat is known from the pixel's point of view, the rays would only have to be emitted in an extremely small arc. With this linear discretization of space, the required number of

rays could be reduced from N to $\sqrt{N}$. This would be equivalent to an immense acceleration. STRAL is too advanced to make such fundamental changes. However, for the diff. ray tracing environment this is still a feasible way.

Parameter	STRAL	Diff. ray tracer
Longitude	6.36	6.36
Latitude	50.92	50.92
Height	0	0
Atmospheric Pressure	1.013 bar	N.A.
Ambient Temperature	20 °C	N.A.
Shading Objects	No	No
Receiver Pos ENU	[0,-3.23,35.89] m	[0,-3.23,35.89] m
Eval Plane	5 m	5 m
Heliostat Position	[13.2, 25.0, 1.795] m	[13.2, 25.0, 1.795] m
Bitmap Resolution	256x256	256x256
Sun Vectors	[-0.43719268, 0.7004466, 0.564125]	[-0.43719268, - 0.7004466, - 0.564125]
Sun Shape	Gaussian	Gaussian
Discrete Points on Heliostat	4096	4096
Enable Shading	No	N.A.
Enable Blocking	No	N.A.
Blocking by Shading Objects	No	N.A.
Enable self Shadowing	No	N.A.
Enable self Blocking	No	N.A.
Ray Multiplicator	100/1000	100/1000

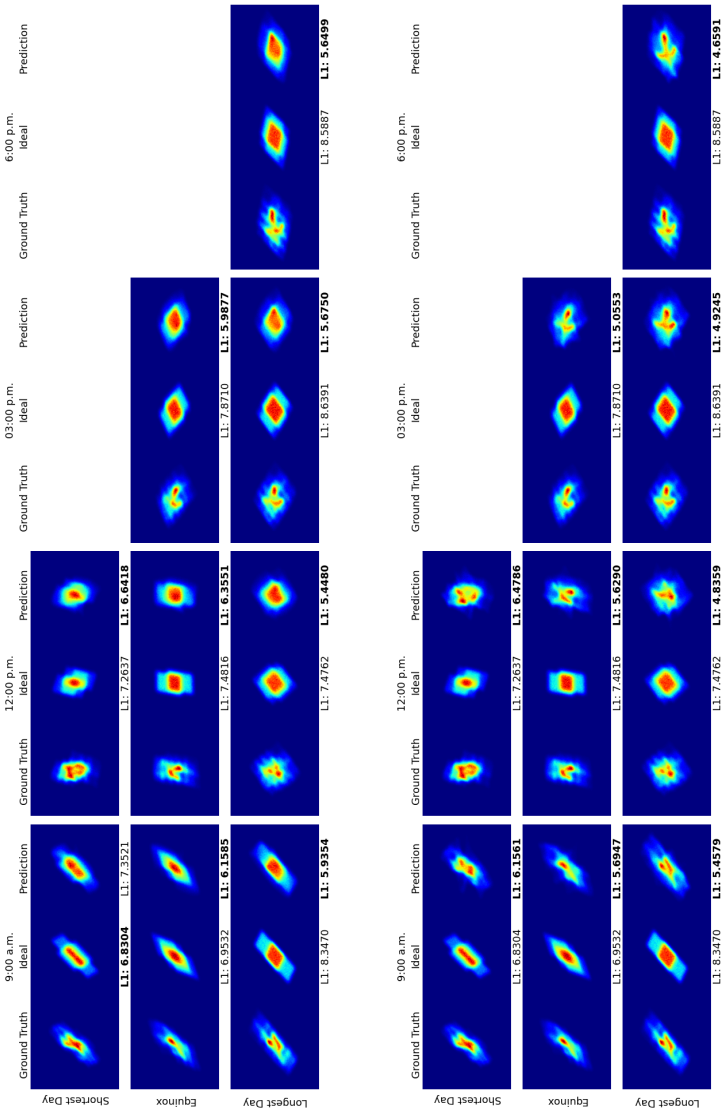
Table 7: Parameters in STRAL and diff. ray tracing for speed comparison. The ray multiplicator was changed for the 2 runtime simulations.

C. Diff. Ray Tracing and STRAL Runtime Comparison Parameters

Table 7 is a list of the parameters used to compare the runtimes of STRAL and the diff. ray tracer. The coordinate systems were adjusted from NWU to ENU. For a better comparability identical variables, which are named differently in the two programs, were also renamed here.

D. Additional Irradiance Maps and Surface Reconstructions

During the discussion of the flux density maps and the surface reconstructions, some surfaces or focal spots were not shown to ensure the reading flow. For the sake of completeness, they are presented here. In Fig. 46 flux density maps are shown from training with 1 to 5 images. The flux maps correspond to the yearly improvements shown in Fig. 43 and the shown surfaces in Fig. 40.



(a) Heliostat model trained on 1 image.

(b) Heliostat model trained on 5 images.

Figure 46: Comparison of deflectometric measured flux density profiles with measured, ideal and trained surfaces throughout the year.

E. References

[1] Kingma, Diederik ; Ba, Jimmy: Adam: A method for stochastic optimization. In: *arXiv preprint arXiv:1412.6980* (2014)

[2] Grieves, Michael: Digital twin: manufacturing excellence through virtual factory replication. In: *White paper* 1 (2014), S. 1–7

[3] Agreement, Paris: Paris agreement. In: *Report of the Conference of the Parties to the United Nations Framework Convention on Climate Change (21st Session, 2015: Paris). Retrived December* Bd. 4 HeinOnline, 2015, S. 2017

[4] Denholm, Paul ; O'Connell, Matthew ; Brinkman, Gregory ; Jorgenson, Jennie: Overgeneration from solar energy in california. a field guide to the duck chart. 2015. – Forschungsbericht

[5] *Deutsches Zentrum für Luft und Raumfahrt*

[6] Michalsky, Ronald ; Parman, Bryon ; Amanor-Boadu, Vincent ; Pfromm, Peter: Solar thermochemical production of ammonia from water, air and sunlight: Thermodynamic and economic analyses. In: *Energy* 42 (2012), Nr. 1, S. 251–260

[7] Pregger, Thomas ; Graf, Daniela ; Krewitt, Wolfram ; Sattler, Christian ; Roeb, Martin

; Möller, Stephan: Prospects of solar thermal hydrogen production processes. In: *International journal of hydrogen energy* 34 (2009), Nr. 10, S. 4256–4267

[8] Lovins, Amory: Decarbonizing Our Toughest Sectors-Profitably. In: *MIT Sloan Management Review* 63 (2021), Nr. 1, S. 46–55

[9] Zhongming, Zhu ; Linong, Lu ; Wangqiang, Zhang ; Wei, Liu u. a.: Solar power for sustainable pasta. (2020)

[10] Arvizu, Dan ; Balaya, Palani ; Cabeza, Luisa ; Hollands, Terry ; Waldau, Jochen ; Kondo, Michio ; Konseibo, Charles ; Meleshko, Valtentin ; Stein, Walter ; Tamaura, Yutaka u. a.: Direct solar energy. (2011)

[11] Schöniger, Franziska ; Thonig, Richard ; Resch, Gustav ; Lilliestam, Johan: Making the sun shine at night: comparing the cost of dispatchable concentrating solar power and photovoltaics with storage. In: *Energy Sources, Part B: Economics, Planning, and Policy* (2021), S. 1–20

[12] Edenhofer, Ottmar ; Pichs-Madruga, Ramón ; Sokona, Youba ; Seyboth, Kristin ; Kadner, Susanne ; Zwickel, Timm ; Eickemeier, Patrick ; Hansen, Gerrit ; Schlömer, Steffen ; Stechow, Christoph von u. a.: *Renewable energy sources and*

climate change mitigation: Special report of the intergovernmental panel on climate change. Cambridge University Press, 2011

[13] Osuna, Rafael ; Morillo, Rafael ; Sánchez, Marcelino ; Cantero, Felipe ; Fernández-Quero, Valerio ; Robles, Pedro ; Esteban, Antonio ; Céron, Francisco u. a.: PS10, Construction of a 11MW solar thermal tower plant in Seville, Spain. In: *Proc. 13th IEA SolarPACES Symp.*, 2006, S. A4–S3

[14] Viebahn, P. ; Zelt, O. ; Fischedick, M. ; Wietschel, M. ; Hirzel, S. ; Horst, J.: Technologien für die Energiewende: Technologiebericht - Band 1. Teilbericht 2 zum Teilprojekt A im Rahmen des strategischen BMWi-Leitprojekts; Trends und Perspektiven der Energieforschung. Version: 2018. `https://ideas.repec.org/p/zbw/wuprep/131.html`. 2018 (13.1). – Wuppertal Reports

[15] Pfahl, Andreas ; Coventry, Joe ; Röger, Marc ; Wolfertstetter, Fabian ; Vásquez-Arango, Juan F. ; Gross, Fabian ; Arjomandi, Maziar ; Schwarzbözl, Peter ; Geiger, Mark ; Liedke, Phillip: Progress in heliostat development. In: *Solar Energy* 152 (2017), S. 3–37

[16] Wells, Drew ; Young, Chris: *Ray tracing cre-*

ations generate 3-D photorealistic images on the PC. Waite Group Press, 1993

[17] Belhomme, Boris ; Pitz-Paal, Robert ; Schwarzbözl, Peter ; Ulmer, Steffen: A new fast ray tracing tool for high-precision simulation of heliostat fields. In: *Journal of Solar Energy Engineering* 131 (2009), Nr. 3

[18] Wendelin, Tim ; Dobos, Aron ; Lewandowski, Allan: SolTrace: a ray-tracing code for complex solar optical systems. 2013. – Forschungsbericht

[19] Yu, Qiang ; Wang, Zhifeng ; Xu, Ershu ; Zhang, Hongli ; Lu, Zhenwu ; Wei, Xiudong: Modeling and simulation of 1MWe solar tower plant's solar flux distribution on the central cavity receiver. In: *Simulation Modelling Practice and Theory* 29 (2012), 123–136. `http://dx.doi.org/https://doi.org/10.1016/j.simpat.2012.07.011`. – DOI https://-doi.org/10.1016/j.simpat.2012.07.011. – ISSN 1569–190X

[20] Cardoso, Amaia ; Marakkos, Costas ; Schoettl, Tiago Peter ; Les: New functionalities for the Tonatiuh ray-tracing software. In: *AIP Conference Proceedings* Bd. 2033 AIP Publishing LLC, 2018, S. 210010

[21] Cardona, Gabriel ; Pujol-Nadal, Ramon: OT-Sun, a python package for the optical analysis of

solar-thermal collectors and photovoltaic cells with arbitrary geometry. In: *Plos one* 15 (2020), Nr. 10, S. e0240735

[22] Branke, Raymond ; Heimsath, Anna: Raytrace3D power tower-A novel optical model for central receiver systems. In: *SolarPACES 2010, 16th Solar Power And Chemical Energy Systems International Symposium*, 2010

[23] Hanrieder, Natalie ; Wilbert, Stefan ; Mancera-Guevara, Daniel ; Buck, Reiner ; Giuliano, Stefano ; Pitz-Paal, Robert: Atmospheric extinction in solar tower plants–A review. In: *Solar Energy* 152 (2017), S. 193–207

[24] Guo, Minghuan ; Sun, Feihu ; Wang, Zhifeng: The backward ray tracing with effective solar brightness used to simulate the concentrated flux map of a solar tower concentrator. In: *AIP Conference Proceedings* Bd. 1850 AIP Publishing LLC, 2017, S. 030023

[25] Imenes, Anne ; Stein, Wes ; Hinkley, Jim ; Benito, Ralf ; Bolling, Ralf ; Schramek, Peter ; Ulmer, Steffen: Ray tracing and flux mapping as a design and research tool at the National Solar Energy Centre. In: *ANZSES 2006 Conference (Australia and New Zealand Solar Energy Society)*, 2006

[26] Oberkirsch, Laurin ; Quinto, Daniel M. ; Schwarzbözl, Peter ; Hoffschmidt, Bernhard: GPU-based aim point optimization for solar tower power plants. In: *Solar Energy* 220 (2021), S. 1089–1098

[27] Li, Tzu-Mao ; Aittala, Miika ; Durand, Frédo ; Lehtinen, Jaakko: Differentiable monte carlo ray tracing through edge sampling. In: *ACM Transactions on Graphics (TOG)* 37 (2018), Nr. 6, S. 1–11

[28] Nicolet, Baptiste ; Jacobson, Alec ; Jakob, Wenzel: Large steps in inverse rendering of geometry. In: *ACM Transactions on Graphics (TOG)* 40 (2021), Nr. 6, S. 1–13

[29] Loubet, Guillaume ; Holzschuch, Nicolas ; Jakob, Wenzel: Reparameterizing discontinuous integrands for differentiable rendering. In: *ACM Transactions on Graphics (TOG)* 38 (2019), Nr. 6, S. 1–14

[30] Kato, Hiroharu ; Beker, Deniz ; Morariu, Mihai ; Ando, Takahiro ; Matsuoka, Toru ; Kehl, Wadim ; Gaidon, Adrien: Differentiable rendering: A survey. In: *arXiv preprint* (2020)

[31] Sun, Qilin ; Wang, Congli ; Qiang, Fu ; Xiong, Dun ; Wolfgang, Heidrich: End-to-end complex lens design with differentiable ray tracing. In: *ACM Transactions on Graphics* 40 (2021), Nr. 4, S. 1–13

[32] Jain, Lakshya ; Wu, Wilson ; Chen, Steven ; Jang, Uyeong ; Chandrasekaran, Varun ; Seshia, Sanjit A. ; Jha, Somesh: Generating Semantic Adversarial Examples with Differentiable Rendering. In: *CoRR* abs/1910.00727 (2019). http://dblp.uni-trier.de/db/journals/corr/corr1910.html#abs-1910-00727

[33] Tu, James ; Li, Huichen ; Yan, Xinchen ; Ren, Mengye ; Chen, Yun ; Liang, Ming ; Bitar, Eilyan ; Yumer, Ersin ; Urtasun, Raquel: *Exploring Adversarial Robustness of Multi-Sensor Perception Systems in Self Driving*. http://dx.doi.org/10.48550/ARXIV.2101.06784. Version: 2021

[34] Noone, Corey J. ; Torrilhon, Manuel ; Mitsos, Alexander: Heliostat field optimization: A new computationally efficient model and biomimetic layout. In: *Solar Energy* 86 (2012), Nr. 2, 792-803. http://dx.doi.org/https://doi.org/10.1016/j.solener.2011.12.007. – DOI https://-doi.org/10.1016/j.solener.2011.12.007. – ISSN 0038–092X

[35] May, Oscar ; Ricalde, Luis J. ; Ali, Bassam ; López, E O. ; Venegas-Reyes, Eduardo ; Jaramillo, Oscar A.: Neural network inverse modeling for optimization. In: *Artificial Neural Networks-Models and Applications* Bd. 1. InTech Rijeka, 2016, S. 168–176

[36] Asselineau, C.-A. ; Zapata, J. ; Pye, J.: Geometrical Shape Optimization of a Cavity Receiver Using Coupled Radiative and Hydrodynamic Modeling. In: *Energy Procedia* 69 (2015), 279-288. http://dx.doi.org/https://doi.org/10.1016/j.egypro.2015.03.032. – DOI https://-doi.org/10.1016/j.egypro.2015.03.032. – ISSN 1876–6102. – International Conference on Concentrating Solar Power and Chemical Energy Systems, SolarPACES 2014

[37] Luo, Lei ; Du, Wei ; Wang, Songtao ; Wang, Lei ; Sundén, Bengt ; Zhang, Xinhong: Multi-objective optimization of a solar receiver considering both the dimple/protrusion depth and delta-winglet vortex generators. In: *Energy* 137 (2017), S. 1–19

[38] Stone, Kenneth: *Automatic heliostat track alignment method*. Januar 14 1986. – US Patent 4,564,275

[39] Pargmann, Max ; Maldonado Quinto, Daniel ; Schwarzbözl, Peter ; Pitz-Paal, Robert: High accuracy data-driven heliostat calibration and state prediction with pretrained deep neural networks. In: *Solar Energy* 218 (2021), 48–56. http://dx.doi.org/https://doi.org/10.1016/j.solener.2021.01.046. – DOI https://doi.org/10.1016/j.solener.2021.01.046. – ISSN 0038–092X

[40] Sattler, Johannes C. ; Röger, Marc ; Schwarzbözl, Peter ; Buck, Reiner ; Macke, Ansgar ; Raeder, Christian ; Göttsche, Joachim: Review of heliostat calibration and tracking control methods. In: *Solar Energy* 207 (2020), S. 110–132

[41] Xie, Qizhe ; Luong, Minh-Thang ; Hovy, Eduard ; Le, Quoc: Self-training with noisy student improves imagenet classification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, S. 10687–10698

[42] Feng, Shuo ; Zhou, Huiyu ; Dong, Hongbiao: Using deep neural network with small dataset to predict material defects. In: *Materials & Design* 162 (2019), S. 300–310

[43] Pidhorskyi, Stanislav ; Adjeroh, Donald ; Doretto, Gianfranco: Adversarial Latent Autoencoders. In: *arXiv:2004.04467v1* (2020)

[44] Munkberg, Jacob ; Hasselgren, Jon ; Shen, Tianchang ; Gao, Jun ; Chen, Wenzheng ; Evans, Alex ; Müller, Thomas ; Fidler, Sanja: Extracting Triangular 3D Models, Materials, and Lighting From Images. In: *arXiv preprint arXiv:2111.12503* (2021)

[45] Mnih, Volodymyr ; Kavukcuoglu, Koray ; Silver, David ; Rusu, Andrei ; Veness, Joel ;

Bellemare, Marc ; Graves, Alex ; Riedmiller, Martin ; Fidjeland, Andreas ; Ostrovski, Georg u. a.: Human-level control through deep reinforcement learning. In: *nature* 518 (2015), Nr. 7540, S. 529–533

[46] Mandhane, Amol ; Zhernov, Anton ; Rauh, Maribeth ; Gu, Chenjie ; Wang, Miaosen ; Xue, Flora ; Shang, Wendy ; Pang, Derek ; Claus, Rene ; Chiang, Ching-Han u. a.: MuZero with Self-competition for Rate Control in VP9 Video Compression. In: *arXiv preprint arXiv:2202.06626* (2022)

[47] Fast, Magnus ; Palme, Tim: Application of artificial neural networks to the condition monitoring and diagnosis of a combined heat and power plant. In: *Energy* 35 (2010), Nr. 2, S. 1114–1120

[48] Bettocchi, Roberto ; Pinelli, Michele ; Spina, Pier R. ; Venturini, Mauro: Artificial intelligence for the diagnostics of gas turbines—Part I: Neural network approach. (2007)

[49] Nowak, Grzegorz ; Rusin, Andrzej: Using the artificial neural network to control the steam turbine heating process. In: *Applied Thermal Engineering* 108 (2016), 204–210. `http://dx.doi.org/https://doi.org/10.1016/j.applthermaleng.2016.07.129`. – DOI https://-

doi.org/10.1016/j.applthermaleng.2016.07.129. – ISSN 1359–4311

[50] Fischer, Stephan ; Frey, Patrick ; Drück, Harald: A comparison between state-of-the-art and neural network modelling of solar collectors. In: *Solar energy* 86 (2012), Nr. 11, S. 3268–3277

[51] Söylemez, Mehmet ; Mohammed Al Dulaimi, Raad: An Innovative Optimization Algorithm for Precis Estimation of the Optical Efficiency of Heliostat Fields. 6 (2016), Juni

[52] Richter, Pascal ; Ábrahám, Erika ; Morin, Gabriel: Optimisation of concentrating solar thermal power plants with neural networks. In: *International Conference on Adaptive and Natural Computing Algorithms* Springer, 2011, S. 190–199

[53] González-Tokman, Mariano ; Ávila-Miranda, Raúl ; Sánchez, Edgar: Neural control for a field of concentrator heliostats. In: *2014 World Automation Congress (WAC)* IEEE, 2014, S. 670–674

[54] Carballo, Jose A. ; Bonilla, Javier ; Berenguel, Manuel ; Fernández-Reche, Jesús ; García, Ginés: New approach for solar tracking systems based on computer vision, low cost hardware and deep learning. In: *Renewable Energy* 133 (2019), 1158–1166. `http://dx.doi.org/https://doi.org/10.1016/j.renene.2018.08.101`. –

DOI https://doi.org/10.1016/j.renene.2018.08.101. – ISSN 0960–1481

[55] Mildenhall, Ben ; Srinivasan, Pratul P. ; Tancik, Matthew ; Barron, Jonathan T. ; Ramamoorthi, Ravi ; Ng, Ren: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In: *CoRR* abs/2003.08934 (2020). https://arxiv.org/abs/2003.08934

[56] Offergeld, Matthias ; Röger, Marc ; Stadler, Hannes ; Gorzalka, Philip ; Hoffschmidt, Bernhard: Flux density measurement for industrial-scale solar power towers using the reflection off the absorber. In: *AIP Conference Proceedings* Bd. 2126 AIP Publishing LLC, 2019, S. 110002

[57] Wang, Zhifeng: Chapter 1 - Introduction. Version: 2019. http://dx.doi.org/https://doi.org/10.1016/B978-0-12-815613-1.00001-8. In: Wang, Zhifeng (Hrsg.): *Design of Solar Thermal Power Plants*. Academic Press, 2019. – DOI https://doi.org/10.1016/B978–0–12–815613–1.00001–8. – ISBN 978–0–12–815613–1, 1–46

[58] Dabiri, Soroush ; Rahimi, Mohammad F.: Basic introduction of solar collectors and energy and exergy analysis of a heliostat plant. In: *3rd International Conference and Exhibition on Solar Energy*

ICESE-2016. Tehran, Iran: University of Tehran, 2016

[59] Yağli, Hüseyin: Examining the receiver heat loss, parametric optimization and exergy analysis of a solar power tower (SPT) system. In: *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 42 (2020), Nr. 17, S. 2155–2180

[60] Raeder, Christian ; Offergeld, Matthias ; Röger, Marc ; Lademann, Alexander ; Zöller, Judith ; Glinka, Matthias ; Escamilla, José ; Kämpgen, Andreas: Proof of Concept: Real-Time Flux Density Monitoring System on External Tube Receivers for Optimized Solar Field Operation. (2021)

[61] Monterreal, Ralf ; Neumann, Adrian: Advanced flux measurement system for solar tower plants. In: *Proc. of Seventh Intern. Symposium on Solar Thermal Concentrating Technologies, Moscow*, 1994

[62] Ballestrín, Jan ; Monterreal, Ralf: Hybrid heat flux measurement system for solar central receiver evaluation. In: *Energy* 29 (2004), Nr. 5-6, S. 915–924

[63] Göhring, Felix ; Bender, Olaf ; Röger, Marc ; Nettlau, Janina ; Schwarzbözl, Peter: Flux density measurement on open volumetric receivers. In: *Proc. of SolarPACES 2011* (2011)

[64] Pacheco, James ; Houser, Ralf ; Neumann, Anton: Concepts to measure flux and temperature for external central receivers / Sandia National Labs., Albuquerque, NM (United States). 1993. – Forschungsbericht

[65] Röger, Marc ; Herrmann, Patrik ; Ebert, Miriam ; Prahl, Christoph ; Ulmer, Steffen ; Göhring, Felix: Flux Density Measurement on Large-Scale Receivers. In: *SolarPACES 2011*, 2011

[66] He, Ya-Ling ; Cui, Fu-Qing ; Cheng, Ze-Dong ; Li, Zeng-Yao ; Tao, Wen-Quan: Numerical simulation of solar radiation transmission process for the solar tower power plant: from the heliostat field to the pressurized volumetric receiver. In: *Applied thermal engineering* 61 (2013), Nr. 2, S. 583–595

[67] Reda, Ibrahim ; Andreas, Afshin: Solar position algorithm for solar radiation applications. In: *Solar energy* 76 (2004), Nr. 5, S. 577–589

[68] Pena-Cruz, Manuel ; Arancibia-Bulnes, Camilo ; Iriarte-Cornejo, Cuitlahuac ; Cabanillas, Rafael ; Estrada, Claudio: Heliostat characterization by optical techniques and image processing. In: *Energy Sustainability* Bd. 44816 American Society of Mechanical Engineers, 2012, S. 353–362

[69] Hagenkamp, Markus: *Trackingverbesserung eines*

Heliostatenfeldes durch Optimierung von Kalibrierverfahren, RWTH Aachen, Diplomarbeit, März 2018. https://elib.dlr.de/119566/

[70] Gross, William: *Closed loop tracking system using signal beam.* April 21 2015. – US Patent 9,010,317

[71] Flesch, Robert ; Ahlbrink, Nils ; Pitz-Paal, Robert ; Belhomme, Boris: Automatic determination of heliostat orientation using an auxiliary mirror. In: *Solarpaces 2012*, 2012

[72] Bern, Gregor ; Schöttl, Peter ; Heimsath, Anna ; Nitz, Peter: Novel imaging closed loop control strategy for heliostats. In: *AIP Conference Proceedings* Bd. 1850 AIP Publishing LLC, 2017, S. 030005

[73] Zavodny, M. ; Slack, M. ; Huibregtse, R. ; Sonn, A.: Tower-based CSP Artificial Light Calibration System. In: *Energy Procedia* 69 (2015), 1488–1497. http://dx.doi.org/https://doi.org/10.1016/j.egypro.2015.03.098. – DOI https://doi.org/10.1016/j.egypro.2015.03.098. – ISSN 1876–6102. – International Conference on Concentrating Solar Power and Chemical Energy Systems, SolarPACES 2014

[74] Hines: *Heliostat characterization using starlight.* 2017. – US Patent US20180299264A1

[75] Hoffschmidt, Bernhard: HelioScan: Machbarkeitsstudie zur Entwicklung einer

radargestützten Positionsregelung von Heliostatenfeldern für Solarturmkraftwerke: Schlussbericht: Laufzeit: 01.07. 2010-31.12. 2012: Förderkennzeichen 325234A.

[76] Gouws, Jurgens E.: *Calibration of heliostats using a drone.*, Stellenbosch: Stellenbosch University, Diss., 2018

[77] Fachinotti, Vincent ; Anca, Amelia ; Cardona, Anton: A method for the solution of certain problems in least squares. In: *Int J Numer Method Biomed Eng* 27 (2011), Nr. 4, S. 595–607

[78] Zimmermann, Felix: *Evaluation and Optimization of Heliostat Calibration Methods*, RWTH Aachen University, Diss., 2021

[79] Khalsa, Salasar ; Ho, Chick ; Andraka, Ceck: An automated method to correct heliostat tracking errors. In: *Proceedings of SolarPACES* (2011), S. 20–23

[80] März, Tobias ; Prahl, Christoph ; Ulmer, Steffen ; Wilbert, Stefan ; Weber, Christian: Validation of two optical measurement methods for the qualification of the shape accuracy of mirror panels for concentrating solar systems. In: *Journal of Solar Energy Engineering* 133 (2011), Nr. 3

[81] Rapp, Holger: *Reconstruction of specular reflective surfaces using auto-calibrating deflectometry*. Bd. 23. KIT Scientific Publishing, 2013

[82] Ulmer, Steffen ; März, Tobias ; Prahl, Christoph ; Reinalter, Wolfgang ; Belhomme, Boris: Automated high resolution measurement of heliostat slope errors. In: *Solar Energy* 85 (2011), Nr. 4, S. 681–687

[83] Judge, Tim ; Bryanston-Cross, Paul: A review of phase unwrapping techniques in fringe analysis. In: *Optics and Lasers in Engineering* 21 (1994), Nr. 4, 199–239. http://dx.doi.org/https://doi.org/10.1016/0143-8166(94)90073-6. – DOI https://doi.org/10.1016/0143–8166(94)90073–6. – ISSN 0143–8166

[84] Immel, David ; Cohen, Michael ; Greenberg, Donald: A radiosity method for non-diffuse environments. In: *Acm Siggraph Computer Graphics* 20 (1986), Nr. 4, S. 133–142

[85] Cook, Robert ; Porter, Thomas ; Carpenter, Loren: Distributed ray tracing. In: *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 1984, S. 137–145

[86] Belhomme, Boris: *Bewertung und Optimierung von Zielpunktstrategien für solare Turmkraftwerke*. Shaker, 2011

[87] Lu, Zhou ; Pu, Hongming ; Wang, Feicheng ; Hu, Zhiqiang ; Wang, Liwei: The expressive power of neural networks: A view from the width. In: *Advances in neural information processing systems*, 2017, S. 6231–6239

[88] Kelp, Makoto ; Tessum, Christopher ; Marshall, Julian: Orders-of-magnitude speedup in atmospheric chemistry modeling through neural network-based emulation. In: *arXiv preprint arXiv:1808.03874* (2018)

[89] Davis, Abe ; Rubinstein, Michael ; Wadhwa, Neal ; Mysore, Gautham ; Durand, Fredo ; Freeman, William: The visual microphone: Passive recovery of sound from video. (2014)

[90] Barry-Straume, Jostein ; Tschannen, Adam ; Engels, Daniel ; Fine, Edward: An evaluation of training size impact on validation accuracy for optimized convolutional neural networks. In: *SMU Data Science Review* 1 (2018), Nr. 4, S. 12

[91] LeCun, Yann ; Bottou, Léon ; Bengio, Yoshua ; Haffner, Patrick: Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* 86 (1998), Nr. 11, S. 2278–2324

[92] Kadra, Arlind ; Lindauer, Marius ; Hutter, Frank ; Grabocka, Josif: Regularization is all you need: Simple neural nets can excel on tabular data. In: *arXiv preprint arXiv:2106.11189* (2021)

[93] Kolbusz, Janusz ; Rozycki, Pawel ; Wilamowski, Bogdan: The study of architecture MLP with linear neurons in order to eliminate the "vanishing gradient" problem. In: *International Conference on Artificial Intelligence and Soft Computing* Springer, 2017, S. 97–106

[94] LeCun, Yan ; Bengio, Yan ; Hinton, Gustav: Deep learning. In: *Nature* 521 (2015), S. 436–444

[95] Klambauer, Günter ; Unterthiner, Thomas ; Mayr, Andreas ; Hochreiter, Sepp: Self-Normalizing Neural Networks. In: *CoRR* abs/1706.02515 (2017). `http://arxiv.org/abs/1706.02515`

[96] Kajiya, James: The rendering equation. In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, S. 143–150

[97] Martin-Brualla, Ricardo ; Radwan, Noha ; Sajjadi, Mehdi S. M. ; Barron, Jonathan T. ; Dosovitskiy, Alexey ; Duckworth, Daniel: NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: *CoRR* abs/2008.02268 (2020). `https://arxiv.org/abs/2008.02268`

[98] Pumarola, Albert ; Corona, Enric ; Pons-Moll, Gerard ; Moreno-Noguer, Francesc: D-nerf: Neural radiance fields for dynamic scenes.

In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, S. 10318–10327

[99] Tretschk, Edgar ; Tewari, Ayush ; Golyanik, Vladislav ; Zollhöfer, Michael ; Lassner, Christoph ; Theobalt, Christian: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, S. 12959–12970

[100] Jain, Ajay ; Tancik, Matthew ; Abbeel, Pieter: Putting NeRF on a Diet: Semantically Consistent Few-Shot View Synthesis. In: *CoRR* abs/2104.00677 (2021). https://arxiv.org/abs/2104.00677

[101] Müller, Thomas ; Rousselle, Fabrice ; Novák, Jan ; Keller, Alexander: Real-time neural radiance caching for path tracing. (2021), Juli, 1–16. http://dx.doi.org/10.1145/3450626.3459812. – DOI 10.1145/3450626.3459812

[102] Müller, Thomas ; Evans, Alex ; Schied, Christoph ; Keller, Alexander: Instant neural graphics primitives with a multiresolution hash encoding. In: *arXiv preprint arXiv:2201.05989* (2022)

[103] Erhan, Dumitru ; Courville, Aaron ; Bengio,

Yoshua ; Vincent, Pascal: Why Does Unsupervised Pre-training Help Deep Learning? In: Teh, Yee W. (Hrsg.) ; Titterington, Mike (Hrsg.): *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* Bd. 9. Chia Laguna Resort, Sardinia, Italy : PMLR, 2010 (Proceedings of Machine Learning Research), 201–208

[104] Aboutalib, Sarah ; Mohamed, Aly ; Berg, Wendie ; Zuley, Margarita ; Sumkin, Jules ; Wu, Shandong: Deep learning to distinguish recalled but benign mammography images in breast cancer screening. In: *Clinical Cancer Research* 24 (2018), Nr. 23, S. 5902–5909

[105] Rojas, Raul: The backpropagation algorithm. In: *Neural networks*. Springer, 1996, S. 149–182

[106] Robert, Hecht-Nielsen u. a.: Theory of the backpropagation neural network. In: *Proc. 1989 IEEE IJCNN* 1 (1989), S. 593–605

[107] Wang, Yangyang ; Zhang, Yaqin ; Huang, Meiyu ; Chen, Zhao ; Jia, Yi ; Weng, Yudong ; Xiao, Lin ; Xiang, Xueshuang: Accurate but fragile passive non-line-of-sight recognition. In: *Communications Physics* 4 (2021), Nr. 1, S. 1–9

[108] Ahlrichs, Patrick ; Dünweg, Burkhard: Lattice-Boltzmann simulation of polymer-solvent systems.

In: *International Journal of Modern Physics C* 9 (1998), Nr. 08, S. 1429–1438

[109] Prasad, Anjana D. ; Balu, Aditya ; Shah, Harshil ; Sarkar, Soumik ; Hegde, Chinmay ; Krishnamurthy, Adarsh: NURBS-Diff: A Differentiable Programming Module For NURBS. In: *Computer-Aided Design* (2022), S. 103199

[110] De Boor, Carl ; De Boor, Carl: *A practical guide to splines*. Bd. 27. springer-verlag New York, 1978

[111] Quigley, Morgan ; Conley, Ken ; Gerkey, Brian ; Faust, Josh ; Foote, Tully ; Leibs, Jeremy ; Wheeler, Rob ; Ng, Andrew u. a.: ROS: an open-source Robot Operating System. In: *ICRA workshop on open source software* Bd. 3 Kobe, Japan, 2009, S. 5

[112] Blanco, Manuel: *Tonatiuh: An object oriented, distributed computing, Monte-Carlo ray tracer for the design and simulation of solar concentrating systems*. 2016

[113] Jafrancesco, David ; Cardoso, Joao ; Mutuberria, Amaia ; Leonardi, Erminia ; Les, Inigo ; Sansoni, Paola ; Francini, Franco ; Fontani, Daniela: Optical simulation of a central receiver system: Comparison of different software tools. In: *Renewable and Sustainable Energy Reviews* 94 (2018), 792-803. `http://dx.doi.org/https:`

//doi.org/10.1016/j.rser.2018.06.028. – DOI https://doi.org/10.1016/j.rser.2018.06.028. – ISSN 1364–0321

[114] Horta, Pedro ; Larcher, Marco ; Hertel, Julian ; Rooyen, De W. ; Heimsath, Anna ; Schneider, Simon ; Benitez, Daniel ; Frein, Antoine ; Denarie, Alice: Ray-Tracing Software Comparison for Linear Focusing Solar Collectors. In: *AIP Conference Proceedings* 1734 (2016), 05

[115] Mitchell, Don ; Netravali, Arun: Reconstruction filters in computer-graphics. In: *ACM Siggraph Computer Graphics* 22 (1988), Nr. 4, S. 221–228

[116] Yu, Hao ; Wilamowski, Bogdan: Levenberg-marquardt training. In: *Industrial electronics handbook* 5 (2011), Nr. 12, S. 1

[117] Suratgar, Amir A. ; Tavakoli, Mohammad B. ; Hoseinabadi, Abbas: Modified Levenberg-Marquardt method for neural networks training. In: *World Acad Sci Eng Technol* 6 (2005), Nr. 1, S. 46–48

[118] Reynaldi, Arnold ; Lukas, Samuel ; Margaretha, Helena: Backpropagation and Levenberg-Marquardt algorithm for training finite element neural network. In: *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation* IEEE, 2012, S. 89–94

[119] Yosinski, Jason ; Clune, Jeff ; Bengio, Yoshua ; Lipson, Hod: How transferable are features in deep neural networks? In: *Advances in neural information processing systems*, 2014, S. 3320–3328

[120] Peng, Alex Y. ; Koh, Yun S. ; Riddle, Patricia ; Pfahringer, Bernhard: Using supervised pretraining to improve generalization of neural networks on binary classification problems. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* Springer, 2018, S. 410–425

[121] Tancik, Matthew ; Srinivasan, Pratul P. ; Mildenhall, Ben ; Fridovich-Keil, Sara ; Raghavan, Nithin ; Singhal, Utkarsh ; Ramamoorthi, Ravi ; Barron, Jonathan T. ; Ng, Ren: Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In: Larochelle, Hugo (Hrsg.) ; Ranzato, Marc'Aurelio (Hrsg.) ; Hadsell, Raia (Hrsg.) ; Balcan, Maria-Florina (Hrsg.) ; Lin, Hsuan-Tien (Hrsg.): *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020

[122] Nieffer, Daniel ; Effertz, Timo ; Macke, Ansgar ; Röger, Marc ; Weinrebe, Gerhard ; Ulmer, Steffen: Heliostat testing according to SolarPACES task III guideline. In: *AIP Conference*

Proceedings Bd. 2126 AIP Publishing LLC, 2019, S. 030039

[123] Upton, Graham ; Cook, Ian: *A dictionary of statistics 3e*. Oxford quick reference, 2014

[124] Huber, Peter: Robust estimation of a location parameter. In: *Breakthroughs in statistics*. Springer, 1992, S. 492–518

[125] Röger, Marc ; Prahl, Christoph ; Ulmer, Steffen: Fast determination of heliostat shape and orientation by edge detection and photogrammetry. In: *Proc. 14th CSP SolarPACES Symposium 2008* SolarPACES, 2008

[126] Prahl, Christoph ; Röger, Marc ; Kiewitt, Wilhelm: Advances in Optical Measurement Techniques for Solar Concentrators. In: *SolarPACES 2009*, SolarPACES, 2009

[127] Kribus, Abraham ; Vishnevetsky, Irina ; Yogev, Amnon ; Rubinov, Tatiana: Closed loop control of heliostats. In: *Energy* 29 (2004), Nr. 5, 905–913. `http://dx.doi.org/https://doi.org/10.1016/S0360-5442(03)00195-6`. – DOI https://doi.org/10.1016/S0360–5442(03)00195–6. – ISSN 0360–5442. – SolarPACES 2002

[128] Schroff, Florian ; Kalenichenko, Dmitry ; Philbin, James: Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of*

the IEEE conference on computer vision and pattern recognition, 2015, S. 815–823

[129] Chen, Weihua ; Chen, Xiaotang ; Zhang, Jianguo ; Huang, Kaiqi: Beyond triplet loss: a deep quadruplet network for person re-identification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, S. 403–412

[130] Ribera, Javier ; Güera, David ; Chen, Yuhao ; Delp, Edward: Weighted hausdorff distance: A loss function for object localization. In: *arXiv preprint arXiv:1806.07564* 2 (2018), S. 1

[131] Pargmann, Max ; Ebert, Jan ; Quinto, Daniel M. ; Pitz-Paal, Robert ; Kesselheim, Stefan: In-Situ Solar Tower Power Plant Optimization byDifferentiable Ray Tracing / In Review. Version: Februar 2023. `http://dx.doi.org/10.21203/rs.3.rs-2554998/v1`. 2023. – preprint

[132] Pargmann, Max ; Leibauer, Moritz ; Nettelroth, Vincent ; Maldonado Quinto, Daniel ; Pitz-Paal, Robert: Enhancing heliostat calibration on low data by fusing robotic rigid body kinematics with neural networks. In: *Solar Energy* 264 (2023), 111962. `http://dx.doi.org/https://doi.org/10.1016/j.solener.2023.111962`. – DOI https://doi.org/10.1016/j.solener.2023.111962. – ISSN 0038–092X