SolarPACES 2022, 28th International Conference on Concentrating Solar Power and Chemical Energy Systems
Analysis and Simulation of CSP and Hybridized Systems
https://doi.org/10.52825/solarpaces.v1i.642

Published: 09 Jan. 2024

In Situ Enhancement of Heliostat Calibration Using Differentiable Ray Tracing and Artificial Intelligence

Max Pargmann[1][https://orcid.org/0000-0002-4705-6285], Jan Ebert[2][https://orcid.org/0000-0001-7118-0481], Stefan Kesselheim[2][https://orcid.org/0000-0003-0940-5752], Daniel Maldonado Quinto[1][https://orcid.org/0000-0003-2929-8667], and Robert Pitz-Paal[1][https://orcid.org/0000-0002-3542-3391]

[1] Institute of Solar Research, German Aerospace Center (DLR), Linder Hoehe, 51147 Cologne,Germany

[2] Jülich Supercomputing Centre, Forschungszentrum Jülich (FZJ), Wilhelm-Johnen-Straße, 52428 Jülich, Germany

Abstract. The camera target method is the most commonly used calibration method for heliostats at solar tower power plants to minimize their sun tracking errors. In this method, individual heliostats are moved to a white surface and their deviation from the targeted position is measured. A regression is used to calculate errors in a geometry model from the tabular data obtained in this way. For modern aim point strategies, or simply heliostats in the rearmost end of the field, extremely high accuracies are needed, which can only be achieved by many degrees of freedom in the geometry model. The problem here is that the camera target method produces only a very small data set per heliostat, which limits the number of free variables and thus the accuracy. In this work, we extend existing ray tracing methods for solar towers with a differentiable description, allowing for the first time a data-driven optimization of object parameters within the ray tracing environment. Therefore, the heliostat calibration can take place directly within the ray tracing environment. Thus, the image data acquired during the measurement can be processed directly and more information about the orientation of the heliostat can be obtained. Within a simple example we show the advantages of the method, which converges faster and corrects errors that could not be considered before. Without any disadvantages or additional costs, the state-of-the-art calibration method can be improved.

Keywords: Heliostat Calibration, Differentiable Raytracing, Machine Learning

1. Introduction

The performance and levelized cost of energy of a solar tower power plant are directly linked to the functionality of its heliostats. The better the heliostats can redirect the sun to the receiver, the more power can be generated. In contrast, incorrect alignment not only results in lower energy output, but can cause temperature spikes or gradients and compromise the longevity of the components. The most frequently used calibration (cal.) method, to correct the heliostat specific errors at commercial power plants is the Camera-Target method. In this method, a single focal spot is moved from the receiver to a white target below. From the sun and heliostat position, as well as the difference between the aimed and actual aiming point of the focal spot, the errors of the heliostat can be determined by means of mathematical regression. The method is used because it can be fully automated, reliable and quite accurate for a certain period of time after the measurement. The biggest weakness of the method is the time needed per measurement. With about 60 seconds per measurement the (tabular) data set of a single heliostat rarely grows by more than a few data points per year. The measurement should also not be done frequently, as the process itself decreases the power on the receiver. In order to describe the heliostat completely with all possible errors (rotation, distortion, displacement,

partly with angle and time dependencies) it needs more degrees of freedom in the regression but the data situation including just pre-extracted features of the focal spot is not sufficient. Thus, the accuracy is not feasible for distant heliostats or modern aimpoint control strategies. There are many other heliostat calibration methods [1] but any new method must measure up to this standard procedure. An improvement of the Camera-Target method raises the benchmark for all other methods.

Today, ray tracing is already a central component of solar tower power plants. It is used for field design, flux density prediction and target point strategies. For this purpose, all information known about the solar region of the power plant is collected and from this a 2D image of the flux density is calculated. However, obtaining the data of the field is a complex process, partly with a high error propagation.

Over the last few years, ray tracers have taken over the data processing themselves. So-called inverse rendering, of which differentiable (diff.) ray tracing is a subcategory, can derive 3-dimensional quantities, such as shape [2], material property, or both [3] from 2D images. In contrast to classical ray tracing, additionally to just calculating the rays directions, partial derivatives of all ray interactions are stored inside a mathematical graph, which allows the use of gradient-based optimization and backpropagation as well as to be integrated into AI optimization routines. The difficulty of this method lies in the differentiable description of the scene to be examined. These techniques became more sophisticated and now make whole scenes differentiable [4] or enable new sources of illumination [5]. The range of tools is very wide [6], but only a few of these tools are used industrially. Main applications are in lens design [7] or some test cases in autonomous driving [8][9]. In fact, these examples are rare, but already show that diff. ray tracing can optimize the data supply chain while accurately reconstructing important object properties. Nevertheless, as far as known to the authors, a theoretical consideration or even a deployment in (any) thermal solar power plants has not taken place yet.

This work proposes an in situ improvement of Camera-Target regression using differentiable ray tracing. The direct use of image data increases the information content of each calibration, which allows faster convergence to the global optimum and even to correct errors which have been completely neglected so far, e.g. rotational displacements, canting or focusing. At the same time, diff. ray tracing allows a physics-regulated optimization, which provides better gradients than common methods. This not only reduces the number of measurements required, it also allows optimization of previously neglected parameters such as canting, focusing, etc.

To show this, the principles of the state-of-the-art calibration will be presented in the following. Then the integration of diff. ray tracing into the solar power plant process will be explained and how the scenery at the solar tower can be made differentiable. Both methods are then compared before the results are demonstrated and discussed on a minimal example.

2. Heliostat Field Control and Calibration

Heliostat calibration is carried out regularly at solar towers to counteract individual heliostat errors, which affect 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% [10]. 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. But 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.

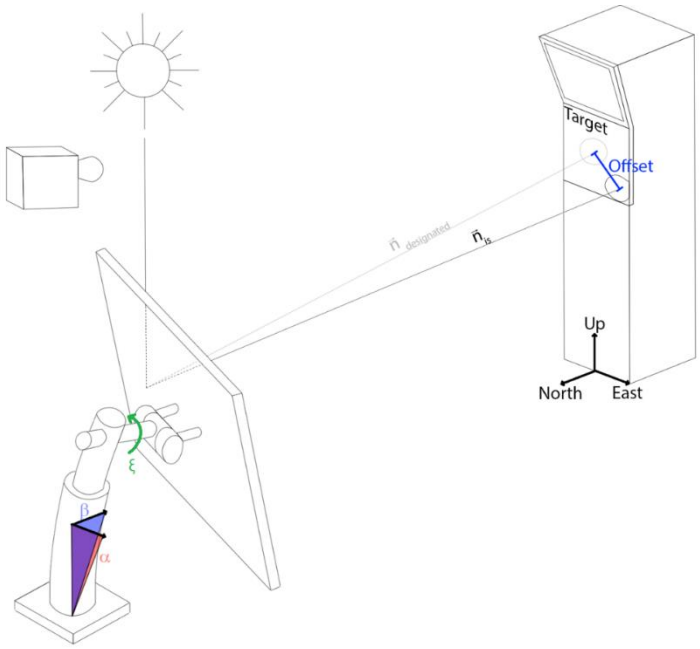

Figure 1. Sketch of a strongly deformed heliostat during the calibration process. The shown heliostat has some deformation on its pedestal and its first axis, which leads to an offset on the target, compared to a heliostat without errors. The shown error can be decomposed into two rotational displacements α, β, leading to the offset, and *x,* deforming the focal spot but not changing the offset. The calibration algorithm tries to quantify the deviations of α and β (as well as other error sources) by means of the deviation between n_{is} and $n_{designated}$.

The Camera-Target method (Stone method [11]) is one of the oldest and as of 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. 1). A camera then takes a picture of the focal spot. From this image, the focal spot's 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 pre-modeled mathematically or physically described errors (In fact, the error parameters don't have to be correlated with physical errors inside a geometrical model. In addition, 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.

But this is not always the case. The time per calibration measurement is the biggest weakness of the Stone method. With about one minute measurement time per heliostat, a single heliostat (in a field of thousands) is measured only a few times a year. The situation is aggravated by the fact that the measurement should be performed as infrequently as possible, since the measurement negatively affects the incoming power on the receiver. At the same time, the required tracking accuracy of the heliostat is very high and therefore induces a huge number of free variables inside the GM for a full description.

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, neglecting non-linearities.

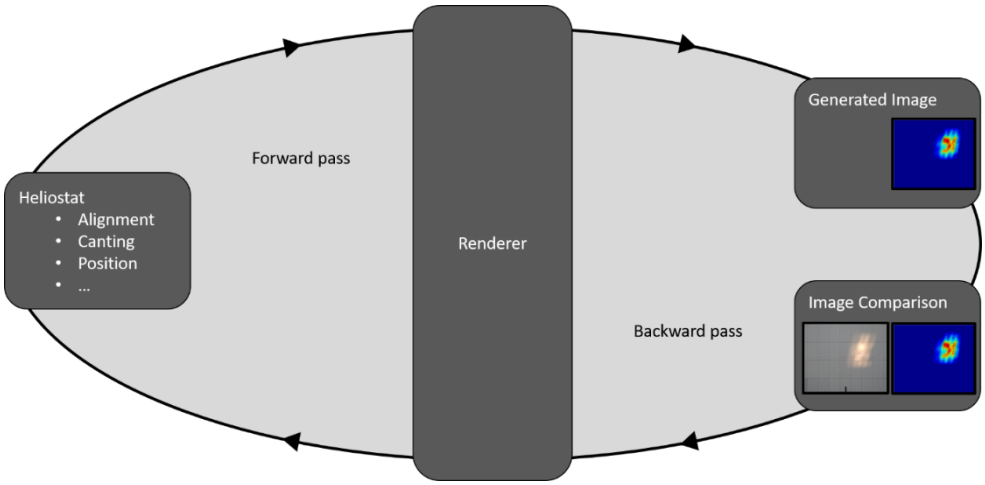

Figure 2. Optimization workflow for heliostat calibration using diff. ray tracing. The update routine strongly resembles that of neural networks, which is only made possible by the differentiable description of the power plant.

3. Differentiable Ray Tracing 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, e.g. 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 (compare Fig. 2). Diff. ray tracing starts like classical (forward) ray tracing by defining environment parameters (material, light, geometry, etc.) as well as camera ex- and intrinsics (camera angle, lenses, etc.). The object properties to be examined (alignment, position, etc.) only need to be defined approximately. Often, an ideal geometry is sufficient [3]. Using this coarse approximation, the ray tracer generates images that are not exactly identical with the observations but deviate according the inaccuracy of the parameter choice.

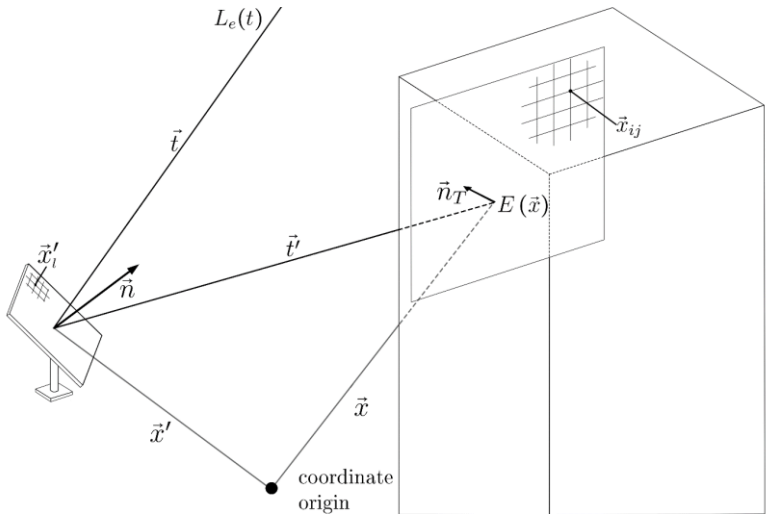

Figure 3. Sketch of the variables for calculating the irradiance profile of the heliostat.

Exactly like for the training of neural networks, the resulting image is then compared with the *ground truth,* which is, in case of the solar tower, the calibration image. The comparison is done by an objective/loss function as it is used for neural networks (e.g. L1 or L2 pixel-wise

losses). In fact, all other NN optimization functions (e.g. *weight decay*, *optimizers* 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 *automatic differentiation,* for example with the *backpropagation algorithm [12]*. Hence the differentiable implementation of the renderer does not explicitly require an explicit implementation of the derivatives, but a suited implementation in a framework supporting automatic differentiation allows calculate the derivates automatically.

When using diff. ray tracing at solar towers, it must be ensured that the mathematical rendering raytracing model under investigation is differentiable. This must also be true for the irradiance reaching the calibration target, which is defined as:

$$E(\overrightarrow{x_{ij}}) = \sum_{\text{rays } k} \sum_{\text{pos } l} \omega_{ijk} \frac{1}{|\overrightarrow{t_l}|^2} \rho(M_l \overrightarrow{t_l}) L_e(M_l \overrightarrow{t_l}) \overrightarrow{n_T} \cdot \overrightarrow{t_l} \tag{1}$$

Where t_l is the vector directing from the position x_{ij} on the cal. target to position x_l in the heliostat. n_T is the normal vector on the cal. target surface. ρ is the reflectivity of the mirror and L the radiance emitted by the sun. Both functions depend on M_l, a rotation matrix depending on the alignment of the heliostat, defined as:

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

ω_{ij} is a binning function, which distributes the incoming rays to the lattice points *ij* of the resulting image (compare Fig. 3). For providing differentiability, it is defined as follows:

$$1 = \omega_{total} = \sum_{\text{rays } k} \sum_{0<n<N} 1 - \left(\frac{\overrightarrow{x_k} - \overrightarrow{x_n}}{\sum(\overrightarrow{x_k} - \overrightarrow{x_n})} \right) \tag{3}$$

where N is the number of considered nearest neighbors and x_n is the nearest neighbor position of the defined lattice. To get the contribution at the point *ij*, the function is evaluated at exactly this point:

$$\omega_{ij} = \sum_{\text{rays } k} \sum_{0<n<N} \left(1 - \left(\frac{\overrightarrow{x_k} - \overrightarrow{x_n}}{\sum(\overrightarrow{x_k} - \overrightarrow{x_n})} \right) \right) \delta(x_n - x_{ij}) \tag{4}$$

This distribution of the ray intensity over neighboring nodes allows the entire ray tracing process at the solar tower to be considered fully differentiable for the very first time. The training is further improved by additional intelligently selected loss functions.

$$L_{total} = a \cdot (L2_{pixel} + WD_{pixel}) + b \cdot (L2_{alignment} + WD_{alignment}) + c \cdot (L2_{miss}) \tag{5}$$

So the total loss is composed of 3 different modules. In general, L2 is defined as:

$$L2(x, y) = \sum_{n=1}^{N} |x_n - y_n|^2 \tag{6}$$

where xn is the prediction and yn the target value. For $L2_{\text{pixel}}$, the pixels of the irradiance map are compared to each other, $L2_{\text{alignment}}$ compares the current alignment n of the heliostat with the ideal one and $L2_{\text{miss}}$ penalizes rays which missed the calibration target by measuring their distance. If all rays are hitting the cal. target, this term is 0. In addition, the appropriate weight decay terms are also added to $L2_{\text{pixel}}$ and $L2_{\text{alignment}}$ to further regularize the function. Lastly, the terms are scaled by constant values *a, b, c,* where

a = 1e-5, b = 1 and *c = 1e5*, which were determined in the course of various simulations.

4. Comparison to State of the Art

The advantages of the new method compared to the state-of-the-art application can be shown very well within a minimal example. For this comparison, an arbitrary setup of heliostat, sun and calibration target is considered in both cases, but not the whole ray tracer is examined but the special case that not a single ray is generated.

The classic Stone method [11], (just like the diff. ray tracer) makes use of a geometry model including various error parameters. The geometry model calculates the orientation n_{is} (directing from the heliostat's mirror's center of mass to the measured centroid of area of its focal spot) and n_{pred} (calculated by the geometric model) of the heliostat from the input parameters (target point, sun position, heliostat position). Both alignments are then compared to each other and the error parameters are updated accordingly. Classically, this is done by a regression algorithm like the Levenberg-Marquardt (LM) or Newton algorithm, for example in the form:

$$F = \min_{\{\Omega\}} \sum_{i=1}^{N} L_{Alignment} = \min_{\{\Omega\}} \sum_{i=1}^{N} \arccos\left(\overrightarrow{n_{\text{is},i}} \cdot \overrightarrow{n_{\text{model},i}}\right) \tag{7}$$

Where *{Ω}* is the set of all considered error parameters and *N* the number of measurement points. The basic procedures of the classical and diff. ray tracing methods differ only in the calculation of the gradients. While e.g. the LM method relies on a numerical calculation of the Jacobian and Hessian matrix, the ray tracer uses only the Jacobian matrix calculated by automatic differentiation for backpropagation. In principle, it is possible to modify the backpropagation accordingly [13][14][15]. This would be very time consuming, but in principle feasible. Considering identical geometric models and applying the same loss function $L_{\text{alignment}}$ in both algorithms, the results of the two algorithms would be identical, except for numerical truncation errors inside the classical approach (automatic differentiation is in principle accurate down to machine accuracy).

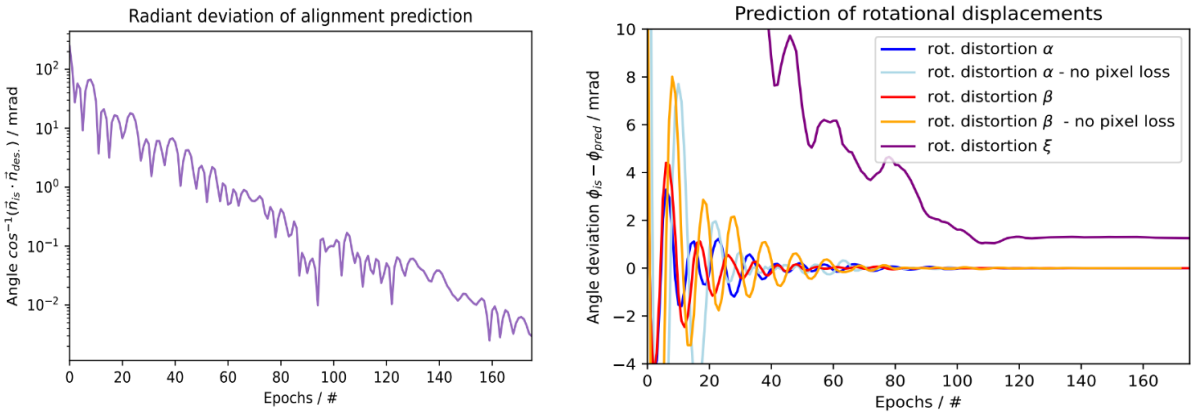

Figure 4. Test losses for heliostat calibration using differentiable ray tracing over several epochs showing in (a) the alignment deviation of the trained heliostat to the target data set heliostat and (b) the angular deviation of the three error parameters inside the GM.

Thus, both approaches can be described almost identically. However, so far only the minimal example has been considered. The focal spot contains much more information than just its center of gravity, e.g. horizontal and vertical tilt or the size of the focal spot can give further information about the orientation of the heliostat. Within the ray tracer, this information does not have to be extracted first; instead, it is possible to work directly with the images from

the heliostat calibration. Thus, significantly more information can be accessed than is the case with the classical method. Furthermore, errors can be optimized which were impossible to detect before, which will be shown in the following.

5. Results

The potential of the method is illustrated by a simple test case. For this purpose, a very simple geometry model for the heliostat is chosen, which contains only 3 error parameters α, β, ξ (compare Fig. 1). Only α and β have an influence on the position of the focal spot. ξ rotates the focal spot around its centroid.

To begin, a dataset containing one image is created. For this, a heliostat model with a surface different from an ideal heliostat is loaded into the ray tracer and random alignment errors in α, β and ξ are assigned to it. By means of this heliostat, exactly one focal spot image is generated on the virtual calibration target. This one image is used as a data set for training. 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. 4 (a), 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 identical 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. Simultaneously it also shows, that ξ is not perceived by the alignment loss. Otherwise, the curve would stop at a constant value.

More exciting is a look at the individual error parameters. Fig. 4 (b) shows very clearly how α and β oscillate continuously towards zero, also because of the very simple GM. More exciting is the comparison between the lines trained with L_{total} (red and blue) with those exclusively trained with $L2_{\text{alignment}}$ (orange, light blue), meaning without using the pixel-wise loss. If all loss terms are used the algorithm converges much faster.

ξ, on the other hand, behaves completely differently. This is due to the fact that the gradients are formed in another way. ξ remains completely constant (far from the range of the plot),

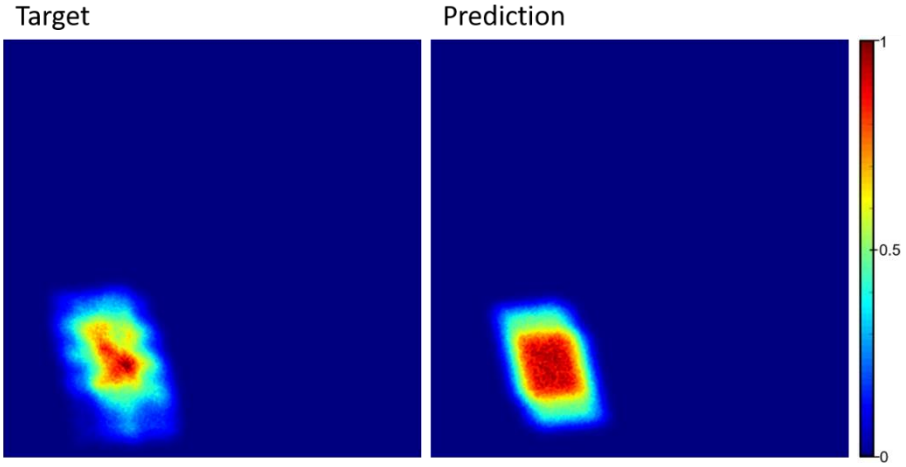

Figure 5. Target Image including surface deformations (left) and prediction by the diff. ray tracer (right) using an ideal heliostat surface.

only using the $L_{\text{alignment}}$ (or $L2_{\text{alignment}}$) loss. ξ is only optimized over the $L2_{\text{pixel}}$ loss. In Fig. 4 (b), ξ was set to 17.5 rad and ends up around 1.5 mrad, a deviation which is not visible to 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.

The result after full convergence can be seen in Fig. 5. Both images show the virtual calibration target including one normed focal spot each. On the left is the data set image (Target) including an imperfect surface for training and on the right side is the ray tracing calibration image (Prediction) using the optimized parameters and a perfect flat, faceted surface is displayed. Despite the considerably different surface, the position and inclination of the heliostat can be recognized well.

6. Discussion

So far, the diff. ray tracer for heliostat calibration could be theoretically presented and tested on minimal examples. Despite the reduced complexity of these, the advantages over classical regression are already apparent.

In comparison with the current state-of-the-art algorithm, it could be shown that the ray tracer, not generating a single ray, and the classical algorithm are almost identical. The only difference in the discussed minimal example lies in the calculation of the differentials. Here, the diff. ray tracer already has an advantage due to the use of automatic differentiation, which provides smaller truncation errors than the numerical differentiation, even if this effect is very small. The greater advantage arises when the minimal example is left behind. Generating rays allows for the very first time to evaluate the calibration images directly inside calibration software. This increase in information per measurement is available without any need of pre-processing. The additional information can then help with faster convergence or higher accuracy on the same count of measurements, which was shown in Fig. 4 (b). On the one hand, the pixel loss helps with a faster convergence of α and β, on the other hand, ξ can be corrected exclusively via this loss, since ξ has no influence on the position on the target. Beside the additional information, like edges, corners, etc., also e.g. errors in the determination of focal spots' centroids can be compensated by means of this loss (compare Fig. 5). Despite the very good qualitative results, a quantitative comparison especially with the state-of-the-art method and a corresponding complex geometry model is still pending. Here, however, the question is not so much whether the new method is better, but by how much. In addition, at this moment only ideal flat heliostats has been taken for training. Surface information can have a large impact on the results. In this case, specialized loss terms can be used for further improvement [16][17][18]. Here, too, a quantitative evaluation is still required.

7. Conclusion and Outlook

Despite the very simplistic GM model used so far, it is becoming unambiguously clear that heliostat calibration using diff. ray tracing is superior to classical regression and thus, with very little effort, the standard calibration method on most commercial towers can be improved in situ. The next steps clearly focus on the implementation of a fully functional geometry model and the realization directly at a solar tower. However, differentiable ray tracing goes far beyond these possibilities at the solar tower. The orientation of the heliostat is by no means the only parameter that can be optimized. This method can be applied to nearly every field parameter and thus opens up completely new possibilities. For example, surface information can be derived from focal spot images too. Furthermore, whole heliostat fields can be optimized for given flux density distributions, including heliostat shape, position, but also e.g. tower height or receiver shape.

Competing Interests

The authors declare no competing interests

Data and Code Availability

The full code of our differentiable ray tracing pipeline will be released on https://github.com/DLR-SF/holisticDIRC.

Author Contribution

Max Pargmann and **Jan Ebert** Conceptualization Methodology Software Validation Formal Analysis Writing - Original Draft Writing - Review & Editing Visualization **Stefan Kesselheim** Conceptualization Methodology Supervision **Daniel Maldonado Quinto** Supervision **Robert Pitz-Paal** Funding acquisition

References

1. C. Sattler et al. Review of heliostat calibration and tracking control methods, Solar Energy, Vol. 207, Elsevier, p. 110-132, 2020 DOI: https://doi.org/10.1016/j.solener.2020.06.030
2. B. Nicolet et al. “Large steps in inverse rendering of geometry” ACM Transactions on Graphics (TOG), Vol. 40, No. 6, p. 1-13, 2021 DOI: https://doi.org/10.1145/3478513.3480501
3. G. Loubet et al. “Reparameterizing discontinuous integrands for differentiable rendering” ACM Transactions on Graphics (TOG), Vol. 38, No. 6, p. 1-14, 2019 DOI: https://doi.org/10.1145/3355089.3356510
4. T. Li et al. “Differentiable monte carlo ray tracing through edge sampling”, ACM Transactions on Graphics (TOG), Vol. 37, No. 6, p. 1-11, 2018 DOI: https://doi.org/10.1145/3272127.3275109
5. S. Pattanaik et al. “Computation of global illumination by Monte Carlo simulation of the particle model of light”, Third Eurographics Workshop on Rendering, p.71-83 1992 DOI: https://doi.org/10.1002/vis.4340040303
6. H. Kato et al. “Differentiable rendering: A survey”, arXiv preprint arXiv:2006.12057, 2020 DOI: https://doi.org/10.48550/arXiv.2006.12057
7. Q. Sun et al. “End-to-end complex lens design with differentiable ray tracing”, ACM Transactions on Graphics, Vol. 40, No. 4, p. 1-13, 2021 DOI: https://doi.org/10.1145/3450626.3459674
8. L. Jain et al. “Generating Semantic Adversarial Examples with Differentiable Rendering”, CoRR, Vol. abs/1910.00727, 2019 DOI: https://doi.org/10.48550/arXiv.1910.00727
9. J. Tu et al. “Exploring Adversarial Robustness of Multi-Sensor Perception Systems in Self Driving”, arXiv 2021, doi: https://doi.org/10.48550/arxiv.2101.06784
10. Y. He, “Numerical simulation of solar radiation transmission process for the solar tower power plant: from the heliostat field to the pressurized volumetric receiver”, Applied thermal engineering, Vol. 61, No. 2, p. 583-595, 2013 DOI: https://doi.org/10.1016/j.applthermaleng.2013.08.015
11. K. Stone Automatic heliostat track alignment method, Google Patents, 1986
12. R. Rojas, “The backpropagation algorithm” Neural Networks, Springer, p. 149-182, 1996 DOI: https://doi.org/10.1007/978-3-642-61068-4_7
13. H. Yu “Levenberg-marquardt training”, Industrial electronics handbook, Vol. 5, No. 12, p.1, 2011 ISBN: 9781315218427
14. A. Suratgar “Modified Levenberg-Marquardt method for neural networks training”, World Acad Sci Eng Technol, Vol. 6, No. 1, p.46-48, 2005

15. A. Reynaldi, “Backpropagation and Levenberg-Marquardt algorithm for training finite element neural network”, Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, p. 89-94, 2012 DOI: https://doi.org/10.1109/EMS.2012.56
16. F. Schroff, “Facenet: A unified embedding for face recognition and clustering”, Proceedings of the IEEE conference on computer vision and pattern recognition, p 815-823, 2015 DOI: https://doi.org/10.48550/arXiv.1503.03832
17. W. Chen, “Beyond triplet loss: a deep quadruplet network for person re-identification”, Proceedings of the IEEE conference on computer vision and pattern recognition, p. 403-412, 2017 DOI: https://doi.org/10.48550/arXiv.1704.01719
18. J. Ribera, “Weighted hausdorff distance: A loss function for object localization”, arXiv preprint arXiv:1806.07564, Vol. 2, p.1, 2018 DOI: https://doi.org/10.48550/arXiv.1806.07564