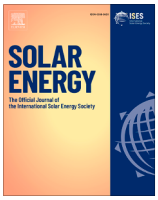
Solar Energy 310 (2026) 114495

Contents lists available at ScienceDirect

Solar Energy

journal homepage: www.elsevier.com/locate/solener



Airborne real-time solar concentrator orientation estimation for heliostat coarse calibration

Alexander Schnerring [a, c, *], Rafal Broda [a, c], Michael Nieslony [a], Niels Algner [a], Eduardo Saez Martinez [a], Marc Röger [a], Sonja Kallio [a], Robert Pitz-Paal [b, c]

[a] German Aerospace Center (DLR), Institute of Solar Research, Calle Doctor Carracido 44, 04005, Almería, Spain
[b] German Aerospace Center (DLR), Institute of Solar Research, Linder Höhe, 51147, Cologne, Germany
[c] RWTH Aachen University, Chair of Solar Technology, Linder Höhe, 51147, Cologne, Germany

HIGHLIGHTS

- Airborne optical method for joint estimation of camera pose and solar concentrator orientation.
- Purely vision-based approach removes dependence on additional hardware sensors.
- Single-image estimation enables real-time applicability.
- Method achieves coarse calibration accuracy.

ARTICLE INFO

Keywords:
Unmanned aerial vehicle
Real-time
Heliostat calibration
Condition monitoring
Concentrated solar power

ABSTRACT

We present a novel method for the airborne real-time solar concentrator orientation estimation for heliostat coarse calibration, termed ARTSCORE-C. The input to the method is the set of image coordinates of identified concentrator corner points within a single image. Additionally, it requires the known heliostat positions, the kinematic model and the concentrator geometry, quantities that are typically well known in heliostat fields. The method is well-suited for real-time applications and lays the foundation for autonomous condition monitoring systems for heliostat fields. It was developed with the aid of a dedicated simulation environment, which enabled a thorough analysis of noise propagation throughout the algorithm. Insights from simulation were successfully transferred to real-world data, demonstrating that simulation tools can effectively support the development of airborne image-based systems. Validation experiments at the solar tower test facility in Jülich, Germany confirmed the method's practical utility: Using 20 observations per concentrator, the method achieved an average angular RMSE of 3.5 mrad across 146 validated concentrators with a range of typical orientations. The achieved accuracy qualifies the method as a coarse calibration system, suitable to be used as an upstream-component for fine calibration systems. Moreover, its ability to optically estimate the camera pose for a given image makes it a promising alternative to hardware-based localization systems, particularly in remote environments, where correction signals may be unavailable. This capability enhances its practical relevance for large-scale, low-maintenance solar tower power facilities.

1. Introduction

The heliostat field of concentrating solar power (CSP) tower plants consists of up to several thousand heliostats, concentrating solar radiation onto a central receiver. To achieve this, each heliostat must continuously track the angle bisector between the sun incident vector and the vector pointing from the heliostat to the receiver. Various causes lead to heliostat tracking errors [1], reducing heliostat field performance. Calibrating the heliostat field to identify and compensate for these errors is therefore essential for maintaining energy yield and system efficiency [2].

Recent work has focused on using computer vision techniques applied to aerial images captured by unmanned aerial vehicle (UAV)-mounted cameras [3–6]. However, existing UAV-based

* Corresponding author at: German Aerospace Center (DLR), Institute of Solar Research, Calle Doctor Carracido 44, 04005, Almería, Spain.
Email address: alexander.schnerring@dlr.de (A. Schnerring).

https://doi.org/10.1016/j.solener.2026.114495
Received 7 October 2025; Received in revised form 16 February 2026; Accepted 3 March 2026
Available online 12 March 2026

Acronyms

CCS	concentrator coordinate system
CSP	concentrating solar power
DBF	descriptor-based filtering
DLR	German Aerospace Center
EOR	exterior orientation
GCS	global coordinate system
HFOV	horizontal field of view
ICS	image coordinate system
IF	image filtering
IOR	interior orientation
NIO	non-intrusive optical
OCS	observer coordinate system
PNP	perspective-N-point
RCS	reference coordinate system
RMSE	root mean squared error
ROF	representative observation filtering
RTK	real-time kinematics
STJ	Solar Tower Jülich
UAV	unmanned aerial vehicle
VFOV	vertical field of view
VSLAM	visual simultaneous localization and mapping

heliostat inspection approaches typically rely on pre-planned flight routes and perform the data analysis offline, after the UAV has already landed. The lack of immediate feedback prevents the application of existing approaches in autonomous systems, where the dynamic adaptation of the UAV flight path requires real-time analysis.

To overcome these limitations, we propose a novel system for the airborne real-time solar concentrator orientation estimation for heliostat coarse calibration, referred to as ARTSCORE-C in the following. The main contributions of this work are: (i) A single-image-based algorithm for the simultaneous estimation of camera pose and concentrator orientations, leveraging prior knowledge of the heliostat positions, the concentrator geometry and the heliostat kinematic system. (ii) A simulation-based analysis of this newly presented algorithm, providing insights into how uncertainty propagates to subsequent system components and supports a deeper understanding of the system's limitations and potential optimization. (iii) An uncertainty-aware strategy to enhance the concentrator orientation estimation accuracy by averaging across multiple views. (iv) An experimental validation, based on a dataset collected at the Solar Tower Jülich (STJ) test facility under realistic flight conditions.

Both single-image-based estimation and averaging over multiple views are designed for real-time use, with processing times comparable to the typical intervals between consecutive images in drone measurement scenarios. While the validation is performed offline in this work, no fundamental limitations prevent the method from being executed in real-time, provided appropriate onboard or ground-based computational resources are available. This real-time capability opens the door to the key innovation of dynamically adapting UAV flight paths based on immediate orientation estimation results. Such closed-loop operation could be a step toward the autonomous inspection and calibration of heliostat fields. To the best of our knowledge, no existing system offers online, image-based concentrator orientation estimation based on a UAV-mounted camera.

This paper is structured as follows. Section 2 reviews relevant literature on heliostat calibration and vision-based approaches for joint camera pose and environment state estimation. Section 3 introduces the necessary preliminaries for the proposed method. Section 4 formulates the estimation framework, decomposing the overall problem into two coupled subproblems. Section 5 presents a preliminary simulation-based analysis that isolates the influence of perspective on estimation quality. This targeted study focuses on component-level behavior and provides the conceptual basis for the subsequent validation using real-world data. Section 6 describes the recording of the validation dataset acquired at the STJ test facility, analyzes the noise characteristics of the perception module and describes how the data is processed prior to validation. Section 7 presents the validation results and provides a detailed discussion of the findings. Finally, Section 8 summarizes the key contributions and outlines directions for future work.

2. Related work

The calibration of heliostat fields is essential for maintaining the optical efficiency of central receiver CSP plants. Several methods have been developed to estimate heliostat orientation and optical quality. This section reviews representative techniques and outlines how the proposed method complements or addresses limitations in existing approaches.

2.1. Camera-target method

To this day, the camera-target method [7] is still the state-of-the-art fine calibration method [2]. It involves directing the reflected solar beam of a heliostat onto a planar target, typically mounted below the central receiver. An image of the resulting flux distribution is captured, and the center of the distribution is extracted through image processing. The heliostat normal vector is then computed as the angle bisector between the known sun vector and the vector from the heliostat to the observed flux centroid. This normal vector serves as a ground-truth calibration point for fitting the degrees of freedom of a kinematic model.

The camera-target method yields high precision while relying on a simple, cost-effective, and robust measurement setup. However, it suffers from several limitations. Since it relies on direct solar illumination and the sequential calibration of individual heliostats, acquiring a sufficient number of calibration points across the field may take several months to a year. Moreover, heliostats with large initial tracking errors may entirely miss the target, rendering the method inapplicable without a preceding coarse calibration (typically requiring an accuracy better than 10 mrad [2]). Additionally, the method is weather-dependent and requires clear-sky conditions.

The method proposed in this work addresses these limitations by enabling fast, image-based estimation of heliostat orientations. It is well-suited for pre-calibrating new heliostats as well as detecting and coarsely recalibrating outlier heliostats that would otherwise be missed by the camera-target method. Hence, the proposed approach complements the camera-target method and can serve as an upstream component to improve and accelerate the fine calibration process.

2.2. Non-intrusive optical (NIO) method

The NIO method is a UAV-based method that jointly estimates mirror slope errors, canting errors, and tracking errors by analyzing reflections of the tower in the concentrator surface [3]. NIO is a promising approach yielding fine calibration accuracies. However, it depends on the presence of the tower, relies on the concentrators to be oriented as expected and requires precise planning of flight trajectories to ensure that reflections appear as anticipated in the image plane. These requirements may complicate its application during early calibration phases, where the tower has not yet been fully integrated and the concentrator orientations are poorly known. In contrast, the method presented here does not rely on

reflections of a tower and requires no a-priori knowledge of the concentrator orientations. It estimates concentrator orientation directly from corner point projections.

2.3. HelioPoint

HelioPoint is another UAV-based fine calibration method, using a point light source instead of a tower. This characteristic enables its use also for initial calibration, even if the tower has not yet been constructed [5]. While it removes the dependency on the tower infrastructure, it still requires a pre-calibration of the measured heliostat to ensure that the light source reflection is captured during the UAV overflight. The proposed method can hence serve as an upstream component in a *HelioPoint*-based calibration procedure.

2.4. Bundle adjustment

Jessen et al. [6] propose a two-stage calibration framework, wherein the first stage, the concentrator orientations are derived from a corner point cloud reconstructed via bundle adjustment. While principally enabling coarse calibration accuracy, this approach has several drawbacks: The bundle adjustment process is computationally intensive, flight trajectories must be optimized for 3D reconstruction, and the method requires a static scene. These constraints prohibit in-flight evaluation and limit its applicability in operating heliostat fields where heliostats are tracking continuously. In contrast, ARTSCORE-C estimates both concentrator orientations and the camera pose from single images, without relying on a static field. This allows for flexible application in dynamic conditions, including tracking fields. If certain hardware requirements are met, ARTSCORE-C's output is available in real-time.

2.5. Camera pose estimation

Accurate camera pose estimation is a critical prerequisite for UAV-based methods in heliostat field inspection, enabling tasks such as heliostat identification and concentrator orientation estimation. While hardware-based solutions like real-time kinematics (RTK)–global positioning system (GPS) can provide accurate positioning, they may suffer from reduced reliability in remote environments due to signal multipath effects and obstructions caused by steel heliostat structures, as noted by Lombard and Smit [8]. On the other hand, standard (non-differential) GPS lacks the necessary accuracy. Even when RTK is available and functioning properly, gimbal accuracy often remains the limiting factor in tasks that rely on precise camera orientation estimates.

Optical methods, such as visual simultaneous localization and mapping (VSLAM), offer an alternative by jointly estimating the camera trajectory and a map of the environment from image sequences. While powerful in general settings, conventional VSLAM systems are designed for unstructured and unknown environments, where no prior scene information is available [9]. In contrast, heliostat fields are highly structured: heliostat positions, concentrator geometry, and the kinematic system are typically known with good accuracy. As a result, using VSLAM in this context leads to redundant estimation of already known quantities and ignores useful prior constraints. Moreover, VSLAM relies on multi-view consistency and bundle adjustment. This assumes static scenes, and introduces challenges such as scale drift and the need for loop closures [9]. These issues are especially challenging in large-scale, outdoor heliostat fields where UAVs may cover long trajectories without returning to previously seen areas. While recent advances in VSLAM address these issues, they remain inherently multi-view and optimization-heavy. The method presented in this work fills this gap by leveraging a-priori knowledge to enable accurate single-image estimation without requiring a static heliostat field and a global joint optimization of camera poses.

2.6. AI-based detection of concentrator corner points

While previous approaches for the detection of concentrator corner points relied on classical computer vision techniques, Broda et al. [10] introduced an AI-based system trained on synthetic images of heliostat fields. This approach circumvents key limitations of classical systems, such as the need for prior knowledge of concentrator orientations and an initial camera pose estimate [6]. Moreover, the system of Broda et al. provides the required input for the present method within processing times compatible with real-time applications. Hence, the work presented in Broda et al. [10] is an enabling factor for the work presented here.

2.7. Contribution of the proposed method

The method proposed in this work enables the joint estimation of concentrator orientations and camera pose from a single image. It builds on the perception module presented in Broda et al. [10] and leverages a-priori knowledge of heliostat field quantities that are typically well known. To the best of our knowledge, no previous approach has achieved such single-image-based joint estimation under conditions representative of solar tower plants. As it does not rely on multi-frame optimization or static scenes, it is compatible with tracking fields and can operate under dynamic conditions. When implemented on an embedded system or paired with a suitable communication link, the approach supports near-instantaneous camera pose and concentrator orientation estimation, enabling applications such as real-time adaptation of the UAV flight path.

By providing concentrator orientation estimates, the proposed method can serve as a calibration tool in its own right or act as a front-end module for fine calibration methods like HelioPoint, NIO, or camera-target-based calibration. Its compatibility with autonomous and scalable inspection workflows contributes to the broader goal of reducing operational costs in CSP systems. This aligns with strategic goals outlined in the HelioCon Roadmap [11], which emphasizes the need to further develop inspection tools to make solar tower technology cost-competitive compared to other energy sources.

3. Preliminaries

3.1. Conventions and notation

The heliostat field is assumed to consist of multiple heliostats, where each heliostat consists of two tracking axes and a concentrator. The concentrator coordinate system (CCS) of a heliostat j is denoted by CCS_j. The heliostat position data is defined w.r.t. the global coordinate system (GCS), chosen to be a local east-north-up coordinate system on the northern hemisphere in accordance with the *SolarPACES Guideline for Heliostat Performance Testing* [12,13]. Likewise, the pose of a camera moving through the heliostat field is described by its own coordinate system, which changes over time as the camera navigates the environment. As the acronym CCS is already in use, the camera coordinate system is denoted by observer coordinate system (OCS). The 2D coordinate system associated with the camera sensor is denoted by image coordinate system (ICS). Fig. 1 illustrates the coordinate systems used throughout this work.

Scalar values are represented as lowercase letters with regular font, vectors as lowercase letters with bold font, and matrices as uppercase letters with bold font.

3.2. Coordinate transformations

The transformation between two coordinate systems CS1 and CS2 is described by a rigid body transformation and conveniently summarized in a 4×4 matrix of the form

$$\boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_{\mathrm{CS2}}^{\mathrm{CS1}} & \boldsymbol{t}_{\mathrm{CS2}}^{\mathrm{CS1}} \\ \boldsymbol{0} & 1 \end{pmatrix}, \tag{1}$$

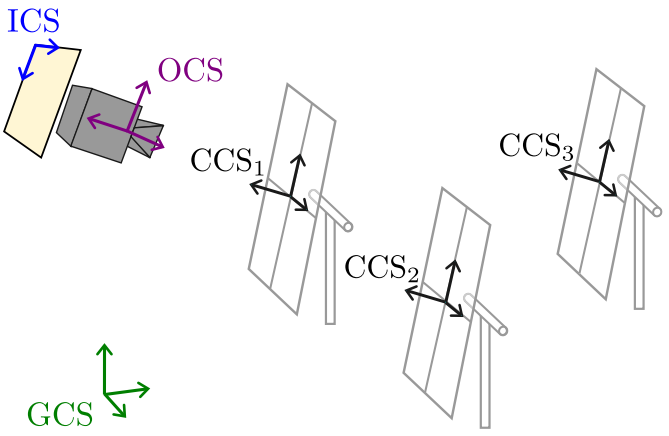


Fig. 1. Overview of the coordinate systems used in this work.

where $\boldsymbol{t}_{\mathrm{CS2}}^{\mathrm{CS1}}$ denotes the translation from CS1 to CS2 and $\boldsymbol{R}_{\mathrm{CS2}}^{\mathrm{CS1}}$ denotes the rotation from CS1 to CS2. We define the operator $\mathrm{Rot}(\cdot)$ to extract the rotational part from a transformation, while the operator $\mathrm{Trans}(\cdot)$ extracts its translational part, i.e.,

$$\begin{aligned} \mathrm{Rot}\left(\boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}}\right) &= \boldsymbol{R}_{\mathrm{CS2}}^{\mathrm{CS1}} , \\ \mathrm{Trans}\left(\boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}}\right) &= \boldsymbol{t}_{\mathrm{CS2}}^{\mathrm{CS1}} . \end{aligned} \tag{2}$$

A point $\boldsymbol{p}^{\mathrm{CS2}} = (p_{\mathrm{X}}^{\mathrm{CS2}}, p_{\mathrm{Y}}^{\mathrm{CS2}}, p_{\mathrm{Z}}^{\mathrm{CS2}})^T$ expressed w.r.t. CS2 can be expressed w.r.t. CS1 in the following way:

$$\begin{pmatrix} p^{\mathrm{CS1}} \\ 1 \end{pmatrix} = \boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}} \cdot \begin{pmatrix} p^{\mathrm{CS2}} \\ 1 \end{pmatrix} . \tag{3}$$

To simplify notation, we denote the transformation of $\boldsymbol{p}^{\mathrm{CS2}}$ to CS1 by

$$\boldsymbol{p}^{\mathrm{CS1}} = \boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}} \cdot \boldsymbol{p}^{\mathrm{CS2}} \tag{4}$$

but implicitly apply Eq. (3). Conversely, a point expressed w.r.t. CS1 can be expressed w.r.t. CS2 by multiplying with the inverse of $\boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}}$:

$$\boldsymbol{p}^{\mathrm{CS2}} = \left(\boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}}\right)^{-1} \cdot \boldsymbol{p}^{\mathrm{CS1}} = \boldsymbol{T}_{\mathrm{CS1}}^{\mathrm{CS2}} \cdot \boldsymbol{p}^{\mathrm{CS1}} . \tag{5}$$

A point cloud $\boldsymbol{P}$ is a collection of N points and can be expressed in the form of a matrix:

$$\boldsymbol{P}^{\mathrm{CS2}} = \begin{pmatrix} p_{\mathrm{X1}}^{\mathrm{CS2}} & p_{\mathrm{Y1}}^{\mathrm{CS2}} & p_{\mathrm{Z1}}^{\mathrm{CS2}} \\ \vdots & \vdots & \vdots \\ p_{\mathrm{XN}}^{\mathrm{CS2}} & p_{\mathrm{YN}}^{\mathrm{CS2}} & p_{\mathrm{ZN}}^{\mathrm{CS2}} \end{pmatrix}^T . \tag{6}$$

Analogously to Eqs. (3) and (4), a point cloud can be expressed w.r.t. another coordinate system, i.e., $\boldsymbol{P}^{\mathrm{CS1}} = \boldsymbol{T}_{\mathrm{CS2}}^{\mathrm{CS1}} \cdot \boldsymbol{P}^{\mathrm{CS2}}$.

3.3. Kinematic model

The kinematic model $\boldsymbol{K}$ describes the motion of CCS_j w.r.t. the GCS as a function of the kinematic parametrization $\boldsymbol{k}_j$ and the heliostat origin $\boldsymbol{o}_j$. In a forward pass, $\boldsymbol{K}^{\mathrm{f}}$ takes in two tracking angles θ_j and τ_j and computes the resulting concentrator pose:

$$\boldsymbol{K}_j = \boldsymbol{K}^{\mathrm{f}}(\theta_j, \tau_j \mid \boldsymbol{o}_j, \boldsymbol{k}_j) = \boldsymbol{T}_{\mathrm{CCS}_j}^{\mathrm{GCS}} . \tag{7}$$

In a backward pass, $\boldsymbol{K}^{\mathrm{b}}$ takes in a desired concentrator normal vector $\boldsymbol{n}_j$ and computes the concentrator pose that results in this normal vector:

$$\boldsymbol{K}_j = \boldsymbol{K}^{\mathrm{b}}(\boldsymbol{n}_j \mid \boldsymbol{o}_j, \boldsymbol{k}_j) = \boldsymbol{T}_{\mathrm{CCS}_j}^{\mathrm{GCS}} . \tag{8}$$

ARTSCORE-C only uses the backward pass $\boldsymbol{K}^{\mathrm{b}}$. When replicating the state of the heliostat field in simulation experiments, each simulated concentrator is aligned using $\boldsymbol{K}^{\mathrm{f}}$ and an estimate of $\boldsymbol{k}_j$, obtained through the camera-target method [7].

3.4. Perspective-N-point (PNP)

PNP algorithms aim to estimate the pose of a calibrated camera $\boldsymbol{T}_{\mathrm{OCS}}^{\mathrm{RCS}}$ w.r.t. a reference coordinate system (RCS) from a set of N points with known 3D positions expressed w.r.t. this RCS and their 2D projections on the camera sensor [14]. In this work, we express the estimated pose using the following notation:

$$\begin{aligned} \boldsymbol{T}_{\mathrm{OCS}}^{\mathrm{RCS}} &= \mathrm{PNP}(\boldsymbol{P}^{\mathrm{ICS}}, \boldsymbol{P}^{\mathrm{RCS}} \mid \mathrm{IOR}) \\ &= \mathrm{PNP}(\boldsymbol{P}^{\mathrm{ICS}}, \boldsymbol{P}^{\mathrm{RCS}}) . \end{aligned} \tag{9}$$

$\boldsymbol{P}^{\mathrm{ICS}}$ is a point cloud of dimension $(2 \times N)$ and contains the point projections detected in the image. $\boldsymbol{P}^{\mathrm{RCS}}$ is a point cloud of dimension $(3 \times N)$ and contains the 3D point positions, expressed w.r.t. the RCS. Note that the number of correspondences N is explicitly stated while the camera interior orientation (IOR) is omitted, assuming a calibrated camera throughout this work. The inverse of the PNP-estimated camera pose is denoted by

$$\boldsymbol{T}_{\mathrm{RCS}}^{\mathrm{OCS}} = \mathrm{PNP}^{-1}(\boldsymbol{P}^{\mathrm{ICS}}, \boldsymbol{P}^{\mathrm{RCS}}) . \tag{10}$$

4. Methodology

4.1. Problem formulation

In this work, a series of images is assumed to be recorded with a camera mounted on a UAV overflying a heliostat field. Each image is indexed by i. Each concentrator is indexed by j. The total number of images is denoted by I and the total number of concentrators by J. In each image i, only a subset of concentrators is visible. Conversely, each concentrator j is only visible in a subset of images.

Each image i is associated with the coordinate system OCS_i, where $\boldsymbol{T}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$ describes the pose of the UAV-mounted camera at the capture time of image i w.r.t. the GCS. Camera i observes the concentrators $j \in C(i)$ and, based on this, the concentrator poses $\boldsymbol{T}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ are first estimated w.r.t. OCS_i. This estimate is expressed w.r.t. the GCS using the estimated camera pose $\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$, resulting in the observation $\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)$.

4.1.1. Bipartite graph

This visibility relation can be described using a bipartite graph [15], whose nodes are divided into two sets, the set of all images and the set of all concentrators. The edges of this graph connect an image node i to a concentrator node j if concentrator j is visible in image i. This bipartite graph can be represented by an $(I \times J)$ matrix $\boldsymbol{V}$, where the rows correspond to the image set and the columns correspond to the concentrator set. In this matrix, $\boldsymbol{V}_{i,j} = 1$ if concentrator j is visible in image i and $\boldsymbol{V}_{i,j} = 0$ otherwise. Using $\boldsymbol{V}$, the set of all concentrators visible in image i is denoted by

$$C(i) = \{j : \boldsymbol{V}_{i,j} = 1\} \tag{11}$$

and the set of all images in which concentrator j is visible by

$$I(j) = \{i : \boldsymbol{V}_{i,j} = 1\} . \tag{12}$$

Fig. 2 shows an exemplary visibility matrix and its corresponding bipartite graph, along with an example of a concentrator subset and an image subset.

4.1.2. Static field assumption

In a tracking heliostat field, concentrator poses vary over time as they continuously follow the sun. Therefore, different images would, in general, refer to different ground-truth poses depending on the image recording time. We assume a static heliostat field in this work, meaning that the true pose $\boldsymbol{T}_{\mathrm{CCS}j}^{\mathrm{GCS}}$ remains constant across all images. This simplifying assumption allows for averaging multiple estimates of the same concentrator pose across images to obtain a refined estimate. However, an application of ARTSCORE-C in a tracking field is possible by synchronizing the image recording time with the heliostat field time.

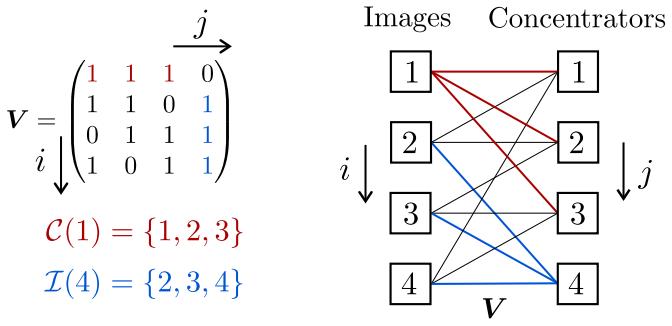


Fig. 2. Exemplary visibility graph and visibility matrix for $I = J = 4$.

4.1.3. Estimated quantities

The method internally estimates poses (i.e., position and orientation) of both the camera and the concentrators. However, the actual estimation goals differ:

- The camera pose $\hat{\boldsymbol{T}}^{\mathrm{GCS}}_{\mathrm{OCS}_i}$ (i.e., both position and orientation) is part of the final output.
- For the concentrators, only their estimated orientations $\hat{\boldsymbol{R}}^{\mathrm{CCS}_j}_{\mathrm{GCS}}$ are relevant. However, their full pose is still estimated as an intermediate quantity, required to compute $\hat{\boldsymbol{T}}^{\mathrm{GCS}}_{\mathrm{OCS}_i}$ and assess the uncertainty of the corresponding orientation estimate.
- Since rotations around the optical axis do not affect the direction of the reflected sunlight, calibration systems usually output a concentrator's surface normal vector rather than its full orientation. The normal vector is obtained from the estimated orientation by applying it to the unit Z-axis of the concentrator coordinate system.

As a result, ARTSCORE-C estimates both camera and concentrator poses for each image i. However, only the concentrator normal vectors are compared against reference data during validation, since estimating a normal vector is the primary focus of the method. A tabular overview of the coordinate systems and the estimated quantities can be found in Appendix A.

4.1.4. Problem 1 and Problem 2

With the aforementioned assumptions, the proposed method can be viewed as the solution to two subsequent problems:

1. Problem 1 is the task of jointly estimating the camera pose $\boldsymbol{T}^{\mathrm{GCS}}_{\mathrm{OCS}_i}$ and the concentrator poses $\boldsymbol{T}^{\mathrm{GCS}}_{\mathrm{CCS}_j}(i)$ for all $j \in C(i)$ from a single image i.
2. Problem 2 is the task of estimating the concentrator normal vector $\boldsymbol{n}^{\mathrm{GCS}}_j$ from all concentrator pose estimates $\hat{\boldsymbol{T}}^{\mathrm{GCS}}_{\mathrm{CCS}_j}(i)$ and camera pose estimates $\hat{\boldsymbol{T}}^{\mathrm{GCS}}_{\mathrm{OCS}_i}$ where $i \in I(j)$.

Fig. 3 illustrates both problems for the visibility graph shown in Fig. 2.

4.2. Problem 1

Problem 1 is solved by first estimating the concentrator orientations w.r.t. OCS_i, based on the concentrator geometry for all $j \in C(i)$. This first step is described in more detail in Section 4.2.1. In the second step, the camera pose for image i is estimated and the estimated concentrator orientations are expressed w.r.t. GCS. Due to a circular dependency, this step is solved using an iterative approach, as presented in more detail in Section 4.2.2.

A summary of all quantities used to solve Problem 1, along with the coordinate systems in which they are known, detected and estimated, can be found in Appendix A.

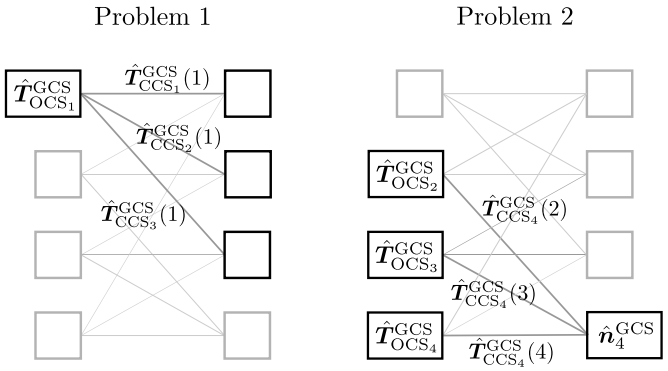


Fig. 3. Problem 1 (left side) is the task of jointly estimating the camera pose and the concentrator poses from a single image. Problem 2 is the task of estimating the concentrator orientations from all observations across different images.

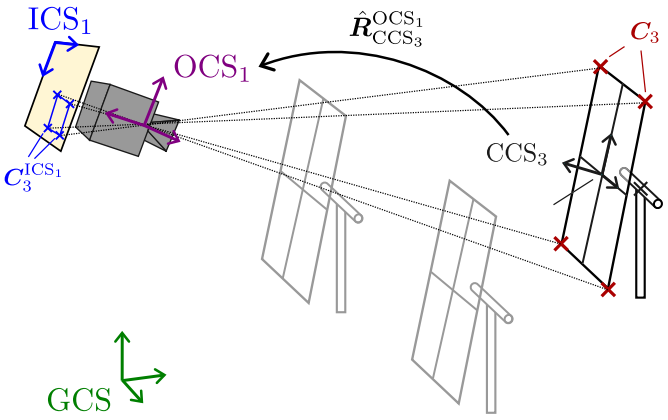


Fig. 4. Overview of Step 1 of Problem 1, in which the concentrator orientations $\hat{\boldsymbol{R}}^{\mathrm{OCS}_i}_{\mathrm{CCS}_j}$ are estimated for each $j \in C(i)$ using a P4P algorithm. This step uses the concentrator corner image coordinates $\boldsymbol{C}^{\mathrm{ICS}_i}_j$ and leverages a-priori knowledge of their coordinates w.r.t. CCS_j.

4.2.1. Step 1: Estimating $\hat{\boldsymbol{R}}^{OCS_i}_{CCS_j}$

The concentrator orientations $\hat{\boldsymbol{R}}^{\mathrm{OCS}_i}_{\mathrm{CCS}_j}$ are first estimated based on the concentrator geometry using a PNP solver. In this work, the four outer concentrator corner points are used, i.e., P4P is applied. A higher number of points can be used, as long as their positions w.r.t. the CCS are known and they can be detected in the captured image. The collection of corner points for concentrator j, expressed w.r.t. CCS_j is denoted by $\boldsymbol{C}^{\mathrm{CCS}_j}_j$. Their detected image coordinates in ICS_i are denoted by $\boldsymbol{C}^{\mathrm{ICS}_i}_j$. The orientation of each concentrator j w.r.t. OCS_i is estimated by

$$\hat{\boldsymbol{R}}^{\mathrm{OCS}_i}_{\mathrm{CCS}_j} = \mathrm{Rot}\left(\mathrm{P4P}^{-1}(\boldsymbol{C}^{\mathrm{ICS}_i}_j, \boldsymbol{C}^{\mathrm{CCS}_j}_j)\right) . \tag{13}$$

This estimation is illustrated in Fig. 4, which shows the process for one concentrator. The same estimation process is repeated for all concentrators j for which the outer four corners are detected in image i.

4.2.2. Step 2: Estimating $\hat{\boldsymbol{T}}^{GCS}_{OCS_i}$ and $\hat{\boldsymbol{T}}^{GCS}_{CCS_j}(i)$

To estimate the camera pose based on the single image i w.r.t. the GCS, a set of 3D passpoints $\boldsymbol{P}^{\mathrm{GCS}}_i$ has to be known along with the corresponding set of image coordinates $\boldsymbol{P}^{\mathrm{ICS}_i}_i$, detectable in ICS_i. One option would be to use dedicated passpoints distributed in the field. However, this approach has several drawbacks: their 3D positions w.r.t. the GCS would have to be measured separately, introducing additional effort.

Their visibility in aerial images is not guaranteed, as they may appear too small in the image, occluded by the concentrator surfaces or covered by dirt, vegetation, or shadows. For these reasons, it is preferable to use features that are structurally part of the heliostats themselves. Their locations are typically known, and the concentrator surfaces are consistently visible in aerial images. As a result, surface-based features provide a more robust and practical basis for passpoint selection in real-world conditions. A natural choice is the heliostat positions $\{\boldsymbol{o}_j \ \forall j \in C(i)\}$, usually measured with cm–level accuracy during the construction phase of the heliostat field. However, these positions typically describe internal mechanical features, such as the intersection of the tracking axes. Hence, their projections are usually not detectable in the image due to occlusion by the concentrator surfaces. Conversely, the 3D positions of image-detectable points on the concentrator surfaces usually vary as a function of the concentrator orientation and thus are not fixed w.r.t. the GCS.

To overcome this issue, we propose to use features located on the concentrator surfaces and to estimate their 3D positions using the concentrator orientations determined in Step 1, together with a-priori knowledge of the heliostat positions and the kinematic system: Given a normal vector $\boldsymbol{n}_j^{\mathrm{GCS}}$ and a known heliostat origin $\boldsymbol{o}_j^{\mathrm{GCS}}$, the kinematic backward pass $\boldsymbol{K}^{\mathrm{b}}$ allows us to estimate the 3D position of any point with known coordinates in CCS_j w.r.t. the GCS. This way, image-detectable points on the concentrator surface can serve as passpoints for the camera pose estimation.

However, the estimated concentrator orientations obtained in Step 1 are only expressed w.r.t. the OCS_i. These orientations must first be expressed w.r.t. the GCS, requiring knowledge of the camera orientation w.r.t. the GCS. To resolve this interdependence, an initial rough estimate of the camera pose is determined using a PNP algorithm, where the image coordinates are set to the detected passpoints $\boldsymbol{P}_i^{\mathrm{ICS}_i} = \{\boldsymbol{p}_j^{\mathrm{ICS}_i} \forall j \in C(i)\}$, located on the concentrator surfaces. In the case that the passpoint is located in the center of the concentrator surface, its image coordinates can be derived from the four corresponding concentrator corner image coordinates $\boldsymbol{C}_j^{\mathrm{ICS}_i}$ using the algorithm described in [16] (see Appendix D).

The 3D passpoint positions are set to the respective heliostat origins $\{\boldsymbol{O}_i^{\mathrm{GCS}} = \boldsymbol{o}_j^{\mathrm{GCS}} \ \forall j \in C(i)\}$:

$$\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}} = \mathrm{P}N_i\mathrm{P}(\boldsymbol{P}_i^{\mathrm{ICS}_i}, \boldsymbol{O}_i^{\mathrm{GCS}}) \,. \tag{14}$$

Note that in the above equation, N_i denotes the number of visible concentrators in image i, i.e., $N_i = |C(i)|$. It is advantageous for the passpoints to lie close to the respective heliostat origins, such that the detected passpoints $\boldsymbol{P}_i^{\mathrm{ICS}_i}$ are sufficiently consistent with the heliostat origins $\boldsymbol{O}_i^{\mathrm{GCS}}$.

This initial camera pose estimate is used to transform the relative orientations $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ into the global orientations $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)$, yielding an estimate of the surface normal vector $\boldsymbol{n}_j^{\mathrm{GCS}}$ for each concentrator:

$$\begin{aligned} \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) &= \mathrm{Rot}\left(\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}\right) \cdot \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i} \,, \\ \hat{\boldsymbol{n}}_j^{\mathrm{GCS}}(i) &= \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) \cdot (0\ 0\ 1)^T \,. \end{aligned} \tag{15}$$

These estimated normal vectors are then fed to the kinematic backward pass (see Eq. 8). Since the aim of ARTSCORE-C is to collect calibration points in order to estimate the kinematic parametrization $\boldsymbol{k}_j$, no prior knowledge is assumed and the design parameters $\boldsymbol{k}$ are used, i.e., $\boldsymbol{k}_j$ is set to $\boldsymbol{k}$ for all concentrators j. Hence, the resulting transformation

$$\hat{\boldsymbol{K}}_j(i) = \boldsymbol{K}^{\mathrm{b}}(\hat{\boldsymbol{n}}_j^{\mathrm{GCS}}(i) \mid \boldsymbol{o}_j^{\mathrm{GCS}}, \boldsymbol{k}) \tag{16}$$

is only an estimation of the true concentrator pose under the assumption that $\boldsymbol{o}_j^{\mathrm{GCS}}$ and $\boldsymbol{k}$ are valid. Furthermore, $\hat{\boldsymbol{K}}_j(i)$ contains the uncertainties of the surface normal vector estimate. Note that $\hat{\boldsymbol{K}}_j(i)$ is solely used to compute the 3D coordinates of the image-detectable passpoint w.r.t. the GCS:

$$\hat{\boldsymbol{p}}_j^{\mathrm{GCS}}(i) = \hat{\boldsymbol{K}}_j(i) \cdot \boldsymbol{p}_j^{\mathrm{CCS}_j} \,. \tag{17}$$

This approach leverages structural knowledge of the heliostat design while avoiding reliance on individually calibrated kinematic parameters. This is particularly relevant in construction-phase scenarios, where individual kinematic parameters $\boldsymbol{k}_j$ may not yet be available. In contrast, if the method is applied to an already calibrated field, the individually identified parameters $\boldsymbol{k}_j$ can be used in place of the design parameters $\boldsymbol{k}$ in Eq. (16), potentially increasing the accuracy of the estimated 3D positions.

The set $\hat{\boldsymbol{P}}_i^{\mathrm{GCS}} = \{\hat{\boldsymbol{p}}_j^{\mathrm{GCS}}(i) \ \forall j \in C(i)\}$ of updated 3D positions is used to refine the camera pose via the PNP algorithm:

$$\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}} = \mathrm{P}N_i\mathrm{P}(\boldsymbol{P}_i^{\mathrm{ICS}_i}, \hat{\boldsymbol{P}}_i^{\mathrm{GCS}}) \,. \tag{18}$$

With the refined camera pose, the global concentrator orientations and normal vectors can be updated as described by Eq. (15). Eqs. (16)–(18) can be repeated iteratively if needed to improve consistency between the estimates. Our experiments indicate that one iteration is sufficient for convergence. In a final step, the concentrator pose estimate is obtained from its orientation estimate and the passpoint estimate:

$$\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) = \begin{pmatrix} \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) & \hat{\boldsymbol{p}}_j^{\mathrm{GCS}}(i) \\ \boldsymbol{0} & 1 \end{pmatrix} \,. \tag{19}$$

Fig. 5 graphically illustrates the quantities that are involved in Step 2 of Problem 1, while Fig. 6 summarizes the solution to Problem 1 in a flowchart. The algorithm to solve Problem 1 can be found in Appendix D.

4.3. Problem 2

Problem 2 aims to estimate the concentrator normal vector $\boldsymbol{n}_j^{\mathrm{GCS}}$ for a given concentrator j under the assumption of a static field, where the true pose of each concentrator does not vary across observations (see Section 4.1.2). Instead of estimating full orientations, Problem 2 focuses on averaging the normal vectors derived from the estimated orientations $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)$ for all images $i \in \mathcal{I}(j)$. This choice is motivated by practical considerations: In solar tower applications, the concentrator's

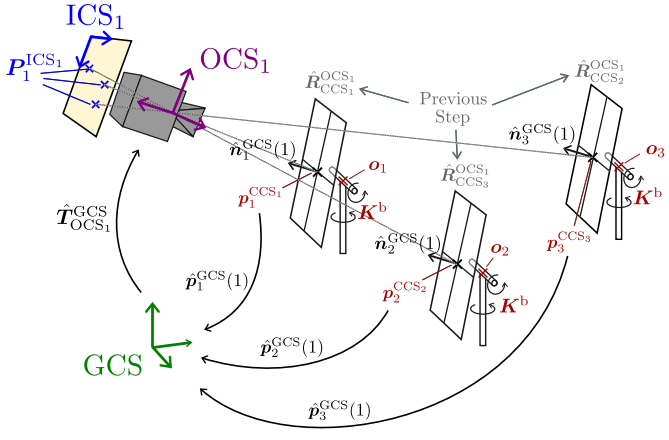


Fig. 5. Overview of Step 2 of Problem 1, in which the camera pose $\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$ and the concentrator orientations $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}$ are jointly refined. This step uses the detected surface passpoints $\boldsymbol{p}_j^{\mathrm{ICS}_i}$ and leverages a-priori knowledge of the heliostat positions $\boldsymbol{o}_j^{\mathrm{GCS}}$, the kinematic model $\boldsymbol{K}$, the known coordinates of the surface passpoints w.r.t. CCS_j and the concentrator orientations $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ from Step 1.

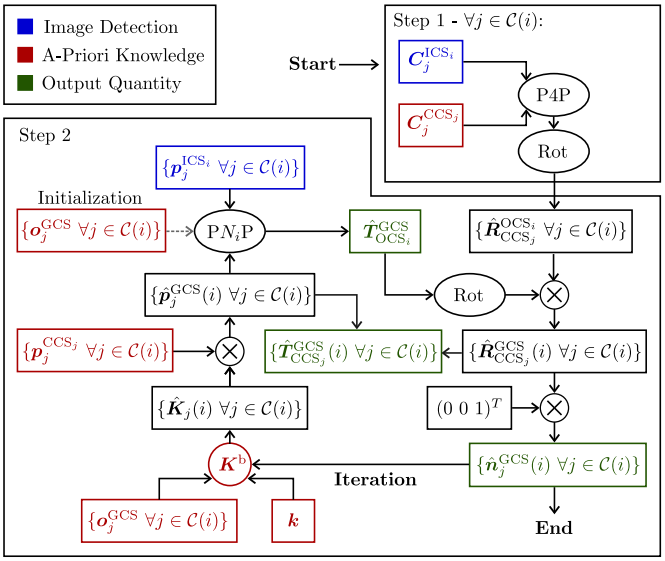


Fig. 6. Flowchart of the solution to Problem 1.

surface normal is typically the quantity of interest when collecting calibration points, as the direction of the reflected sunbeam depends solely on this normal vector and is invariant to rotations around it. In line with this, established approaches such as the camera-target method also output only the normal vector. Note that ARTSCORE-C could also be formulated by averaging orientations. However, such an extension lies beyond the scope of this work.

4.3.1. Weighted averaging

A simple approach is to compute the unweighted mean of the estimated unit normal vectors from each image $i \in \mathcal{I}(j)$. However, the reliability of individual estimates varies due to several factors: As pointed out by Röger et al. [17], the pose between the camera and concentrator affects the estimation uncertainty. Furthermore, pose estimation algorithms suffer from pose ambiguities for coplanar point configurations [18], as demonstrated in Section 5.1. For this reason, Problem 2 not only averages $\hat{\boldsymbol{n}}_j^{\mathrm{GCS}}(i)$ but also considers the perspective under which a given observation was obtained by assigning the weights w_{ij} prior to averaging.

Section 5.2 shows that the uncertainty in the global estimate $\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}$ is dominated by the uncertainty in the relative pose estimate $\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$. Hence, the weight w_{ij} is computed based on the relative pose estimate:

$$w_{ij} = w\left(\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}\right) . \tag{20}$$

These weights are then used in a weighted averaging scheme to compute the final normal vector estimate:

$$\hat{\boldsymbol{n}}_j^{\mathrm{GCS}} = \frac{\sum_{i\in\mathcal{I}(j)} w_{ij} \cdot \hat{\boldsymbol{n}}_j^{\mathrm{GCS}}(i)}{\left|\left| \sum_{i\in\mathcal{I}(j)} w_{ij} \cdot \hat{\boldsymbol{n}}_j^{\mathrm{GCS}}(i) \right|\right|} . \tag{21}$$

To simplify, $w(\cdot)$ may be based on a lower-dimensional descriptor of the relative pose. The next subsection introduces such a descriptor.

4.3.2. Perspective descriptor

In principle, the relative pose $\boldsymbol{T}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ between the camera and the concentrator is described by six degrees of freedom, i.e., three translational and three rotational components. However, many constellations in this six-dimensional pose space may result in the same or similar estimation performances.

This work introduces a lower-dimensional descriptor that captures the aspects of perspective most relevant to estimation performance. Such a descriptor simplifies the analysis of performance trends, eases interpretation and allows for visualization of key insights. This descriptor is later used in both simulation and validation experiments, for example to guide the sampling of reliable observations during validation. In Section 5, it is demonstrated through Monte-Carlo experiments that this descriptor largely predicts the estimation quality of $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$, supporting its use in the subsequent analyses.

Each pose $\boldsymbol{T}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ is represented by the tuple (d_{ij}, ψ_{ij}), where d_{ij} describes the distance between the camera and the concentrator, and ψ_{ij} describes the angle between the vector connecting the camera and the concentrator center, and the normal vector $\boldsymbol{n}_{ij}$ of concentrator j, expressed w.r.t. OCS_i, i.e.,

$$\begin{aligned} d_{ij} &= \left| \boldsymbol{t}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i} \right| , \\ \psi_{ij} &= \arccos\left(\frac{\boldsymbol{t}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i} \cdot \boldsymbol{n}_{ij}}{|\boldsymbol{t}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}| \cdot |\boldsymbol{n}_{ij}|} \right) , \end{aligned} \tag{22}$$

where

$$\begin{aligned} \boldsymbol{t}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i} &= -\mathrm{Trans}\left(\boldsymbol{T}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}\right) , \\ \boldsymbol{n}_{ij} &= \mathrm{Rot}\left(\boldsymbol{T}_{\mathrm{CCS}_i}^{\mathrm{OCS}_j}\right) \cdot (0\ 0\ 1)^T . \end{aligned} \tag{23}$$

4.4. Implementation

The proposed method is implemented in Python. Coordinate transformations rely on the packages NumPy [19] and SciPy [20] for efficient vectorized computation. The visibility graph is represented using SciPy sparse arrays. Camera operations are implemented with the opencv-python package, a Python wrapper for the open-source computer vision library OpenCV [21]. For pose estimation in both Step 1 and Step 2 of Problem 1, the SQPnP algorithm [22] is employed. To enhance robustness and accuracy, the initial SQPnP estimates are refined via a *Levenberg–Marquardt* optimization [23] available in OpenCV, minimizing the reprojection error across all correspondences. The perception module is built with Python and the deep-learning framework PyTorch, and can therefore be integrated seamlessly. For further details on the perception module, see [10].

5. Simulation-based preliminary analysis

This section provides a simulation-based analysis of how noise propagates through the estimation pipeline. In the first part of the analysis, the error distribution of the concentrator orientation estimates $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ is characterized and it is demonstrated that the estimation quality is predicted by the perspective descriptor (see Section 5.1). In the second part of the analysis, it is shown that the error contribution of the camera orientation estimate $\hat{\boldsymbol{R}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$ is small compared to the error contribution of $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ (see Section 5.2). These findings justify the weighting strategy $w(\cdot)$ in Problem 2 to be based solely on the relative perspective descriptor, without the need to consider the camera pose quality.

5.1. Uncertainty analysis of $\hat{\boldsymbol{R}}_{CCS_j}^{OCS_i}$

The perspective-dependent characteristics of $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$ are first illustrated in a simple simulation setup with three distinct relative perspectives: A virtual concentrator with the dimensions $3.22\,\mathrm{m} \times 2.56\,\mathrm{m}$ of the concentrators at the STJ is placed at a fixed distance $d = 60\,\mathrm{m}$ from a virtual camera, whose IOR is set according to the measurement camera described in Section 6.1.2. The concentrator is positioned in the image center, with its surface normal vector initially aligned with the optical

axis of the camera. The concentrator is then rotated around its X-axis by the angles $\psi_1 = 0°$, $\psi_2 = 8°$, and $\psi_3 = 16°$, to obtain three distinct relative perspectives, with the orientations $\boldsymbol{R}^{\mathrm{OCS}}_{\mathrm{CCS}_1}$, $\boldsymbol{R}^{\mathrm{OCS}}_{\mathrm{CCS}_2}$ and $\boldsymbol{R}^{\mathrm{OCS}}_{\mathrm{CCS}_3}$. To simulate noise in the detected corner point image coordinates, four independent error vectors $(e^{\mathrm{ICS}}_{\mathrm{X}}, e^{\mathrm{ICS}}_{\mathrm{Y}})^T$, one for each corner, are sampled $K = 10^5$ times according to a bivariate normal distribution for each relative perspective, i.e.,

$$\begin{pmatrix} e^{\mathrm{ICS}}_{\mathrm{X}} \\ e^{\mathrm{ICS}}_{\mathrm{Y}} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0\,\mathrm{px} \\ 0\,\mathrm{px} \end{pmatrix}, \begin{pmatrix} (2\,\mathrm{px})^2 & 0 \\ 0 & (2\,\mathrm{px})^2 \end{pmatrix} \right) . \tag{24}$$

Each set of four error vectors is added to the image projections of the concentrator's corner points. The P4P algorithm is then applied to every noisy concentrator projection, resulting in the rotation estimates $\hat{\boldsymbol{R}}^{\mathrm{OCS}}_{\mathrm{CCS}_{j,k}}$ for every concentrator $j \in \{1, 2, 3\}$ and estimation run $k = 1, \ldots, 10^5$.

To ease the interpretation, the analysis focuses on the estimation error of the normal vector rather than on the full rotation. The estimated and true normal vectors are

$$\begin{aligned} \hat{\boldsymbol{n}}^{\mathrm{OCS}}_{j,k} &= \hat{\boldsymbol{R}}^{\mathrm{OCS}}_{\mathrm{CCS}_{j,k}} \cdot (0\ 0\ 1)^T, \\ \boldsymbol{n}^{\mathrm{OCS}}_{j} &= \boldsymbol{R}^{\mathrm{OCS}}_{\mathrm{CCS}_{j}} \cdot (0\ 0\ 1)^T . \end{aligned} \tag{25}$$

The angular estimation error is defined as

$$\theta_{j,k} = \arccos\left(\hat{\boldsymbol{n}}^{\mathrm{OCS}}_{j,k} \cdot \boldsymbol{n}^{\mathrm{OCS}}_{j}\right). \tag{26}$$

While $\theta_{j,k}$ provides a compact, one-dimensional measure of the estimation error, it does not capture the full structure of how these errors are distributed. To gain more insight, the true normal vector is also represented in the coordinate system defined by the orientation estimate, i.e.,

$$\boldsymbol{n}^{\mathrm{err}}_{j,k} = \left(\hat{\boldsymbol{R}}^{\mathrm{OCS}}_{\mathrm{CCS}_{j,k}}\right)^T \cdot \boldsymbol{n}^{\mathrm{OCS}}_{j} . \tag{27}$$

This three-dimensional representation makes it possible to cluster the error distribution, as described further below. Fig. 7 shows the scatter plots of the X-component of $\boldsymbol{n}^{\mathrm{err}}_{j,k}$ over the Y-component of $\boldsymbol{n}^{\mathrm{err}}_{j,k}$ for every concentrator $j \in \{1, 2, 3\}$. In addition, it shows the histograms of the angular error for every concentrator $j \in \{1, 2, 3\}$.

The distribution for $\psi_1 = 0°$ shows a ring-like structure. For $\psi_2 = 8°$ and $\psi_3 = 16°$, the distributions exhibit a characteristic bimodal structure, with two distinct clusters: A dominant mode, corresponding to estimates centered around the true value, and a secondary mode containing erroneous estimates. This behavior reflects the pose ambiguity effect described by Schweighofer et al. [18]. As ψ increases, the separation between the two clusters becomes more pronounced, while the probability mass of the secondary cluster decreases. At the same time, the variance within each cluster is reduced, indicating greater consistency in both correct and incorrect estimates under more oblique views.

To further analyze the orientation estimation characteristics, the set of $\boldsymbol{n}^{\mathrm{err}}_{j,k}$ is partitioned using the KMeans algorithm [24] for each j: Two clusters $\mathcal{C}_{\mathrm{true}}$ and $\mathcal{C}_{\mathrm{flip}}$ are identified such that the within-cluster variance is minimized, where $\mathcal{C}_{\mathrm{true}}$ corresponds to all estimates close to the true orientation and $\mathcal{C}_{\mathrm{error}}$ corresponds to all estimates that are "flipped". The probability of a flip p_{flip} is estimated by $|\mathcal{C}_{\mathrm{flip}}|/K$. The amplitude of a flip a_{flip} is estimated by the distance between the cluster centroids, with the cluster centroid defined as the mean of all cluster members. Fig. 8 illustrates the process of clustering the normal vector error distribution for the case $(\psi_3 = 16°, d_3 = 60\,\mathrm{m})$. For visualization purposes, the figure shows only the projections of $\boldsymbol{n}^{\mathrm{err}}_{j,k}$ onto the XY-plane. However, the clustering is performed in the full 3D space.

The three relative poses presented earlier cover only a narrow subset of all possible relative poses. In realistic measurement scenarios, concentrators not only appear in the image center but also toward the image corners. They can exhibit in-plane rotations due to variations in their kinematic configuration. Additionally, the changing position and orientation of the camera further extend the space of possible relative poses. To better reflect this variability, a second simulation samples a broader and more representative set of $J = 5000$ relative poses, with $K = 1000$ orientation estimation runs per sampled pose. The perspectives are chosen such that they cover the space of relative poses present in the validation data (see Section 6.1). For details on the sampling process, refer to Appendix C.

For each sampled relative pose $\boldsymbol{R}^{\mathrm{OCS}}_{\mathrm{CCS}_j}$, the flip probability $\hat{p}_{\mathrm{flip}}$ and the flip amplitude $\hat{a}_{\mathrm{flip}}$ are estimated from the clustered error distribution, as described earlier. Fig. 9 visualizes the resulting simulated $\hat{p}_{\mathrm{flip}}$ and $\hat{a}_{\mathrm{flip}}$

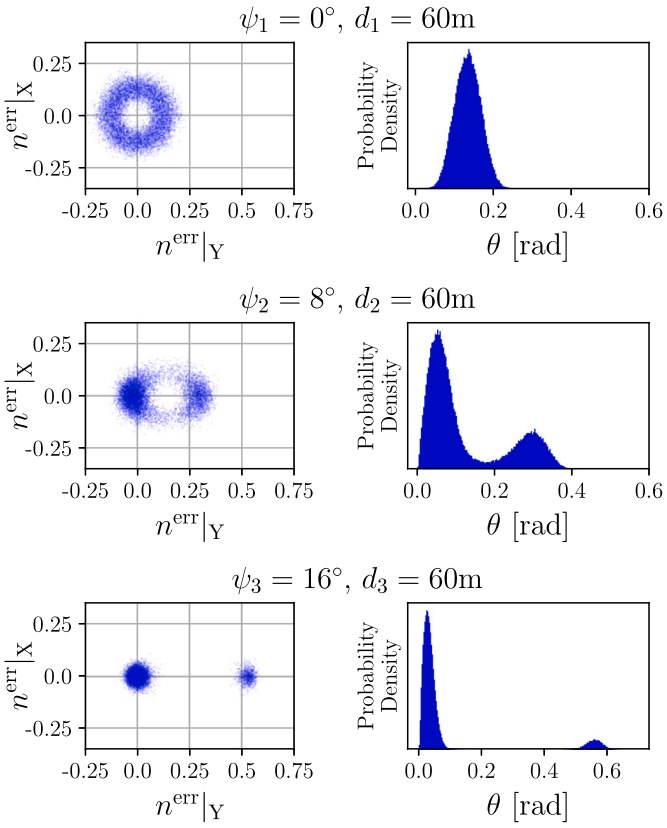


Fig. 7. Scatter plot of X-component over Y-component of $\boldsymbol{n}^{\mathrm{err}}_{j,k}$ (left) and the angular estimation error distributions (right) for the relative poses with $(\psi_1, d_1) = (0°, 60\,\mathrm{m})$ (top), $(\psi_2, d_2) = (8°, 60\,\mathrm{m})$ (center) and $(\psi_3, d_3) = (16°, 60\,\mathrm{m})$ (bottom).

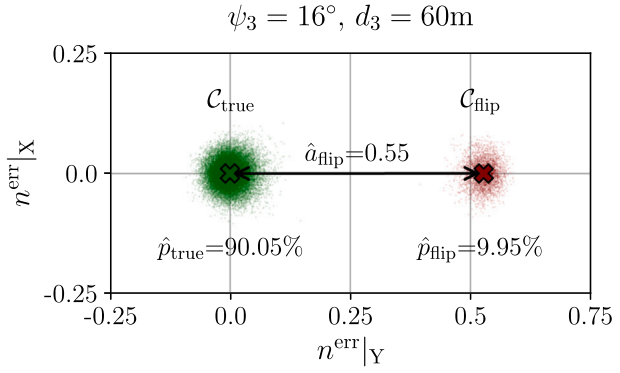


Fig. 8. Scatter plot of X-component over Y-component of $\boldsymbol{n}^{\mathrm{err}}_{j=3,k}$ for the relative pose with $(\psi_3, d_3) = (16°, 60\,\mathrm{m})$. Using the KMeans algorithm, the distribution of $\boldsymbol{n}^{\mathrm{err}}_{j=3,k}$ is classified into two clusters, $\mathcal{C}_{\mathrm{true}}$ and $\mathcal{C}_{\mathrm{flip}}$. Based on these clusters, the metrics $\hat{p}_{\mathrm{flip}}$ and $\hat{a}_{\mathrm{flip}}$ are computed.

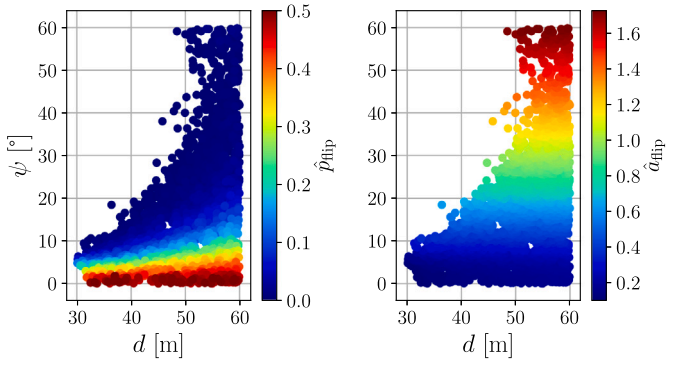


Fig. 9. Flip probability $\hat{p}_{\text{flip}}$ (left) and flip amplitude $\hat{a}_{\text{flip}}$ (right) plotted in color in the d–ψ plane. Each point corresponds to one of 5000 sampled relative pose, with its corresponding value being estimated using 1000 orientation estimates for this relative pose. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in the d–ψ plane. In the left plot, the color of each point represents the corresponding simulated flip probability $\hat{p}_{\text{flip}}$ for the respective (d, ψ). In the right plot, the color indicates the simulated flip amplitude $\hat{a}_{\text{flip}}$ for the respective (d, ψ). Due to the limited number of orientation estimation runs per relative pose, a distinct flip cluster C_{flip} may not emerge when the flip probability is very low. In such cases, the KMeans algorithm splits the true cluster C_{true} into two clusters with centroids that lie very close together. To avoid misinterpreting these artifacts as meaningful flips, these cases are excluded in Fig. 9, showing only those poses with an estimated flip amplitude $\hat{a}_{\text{flip}} > 0.1$.

The left plot shows that the flip probability decreases as ψ increases and d decreases. In contrast, the right plot indicates that the flip amplitude increases with ψ and remains unaffected by d. From Fig. 9, it is not immediately clear which relative perspectives are optimal: Ideally, both the flip probability and amplitude should be low. However, the simulation results suggest a trade-off where reducing one tends to increase the other. To evaluate the overall estimation quality as a function of d and ψ, the mean angular estimation error

$$\bar{\theta}_j = \frac{1}{K} \sum_{k=1}^{K} \theta_{j,k} \tag{28}$$

is computed for each relative pose j, where $\theta_{j,k}$ is computed according to Eq. (26). The resulting colored scatter plot is depicted in Fig. 10.

As for the flip probability, a diagonal structure across the d–ψ plane can be observed, with the best estimation quality in the top-left region (short distance, large angle) and the worst in the bottom-right region (long distance, small angle). While high ψ values lead to larger flip amplitudes, their impact on the overall estimation quality is small, as such large amplitude flips arise with low probability. Still, the effect of large flip amplitudes should be kept in mind as a potential explanation for concentrator orientation estimation outliers.

Furthermore, the plots suggest that the estimation quality is well captured by the perspective descriptor defined by (d, ψ): For any given pair of values, the associated metrics are effectively determined, with no visible residual variance in directions other than the d-axis and the ψ-axis. This indicates that the descriptor reliably summarizes the relevant geometric conditions influencing estimation performance in Step 1 of Problem 1.

5.2. Uncertainty analysis of $\hat{\boldsymbol{R}}_{GCS}^{OCS_i}$

In Step 2 of Problem 1, the concentrator orientation estimates are transformed from OCS to GCS, i.e.,

$$\hat{\boldsymbol{R}}_{\text{CCS}_j}^{\text{GCS}}(i) = \hat{\boldsymbol{R}}_{\text{OCS}_i}^{\text{GCS}} \cdot \hat{\boldsymbol{R}}_{\text{CCS}_j}^{\text{OCS}_i} . \tag{29}$$

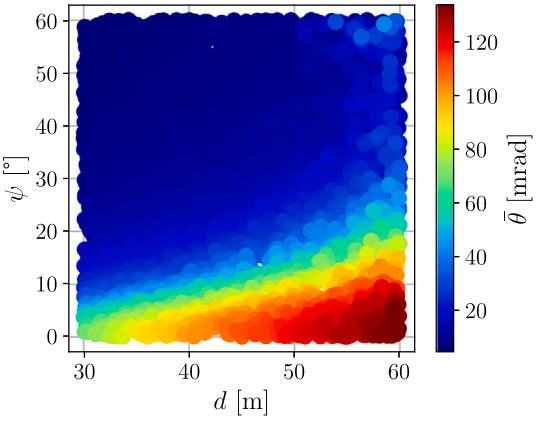


Fig. 10. Mean orientation error $\bar{\theta}$ plotted in color in the d–ψ plane. Each point corresponds to one of 5000 sampled relative pose with its corresponding value being estimated using 1000 orientation estimates for this relative pose. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Since the camera orientation estimate $\hat{\boldsymbol{R}}_{\text{OCS}_i}^{\text{GCS}}$ is itself subject to noise, this transformation introduces additional uncertainty in the final estimate $\hat{\boldsymbol{R}}_{\text{CCS}_j}^{\text{GCS}}$. This subsection analyzes the uncertainty contribution of both steps to the final estimate and concludes that for typical noise levels, the uncertainty introduced by Step 2 is negligible compared to the uncertainty introduced by Step 1.

The error of an orientation estimate $\hat{\boldsymbol{R}}$ w.r.t. the true orientation $\boldsymbol{R}$ can be expressed by the geodesic distance:

$$d(\hat{\boldsymbol{R}}, \boldsymbol{R}) = \arccos\left(\frac{\text{Tr}(\hat{\boldsymbol{R}}^T \cdot \boldsymbol{R} - 1)}{2} \right) . \tag{30}$$

Since $\hat{\boldsymbol{R}}_{\text{CCS}_j}^{\text{OCS}_i}$ is estimated in Step 1 and $\hat{\boldsymbol{R}}_{\text{OCS}}^{\text{GCS}}$ is estimated in Step 2, we denote the geodesic distances

$$\begin{aligned} d(\hat{\boldsymbol{R}}_{\text{CCS}_j}^{\text{OCS}_i}, \boldsymbol{R}_{\text{CCS}_j}^{\text{OCS}_i}) &= \theta_{i,j}^{(1)} , \\ d(\hat{\boldsymbol{R}}_{\text{OCS}_i}^{\text{GCS}}, \boldsymbol{R}_{\text{OCS}_i}^{\text{GCS}}) &= \theta_{i}^{(2)} , \\ d(\hat{\boldsymbol{R}}_{\text{CCS}_j}^{\text{GCS}}(i), \boldsymbol{R}_{\text{CCS}_j}^{\text{GCS}}(i)) &= \theta_{i,j}^{\text{final}} . \end{aligned} \tag{31}$$

The statistics of $\theta_i^{(1)}$ and $\theta_{i,j}^{(2)}$ are then obtained by running ARTSCORE-C on a simulated flight, replicated from a real measurement flight.

1. The simulated concentrators are aligned using the kinematic forward pass $\boldsymbol{K}^{\text{f}}$ under the assumption of the kinematic parameterization $\boldsymbol{k}_j$ (obtained using the camera–target method) and the heliostat origins $\boldsymbol{o}_j$.
2. The simulated camera is positioned on the trajectory $\boldsymbol{T}_{\text{OCS}_i}^{\text{GCS}}$, with each ground truth camera pose estimated by ARTSCORE-C on real validation data. The collection of this validation dataset is outlined in Section 6.1. This way, the simulation analysis is performed under the assumption of a real flight trajectory.
3. Corner detections are then generated by reprojecting the simulated concentrator corners onto the ICS for each camera pose $\boldsymbol{T}_{\text{OCS}_i}^{\text{GCS}}$.

This yields a number of $I = 653$ simulated images, each with a number of $N_i \times 4$ concentrator corner point projections $\boldsymbol{C}_j^{\text{ICS}_i}$, where N_i denotes the number of visible concentrators in image i. In addition to the true camera orientation, the true relative orientations $\boldsymbol{R}_{\text{CCS}_j}^{\text{OCS}_i}$ are computed.

Prior to running ARTSCORE-C, noise is sampled for both the assumed heliostat origins $\boldsymbol{o}_j$ as well as for the reprojected concentrator corner points $\boldsymbol{C}_j^{\mathrm{ICS}_i}$:

1. As described in Eq. (16), ARTSCORE-C maintains a list of assumed heliostat origins, which deviate from the true heliostat origins. In [25], the error magnitude is estimated to be upper-bounded by 12.8 mm for the heliostat field at the STJ. Assuming an isotropic Gaussian for their error (i.e., $\boldsymbol{e}_{\boldsymbol{o}_j} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{\mathrm{GCS}}^2 \cdot \boldsymbol{I}_3)$), we choose $\sigma_{\mathrm{GCS}} = 12.8\,\mathrm{mm}/\sqrt{3} \approx 7.5\,\mathrm{mm}$ to simulate noise along the X-, Y- and Z-axis.
2. In addition, the perception module is subject to noise. We sample noise in the ICS for each projected corner point according to an isotropic Gaussian, i.e., $\boldsymbol{e}_{\boldsymbol{C}_{j,c}^{\mathrm{ICS}_i}} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{\mathrm{ICS}}^2 \cdot \boldsymbol{I}_2)$. The standard deviation is set to $\sigma_{\mathrm{ICS}} = 1\,\mathrm{px}$ along both, the ICS X-axis and Y-axis, as this roughly matches the perception noise levels identified on real data, as outlined in Section 6.2.

For each noisy simulated image i and under the assumption of noisy heliostat origins, Problem 1 is solved, yielding a camera orientation estimate, as well as a list of relative orientation estimates for all $j \in C(i)$ with $i = 1, \ldots, 653$. Note that the detected passpoints $\boldsymbol{p}_j^{\mathrm{ICS}_i}$ are computed from the noisy concentrator corner detections $\boldsymbol{C}_j^{\mathrm{ICS}_i}$, using the algorithm described in Appendix D. Under the assumption of small-angle perturbations and statistical independence between the noise sources, the variance of the final orientation estimate can be approximated by the sum of the individual variances [26]:

$$\sigma_{\theta_{i,j}^{(\mathrm{final})}}^2 = \sigma_{\theta_{i,j}^{(1)}}^2 + \sigma_{\theta_i^{(2)}}^2 , \tag{32}$$

To empirically compare the two sources of uncertainty, the ratio of the geodesic errors is computed for every concentrator observation j for every image i, resulting in a list of ratios

$$r = \left\{ \frac{\theta_{i,j}^{(1)}}{\theta_i^{(2)}} \;\middle|\; i \in \{1, \ldots, I\},\ j \in C(i) \right\} . \tag{33}$$

Fig. 11 shows the histogram of all squared ratios obtained in the simulated flight. The total number of observations amounts to $\sum_{i=1}^{653} |C(i)| = 26485$.

The mean value of the ratio $r_{i,j}$ for all observations (i, j) amounts to $\bar{r} \approx 0.03$, with 95% of all ratios lying below 0.1. This finding suggests that the contribution of the camera orientation uncertainty to the uncertainty in the global orientation estimate is negligible compared to that of the object orientation estimation in the OCS. Note that the probability mass of 1.5% for $r \geq 0.25$ corresponds to object orientation estimates that happen to be unusually accurate, likely due to favorable noise realizations. This behavior is expected when considering a large number of observations and does not contradict the overall conclusion.

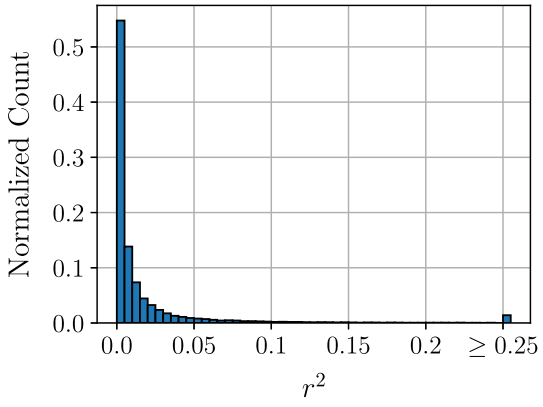


Fig. 11. Histogram of the squared ratio r between the geodesic error of the camera orientation estimate and the concentrator orientation estimate for every concentrator $j \in C(i)$ in every simulated image $i = 1, \ldots, 653$.

Based on the experiments in Sections 5.1 and 5.2, the descriptor (d, ψ) of the relative pose between the concentrator and the camera is concluded to be a good predictor of the overall orientation estimation error of $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)$. However, these experiments are conducted under assumptions specific to the STJ, with noise levels representative of the perception module of Broda et al. [10]. An analysis for a different heliostat field with different perception noise may lead to a different conclusion.

6. Validation methodology

This section describes the methodology used to validate the presented method. Section 6.1 presents the setup for acquiring the raw validation data. Section 6.2 describes how a noise model is fitted to the perception noise to characterize the underlying noise process and remove systematic biases from the detected corner points. Section 6.3 explains how the validation data is processed to emulate the conditions of real measurement flights. An additional data-processing step selects a subset of observations to enhance the method's performance. Finally, Section 6.4 summarizes the notation used to denote a filtered validation dataset.

6.1. Validation data measurement setup

The validation dataset used in this work is designed to reflect realistic operating conditions encountered during UAV-based data acquisition: The concentrators were oriented as described in Section 6.1.1. The validation camera settings and UAV flight path were chosen as described in Section 6.1.2. After image acquisition, the corner points of the concentrators are detected and uniquely identified in each image, as detailed in Section 6.1.3. Currently, this processing is performed offline (i.e., after the flight) to ensure reproducibility and facilitate detailed analysis. However, real-time execution is feasible in principle. It could be achieved either by transmitting image data to a ground station via a communication link or by integrating a sufficiently powerful onboard companion computer. The corner point detection step currently represents the most computationally intensive part of the pipeline: While the current detection algorithm already achieves processing times under one second per image and is suited for real-time operation, future work could explore algorithmic refinements or hardware acceleration to further reduce latency and enable more efficient in-flight deployment.

6.1.1. Solar field configuration

The heliostat field was divided into six groups of concentrators, denoted by the group index $g = 1, \ldots, 6$. Within each group g, all concentrator normal vectors were aligned in the same direction, while vectors from different groups pointed toward distinct directions. This setup was chosen to ensure that multiple validation concentrators shared the same orientation, allowing verification that the method yields consistent results under similar viewing conditions. In effect, this discretization of the orientation space allows the statistical evaluation of the method's performance across the different heliostat groups. While the exact viewing angles still vary within each group due to several factors, the configuration provides a practical basis for analyzing the method's behavior under repeatable, orientation-dependent conditions. Fig. 12 shows the location of each group g in the heliostat field.

Fig. 13 shows the normal vectors of each group in the azimuth-elevation plane, where az_g^{ref} and el_g^{ref} denote the reference azimuth and elevation angles obtained from the field operator. According to the *SolarPACES Guideline for Heliostat Performance Testing* [12,13], azimuth

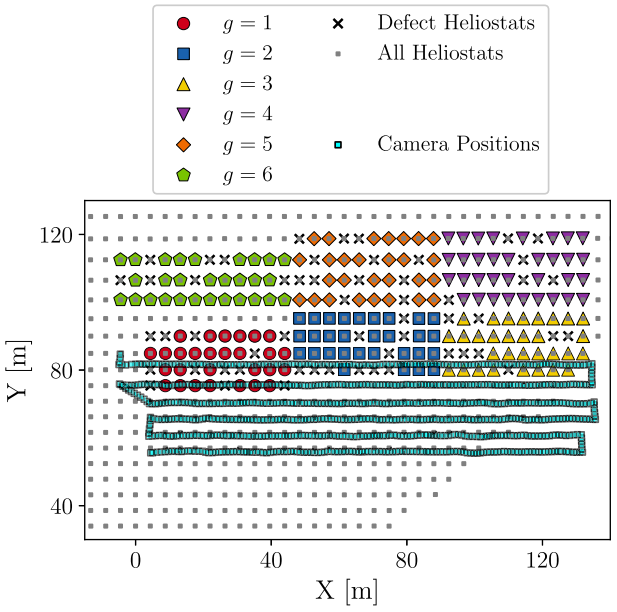


Fig. 12. Overview of the camera flight path and the locations of the six groups of heliostats within the STJ, used for validating the presented method.

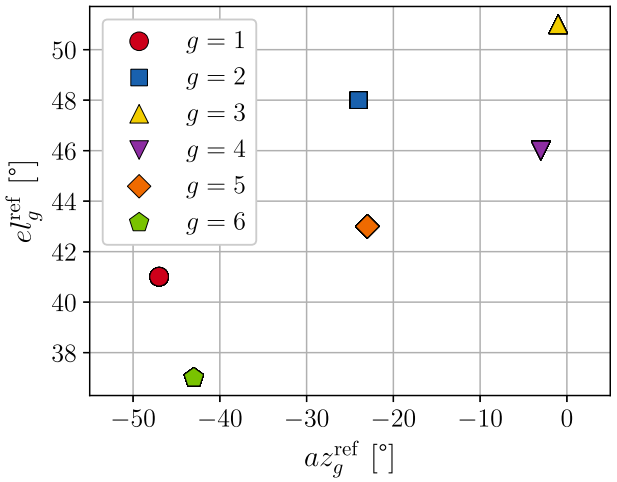


Fig. 13. Scatter plot of the reference azimuth angles az_g^{ref} over the reference elevation angles el_g^{ref} for the validation groups $g = 1, \ldots, 6$. The convention is chosen in accordance with [12,13]: Azimuth is measured clockwise from the negative Y-axis in the horizontal plane. Elevation ranges from −90° (concentrator facing down) to 90° (concentrator facing up).

is defined as the angle between the concentrator's normal vector and the negative y-direction of the GCS, projected onto the horizontal plane and measured clockwise. Elevation is the angle between the concentrator normal vector and the horizontal plane of the GCS, where a mirror surface facing the ground corresponds to −90° and one facing the sky corresponds to 90°.

6.1.2. Measurement camera and flight path

The measurement images were acquired using a DJI Zenmuse P1 camera mounted on a DJI Matrice 300 UAV platform. The camera features a high-resolution sensor of 8192 px × 5460 px with a physical size of 35.9 mm × 24 mm, yielding a pixel size of approximately 4.4 µm. It was operated with a fixed focal length of 35 mm, resulting in a horizontal field of view (HFOV) of 54.3°. The focus was set to infinity to ensure consistent sharpness across the scene and between images. The exposure mode was set to manual to maintain uniform brightness and image quality under varying lighting conditions. A global mechanical shutter was used to preserve geometric accuracy. The camera IOR was determined through photogrammetric calibration using the commercial software AICON 3D STUDIO. This calibration employed a dedicated flight over a set of AICON optical markers. From the calibration images, AICON 3D STUDIO simultaneously estimated the 3D marker positions, the camera's exterior orientations (EORs), and its IOR.

During the measurement flight, the UAV maintained a constant altitude of $h_{\text{cam}} = 35$ m and followed a lawnmower pattern flight path, illustrated in light blue in Fig. 12. The UAV traversed east–west and west–east transects in alternating passes, gradually progressing northward with each loop. The onboard camera was gimbal-stabilized with a fixed pitch angle of 30°, where 0° corresponds to a nadir (downward-facing) view and 90° to a horizontal view. Both the yaw and roll angles were actively stabilized to remain near 0°, ensuring that the camera consistently pointed north and that its image plane remained level with the horizon. This configuration minimized in-plane rotation across the captured images, thereby simplifying the detection and identification of concentrator corner points, outlined in the following subsection. More details on the flight path are discussed in Section 7.5. Fig. 14 shows an exemplary image recorded during the measurement flight.

6.1.3. Corner point detection and identification

The 2D image coordinates of the four concentrator corner points are detected in every image by the AI-based perception module presented by Broda et al. [10]. The detected corner points are identified locally by assigning local IDs based on visual tracking across image sequences. These local IDs enable consistent identification of the same corner points across multiple images. By combining field geometry data with an initial camera pose estimate, these local IDs are mapped to global heliostat IDs, referring to fixed physical locations in the field. As more global IDs become known, they can be used to estimate the camera pose for new images, enabling the assignment of global IDs to previously unassigned local IDs based on their projected positions relative to the field layout. This process is refined iteratively to increase the set of globally identified points. Alternatively, global identification can be performed directly using a hardware-based camera pose estimate (e.g., RTK and gimbal orientation). This pose is used to reproject the known

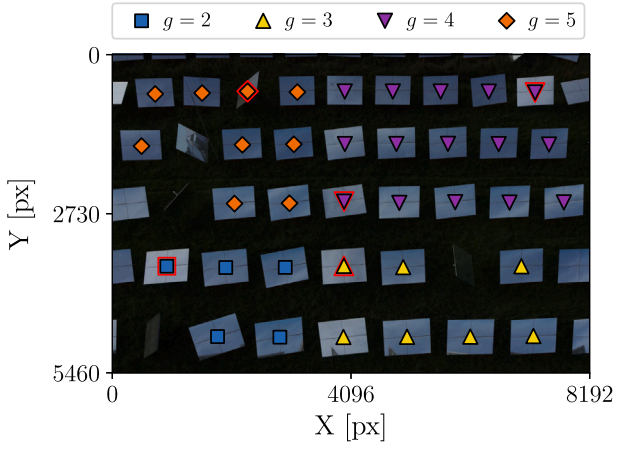


Fig. 14. Exemplary image recorded by the measurement camera during a validation flight at the STJ. For reference, each concentrator has a mirror surface area of 3.22 m×2.56 m. Detected concentrators are highlighted using markers and color coding consistent with Figs. 12 and 13. Concentrators additionally marked in red were detected by the perception module but excluded from the validation, as no reference normal vector was available due to defects identified by the field operator. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

field geometry into the image, allowing corner points to be associated with their corresponding global IDs.

6.2. Perception noise model

To analyze the noise introduced by the perception module in the ICS, five concentrators are selected for each of the six validation groups, resulting in a total of 30 unique concentrators. For each concentrator, the corner point detections are manually labeled for 30 images each. The difference between the detected and manually labeled corner positions yields a 2D error vector for each corner, expressed in pixel units. Collecting these vectors across the entire annotated dataset results in four sets of 900 2D error vectors, one for each of the four corner points. The scatter plots of all 2D error vectors for each corner are depicted in Fig. 15.

To statistically model the perception noise, the parameters of a bivariate normal distribution are estimated from each collection of error vectors. Let $c \in \{\mathrm{UL}, \mathrm{UR}, \mathrm{LL}, \mathrm{LR}\}$ denote the four corners of the concentrator, with UL = upper left, UR = upper right, LL = lower left and LR = lower right. For each corner c, the set of vectors $\{\boldsymbol{e}_{c,i}\}_{i=1}^{900}$ with $\boldsymbol{e}_{c,i} = (e_\mathrm{X}, e_\mathrm{Y})_{c,i}^T$ represents the error between the automatically detected and manually corrected image coordinates. From each set, a 2D mean vector $\hat{\boldsymbol{\mu}}_c$ and a 2×2 covariance matrix $\hat{\boldsymbol{\Sigma}}_c$ are estimated:

$$\hat{\boldsymbol{\mu}}_c = \sum_{i=1}^{900} \boldsymbol{e}_{c,i}\,, \quad \hat{\boldsymbol{\Sigma}}_c = \frac{1}{899} \sum_{i=1}^{900} (\boldsymbol{e}_{c,i} - \hat{\boldsymbol{\mu}}_c)(\boldsymbol{e}_{c,i} - \hat{\boldsymbol{\mu}}_c)^T\,. \tag{34}$$

The estimated distributions are depicted in Fig. 16, and the corresponding parameters for each distribution are listed in Appendix B. Note that the process of manually labeling the concentrator corners is itself subject to noise. Consequently, the error distributions also reflect annotation uncertainty that cannot be attributed solely to the perception module.

While the perception module utilized in this work exhibits specific corner-dependent biases, different or future perception systems exhibit different statistical properties. To reduce the impact of system-specific biases on the validation of the proposed method, the corner detections are debiased prior to applying the method by subtracting the estimated error mean $\hat{\boldsymbol{\mu}}_c$ from each detection.

6.3. Validation data processing

As described in Section 4.1, the visibility relationship between images i and concentrators j is represented by the visibility graph $\boldsymbol{V}$.

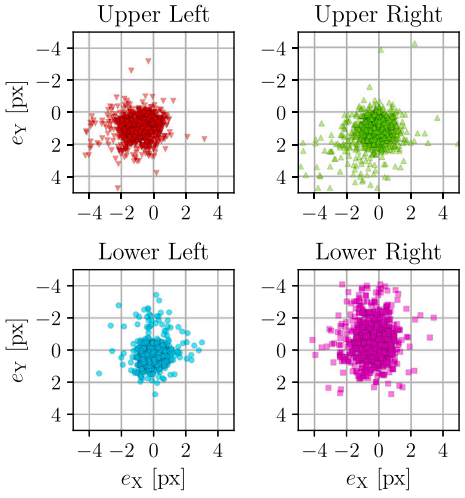


Fig. 15. Scatter plots of the 2D error vectors for each corner point, determined by comparing the image coordinates detected by the perception module with their manually labeled counterparts.

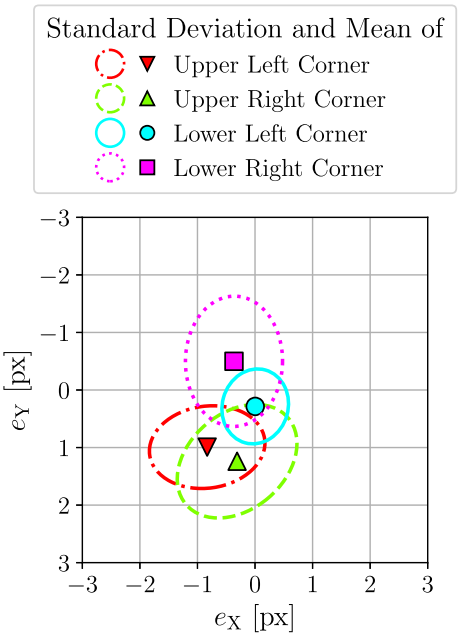


Fig. 16. Illustration of the estimated corner detection error distribution for each corner. The ellipses depict the 1σ contour lines.

After correcting for biases in the concentrator corner point detection (see Section 6.2), $\boldsymbol{V}$ is generated by solving Problem 1 for each validation image. The resulting visibility graph $\boldsymbol{V}$ is then refined in three sequential processing steps:

1. Image filtering (IF) (see Section 6.3.1): Iterates over the image nodes i in $\boldsymbol{V}$, identifying images of impaired quality via a plausibility check based on a priori field data knowledge. For each image failing this check, the corresponding image node and all its incident edges are removed from $\boldsymbol{V}$.
2. Representative observation filtering (ROF) (see Section 6.3.2): Iterates over the concentrator nodes j, selecting a fixed number R of valid observations per concentrator to ensure representativeness of typical measurement flights in both observation count and spatial diversity. For any concentrator observed in fewer than R images, its corresponding concentrator node and all its incident edges are removed entirely from $\boldsymbol{V}$.
3. Descriptor-based filtering (DBF) (see Section 6.3.3): Iterates over the concentrator nodes j retained after ROF, selecting a subset of D observations in order to improve the method's performance.

6.3.1. Image filtering

For a given image i, the accuracy of both the camera pose estimate $\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$ and all concentrator orientation estimates $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}$ for $j \in C(i)$ is impaired by outliers in the concentrator orientation estimates $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$. As these estimates depend on the four concentrator corner points detected by the AI-based perception module, a filter is applied to identify large errors in the detected corner points: First, the concentrator pose estimates $\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}$ are used to estimate the corner point positions w.r.t. the GCS for a given image i. These points are projected onto the ICS using the camera pose estimate $\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$ and the camera IOR. The image coordinates obtained from this projection are then compared to the perception module's detections.

Fig. 17 shows examples of concentrators alongside their detected and estimated corner points, as well as the reprojection error for each corner. Large errors can be attributed in part to the fact that the perception module was originally trained on a different concentrator geometry and

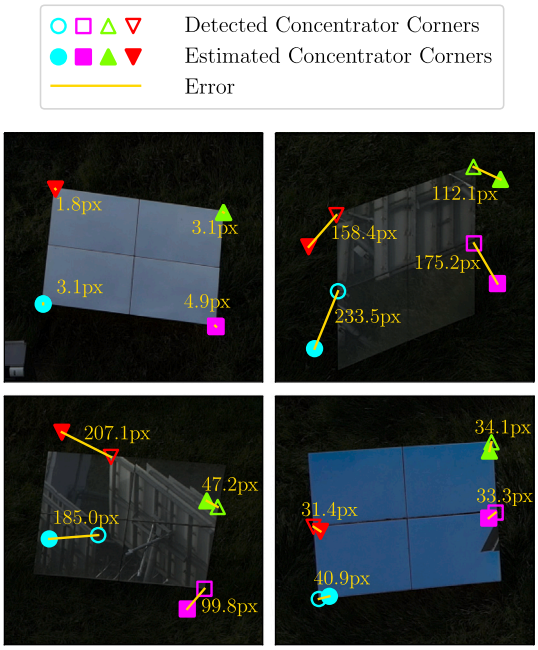


Fig. 17. Examples of concentrators and their detected and estimated corner points. The upper left plot shows a case where all corner detections (unfilled markers) closely match the estimated points (filled markers), and the image is included in the analysis. The remaining plots show cases with large mismatches between detected and estimated points, resulting in the corresponding images being excluded from the subsequent analysis. The error in pixel units is depicted by a yellow line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

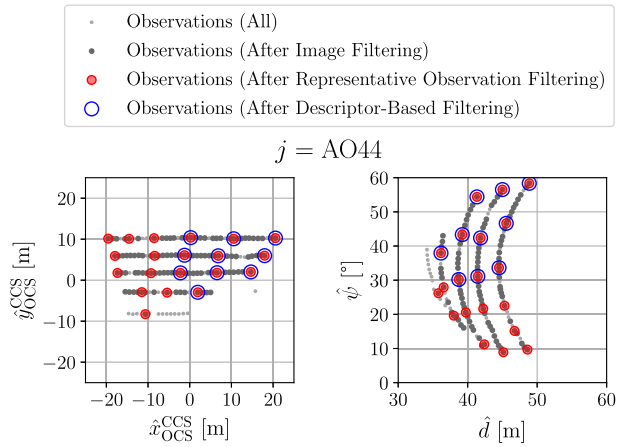


Fig. 18. Illustration of the ROF and DBF for the exemplary concentrators j = AO44 with $R = 20$ and $D = 10$. For the ROF step, observations are selected such that their corresponding camera poses cover the XY-plane of CCS_j as uniformly as possible. For the DBF step, the ROF-processed observations are ordered by their descriptor ψ, and the D observations with the highest ψ values are selected.

under lighting conditions that differed from the measurement setup, where many images are underexposed. While already low, the frequency of such incorrect detections could likely be further reduced by fine-tuning the model on data that more accurately reflects the actual measurement conditions.

Images for which at least one corner point error exceeds a threshold of τ_{px} are excluded from the subsequent analysis, i.e., their corresponding image node is removed from $\boldsymbol{V}$ along with all its incident edges. Note that this strategy leads to a higher image exclusion rate than the perception quality suggests: While misdetections of individual concentrators occur rarely, even a single error among all visible concentrators in a given image (i.e., roughly 30 to 50) can lead to the exclusion of the entire image. In principle, it would be possible to identify the specific concentrator responsible for the large error and rerun Problem 1 with that concentrator discarded, thereby reducing the number of excluded images. However, for the present work the simpler strategy of excluding entire images was chosen, as it still left a sufficient number of images after this filtering step.

6.3.2. Representative observation filtering

To ensure that the validation reflects realistic measurement conditions and allows for a fair comparison between concentrators, a fixed number of images R is selected for each validated concentrator. Here, R denotes the number of images in which a concentrator is expected to appear during a measurement flight. A derivation of R as a function of the camera parameters and variables describing the flight path is provided in Section 7.5.

The objective is to select R observations for each concentrator j from the raw validation data such that the associated camera poses are representative of those encountered during a typical measurement flight. To achieve this, all available camera poses are first expressed in the estimated concentrator coordinate system CCS_j. R poses are chosen so that their projections onto the concentrator XY-plane are as uniformly distributed as possible, reflecting the observation patterns characteristic of a lawnmower pattern flight path. Selection is performed using a greedy algorithm: Starting from an empty set, poses are added iteratively, each chosen to maximize the minimum distance to all previously selected poses in the XY-plane of CCS_j. The left plot of Fig. 18 illustrates the ROF processing step with $R = 20$ for an exemplary concentrator. Note that concentrators observed in less than R images are removed from $\boldsymbol{V}$ along with all their adjacent edges.

6.3.3. Descriptor-based filtering

As the simulation-based analysis indicates that larger values of the angle ψ are associated with lower orientation estimation errors (see Section 5.1), the additional DBF step is applied: For each concentrator j, the D observations with the highest $\hat{\psi}_{ij}$ values are retained. This choice is motivated by the finding that relying on more reliable, low-variance observations generally outweighs the drawback of averaging over fewer observations, which in principle increases estimation variance. By prioritizing perspectives that are less susceptible to pose ambiguity and high variance, this step improves the robustness and reliability of the averaged orientation estimates.

In light of Eq. (20), this procedure can be interpreted as a weighting function $w(\cdot)$ that only takes the values 0 or 1: Each observation is either fully included or completely discarded. This binary selection rule makes the method transparent and easy to analyze, but it also represents a simplification. More advanced strategies could assign continuous weights that take into account both $\hat{\psi}_{ij}$ and $\hat{d}_{ij}$, potentially improving performance. The present binary approach is deliberately chosen as a simple and practical baseline: It enforces exactly D observations per validated concentrator based only on $\hat{\psi}_{ij}$, ensuring fairness and comparability across the dataset. The right plot of Fig. 18 illustrates the retained observations in the d–ψ plane after the DBF step for an exemplary concentrator.

6.4. Notation for filtered validation data

The validation data resulting from the processing steps IF, ROF, and DBF depends on the choice of the filtering parameters: The pixel plausibility threshold τ_{px}, the number of representative observations R, and the number of descriptor-selected observations D. Throughout this work, $\tau_{\mathrm{px}} = 10\,\mathrm{px}$ is chosen: Initial experiments show this effectively filters

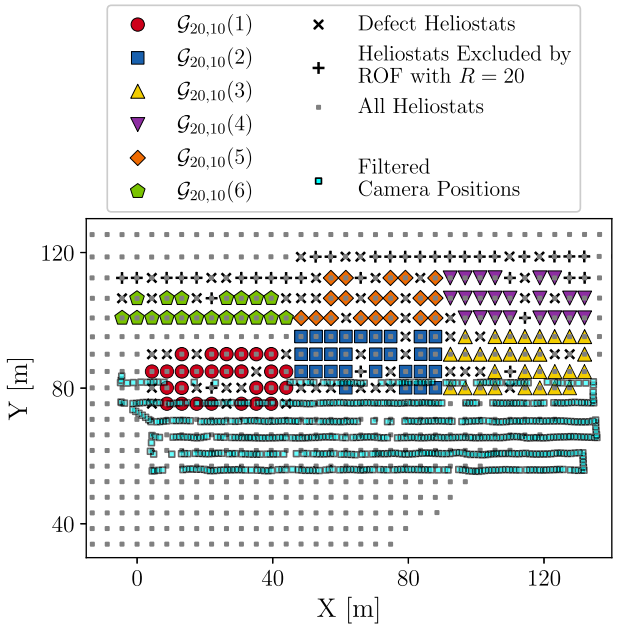


Fig. 19. Overview of the resulting validation dataset after applying IF with $\tau_{\text{px}} = 10\,\text{px}$, ROF with $R = 20$ and DBF with $D = 10$.

out images with clear misdetections, while retaining a sufficient number of valid images in the dataset. To facilitate subsequent analyses, the following notation is used:

- The set of concentrators j in each validation group $g = 1, \dots, 6$ after filtering is denoted by $\mathcal{G}_{R,D}(g)$.
- The set of all images observing concentrator j after filtering is denoted by $\mathcal{I}'_{R,D}(j)$.
- Note that $|\mathcal{I}'_{R,D}(j)| = D$ for all $j \in \mathcal{G}_{R,D}(g)$ with $g = 1, \dots, 6$.

Fig. 19 shows the filtered validation dataset for the exemplary case of $R = 20$, $D = 10$.

7. Validation results and discussion

Due to the complexity of the study and the layered nature of the validation metrics, this section combines results and discussion rather than treating them separately. The analysis starts with the simple case of a single concentrator and progresses toward multiple groups of concentrators. To help readers navigate the notation, all validation metrics are summarized in a table in Appendix A, which can be used as a quick reference throughout the section.

Section 7.1 introduces the validation metrics for a single exemplary concentrator. In Section 7.2, the analysis is extended to a representative group of concentrators, additional group-wise metrics are defined, and the relationship between concentrator-wise and group-wise metrics is examined. Section 7.3 reports these metrics for all six validation groups, providing a complete overview of the results and the corresponding averaged values.

To maintain clarity during this gradual introduction of metrics, the parameter combination $(R, D) = (20, 10)$ is fixed in these first three subsections. The value $R = 20$ is selected as a practical choice: It offers good validation performance while requiring a reasonable number of observations, meaning data collection can tolerate fast drone flights or low image frequencies. Given $R = 20$, the value $D = 10$ is chosen because it yields the best overall results. This is demonstrated in Section 7.4, which investigates the effect of varying (R, D) across all validated concentrators and explores the achievable performance by also considering larger values of R together with their corresponding optimal D. Section 7.5 concludes this section with an analysis of the expected flight time and practical limitations.

For readability, the notations $\mathcal{G}_{R,D}(g)$ and $\mathcal{I}'_{R,D}(j)$ are abbreviated as $\mathcal{G}(g)$ and $\mathcal{I}'(j)$ unless stated otherwise.

7.1. Concentrator-wise analysis

Let $\hat{az}_j(i)$ and $\hat{el}_j(i)$ denote the azimuth and elevation angles of the estimated normal vector for concentrator j, based on image i. Then the estimated azimuth angle $az_{\text{ART},j}$ and elevation angle $el_{\text{ART},j}$ are obtained as

$$\begin{aligned} az_{\text{ART},j} &= \frac{1}{D} \sum_{i \in \mathcal{I}'(j)} \hat{az}_j(i) \,, \\ el_{\text{ART},j} &= \frac{1}{D} \sum_{i \in \mathcal{I}'(j)} \hat{el}_j(i) \,. \end{aligned} \tag{35}$$

The azimuth and elevation biases are defined as

$$\begin{aligned} b^{az}_{\text{ART},j} &= az_{\text{ART},j} - az^{\text{ref}}_j \,, \\ b^{el}_{\text{ART},j} &= el_{\text{ART},j} - el^{\text{ref}}_j \,, \end{aligned} \tag{36}$$

where az^{ref}_j and el^{ref}_j denote the corresponding reference azimuth and elevation angles. Assuming that the images $i \in \mathcal{I}'(j)$ yield i.i.d. observations, the uncertainties of $az_{\text{ART},j}$ and $el_{\text{ART},j}$ are given by [27]

$$\begin{aligned} u^{az}_{\text{ART},j} &= \sqrt{\frac{1}{D \cdot (D-1)} \sum_{i \in \mathcal{I}'(j)} \left(\hat{az}_j(i) - az_{\text{ART},j}\right)^2} \,, \\ u^{el}_{\text{ART},j} &= \sqrt{\frac{1}{D \cdot (D-1)} \sum_{i \in \mathcal{I}'(j)} \left(\hat{el}_j(i) - el_{\text{ART},j}\right)^2} \,. \end{aligned} \tag{37}$$

Fig. 20 shows the aforementioned quantities for an exemplary concentrator with the internal identifier j = AO44, using $R = 20$, $D = 10$.

The black ellipse in Fig. 20 indicates the region in which the angular error exceeds a threshold of 10 mrad. The larger radius in the azimuth direction than in the elevation direction results from the cosine correction:

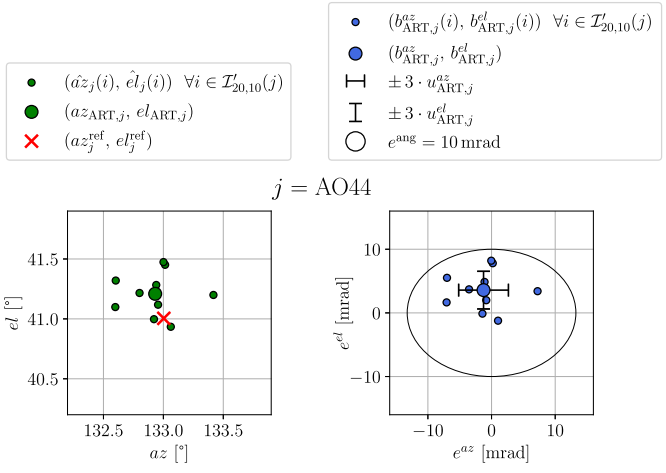


Fig. 20. Left: Scatter plot of the elevation estimates over the azimuth estimates for concentrator j = AO44 in degrees. The small green circles represent the image-wise estimates, while the large circle represents the averaged estimate. Right: Scatter plot of the elevation estimation errors over the azimuth estimation errors for concentrator j = AO44 in milliradians. The small blue circles represent the image-wise estimation errors, while the large blue circle represents the error of the average estimate. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

For small errors, the angular error between the estimated and reference normal vector can be approximated by the azimuth and elevation error:

$$b^{\text{ang}}_{\text{ART},j} = \arccos(\hat{\boldsymbol{n}}^{\text{GCS}}_j \cdot \boldsymbol{n}^{\text{Ref}}_j) \approx \sqrt{\left(b^{az}_{\text{ART},j} \cdot \cos(el^{\text{ref}}_j)\right)^2 + \left(b^{el}_{\text{ART},j}\right)^2}. \tag{38}$$

According to [2], an estimation error below 3 to 10 mrad is required for a system to qualify as a coarse calibration system. Thus, all estimates lying outside the ellipse correspond to estimates that don't achieve coarse calibration accuracy. For the exemplary concentrator AO44, all image-wise estimates remain within the coarse calibration region.

The average estimate deviates by $b^{\text{ang}}_{\text{ART},j=\text{AO44}} = 3.7\,\text{mrad}$ from the reference value, with a bias of $b^{az}_{\text{ART},j=\text{AO44}} = -1.3\,\text{mrad}$ in azimuth and $b^{el}_{\text{ART},j=\text{AO44}} = 3.6\,\text{mrad}$ in elevation. The uncertainties are $u^{az}_{\text{ART},j=\text{AO44}} = 1.3\,\text{mrad}$ and $u^{el}_{\text{ART},j=\text{AO44}} = 1.0\,\text{mrad}$, respectively. The average estimate lies inside the coarse calibration region. Its associated uncertainties, expressed as the confidence bounds $\pm 3 \cdot u^{az}_{\text{ART},j=\text{AO44}}$ and $\pm 3 \cdot u^{el}_{\text{ART},j=\text{AO44}}$, also remain within these boundaries. A systematic bias of this order of magnitude is observed not only for concentrator AO44 but consistently across all concentrators in its group (i.e., group 1), as shown in the following subsection.

7.2. Group-wise analysis

As described in Section 6.1.1, the measured concentrators were divided into six validation groups, such that all concentrators in one group have the same reference orientation. This way, the estimated orientations can be compared for concentrators observed under similar viewing conditions, and statistics can be derived from the ensemble of all orientation estimates for a given group g. The mean azimuth and elevation biases for group g are obtained as

$$\bar{b}^{az}_{\text{ART},g} = \frac{1}{|\mathcal{G}(g)|} \sum_{j \in \mathcal{G}(g)} b^{az}_{\text{ART},j}\,, \qquad \bar{b}^{el}_{\text{ART},g} = \frac{1}{|\mathcal{G}(g)|} \sum_{j \in \mathcal{G}(g)} b^{el}_{\text{ART},j}\,. \tag{39}$$

Analogously to Eq. (38), the angular mean bias is obtained as

$$\bar{b}^{\text{ang}}_{\text{ART},g} = \sqrt{\left(\bar{b}^{az}_{\text{ART},g} \cdot \cos(el^{\text{ref}}_g)\right)^2 + \left(\bar{b}^{el}_{\text{ART},g}\right)^2}\,. \tag{40}$$

The overall dispersion in azimuth and elevation are obtained as the sample standard deviations

$$s^{az}_{\text{tot},g} = \sqrt{\frac{1}{|\mathcal{G}(g)|-1} \sum_{j \in \mathcal{G}(g)} \left(b^{az}_{\text{ART},j} - \bar{b}^{az}_{\text{ART},g}\right)^2}\,, \qquad s^{el}_{\text{tot},g} = \sqrt{\frac{1}{|\mathcal{G}(g)|-1} \sum_{j \in \mathcal{G}(g)} \left(b^{el}_{\text{ART},j} - \bar{b}^{el}_{\text{ART},g}\right)^2}\,. \tag{41}$$

These componentwise terms can be summarized into the angular dispersion $s^{\text{ang}}_{\text{tot},g}$, analogously to Eq. (38):

$$s^{\text{ang}}_{\text{tot},g} = \sqrt{\left(s^{az}_{\text{tot},g} \cdot \cos(el^{\text{ref}}_g)\right)^2 + \left(s^{el}_{\text{tot},g}\right)^2}\,. \tag{42}$$

It is important to note that multiple factors contribute to these group-wise dispersions:

1. Intrinsic uncertainties of the presented method arise due to noise in the perception module, the true concentrator dimensions deviating from the assumed concentrator geometry, noise in the camera pose estimate, etc. These uncertainties can be estimated by averaging over the uncertainty estimates for all concentrators in group g (see [27]) as defined in Eq. (37), i.e.,

$$\bar{u}^{az}_{\text{ART},g} = \sqrt{\frac{1}{|\mathcal{G}(g)|} \sum_{j \in \mathcal{G}(g)} \left(u^{az}_{\text{ART},j}\right)^2}\,, \qquad \bar{u}^{el}_{\text{ART},g} = \sqrt{\frac{1}{|\mathcal{G}(g)|} \sum_{j \in \mathcal{G}(g)} \left(u^{el}_{\text{ART},j}\right)^2}\,, \qquad \bar{u}^{\text{ang}}_{\text{ART},g} = \sqrt{\left(\bar{u}^{az}_{\text{ART},g} \cdot \cos(el^{\text{ref}}_g)\right)^2 + \left(\bar{u}^{el}_{\text{ART},g}\right)^2}\,. \tag{43}$$

2. In addition, the reference orientations are subject to noise, e.g., due to shading and blocking effects in the calibration points collected using the camera-target method. Furthermore, the tracking accuracy of a heliostat is reported to depend on the proximity of its set direction to the closest calibration point used for its calibration [28]. The angular reference error can be described by a Rayleigh distribution [29] with a scale parameter of roughly $\sigma^{\text{rayleigh}}_{\text{ref}} = 1\,\text{mrad}$. The resulting contribution of tracking errors to the group-wise dispersion is given by the square root of the distributions second moment, which amounts to

$$u^{\text{ang}}_{\text{ref}} = \sqrt{2}\sigma^{\text{rayleigh}}_{\text{ref}} \approx 1.4\,\text{mrad}\,. \tag{44}$$

3. The remaining uncertainty in $s^{\text{ang}}_{\text{tot},g}$ that can't be explained by uncertainties of the method or the reference are summarized in a residual uncertainty $u^{\text{ang}}_{\text{res},g}$. This term partially stems from varying perspective and lighting conditions across different concentrators in a group, leading to different non-zero means of the error distribution of each concentrator. This causes an additional spread of the orientation estimation error in each group. Furthermore, Eq. (37) assumes homoscedasticity across all observations of a given concentrator. This assumption tends to underestimate the true uncertainty $u^{az}_{\text{ART},j}$ and $u^{el}_{\text{ART},j}$ if the standard deviations vary strongly across different observations of a concentrator. Consequently, also $\bar{u}^{\text{ang}}_{\text{ART},g}$ tends to be underestimated. Additionally, the aforementioned assumption of $u^{\text{ang}}_{\text{ref}}$ being constant across different groups may not hold, as indicated by experiments in [25], where the heliostat tracking error appears to depend on its orientation or position in the heliostat field. Hence, errors in the reference may be falsely attributed to $u^{\text{ang}}_{\text{res},g}$.

All uncertainties are summarized in the following model:

$$\left(s^{\text{ang}}_{\text{tot},g}\right)^2 = \left(\bar{u}^{\text{ang}}_{\text{ART},g}\right)^2 + \left(u^{\text{ang}}_{\text{ref}}\right)^2 + \left(u^{\text{ang}}_{\text{res},g}\right)^2\,. \tag{45}$$

Using this model, the residual uncertainty $u^{\text{ang}}_{\text{res},g}$ can be determined by rearranging Eq. (45):

$$\left(u^{\text{ang}}_{\text{res},g}\right)^2 = \max\left\{0, \left(s^{\text{ang}}_{\text{tot},g}\right)^2 - \left(\bar{u}^{\text{ang}}_{\text{ART},g}\right)^2 - \left(u^{\text{ang}}_{\text{ref}}\right)^2\right\}\,. \tag{46}$$

The overall angular deviation is quantified by the root mean squared error (RMSE), which can be expressed as a function of the angular mean bias and the angular dispersion:

$$\text{RMSE}^{\text{ang}}_g = \sqrt{\frac{1}{|\mathcal{G}(g)|} \sum_{j \in \mathcal{G}(g)} \left(b^{\text{ang}}_{\text{ART},j}\right)^2} \approx \sqrt{\left(\bar{b}^{\text{ang}}_{\text{ART},g}\right)^2 + \frac{|\mathcal{G}(g)|-1}{|\mathcal{G}(g)|} \cdot \left(s^{\text{ang}}_{\text{tot},g}\right)^2}\,. \tag{47}$$

Fig. 21 shows the biases $(b^{az}_{\text{ART},j}, b^{el}_{\text{ART},j})$ for all $j \in \mathcal{G}(g=1)$ as well as the mean bias $(\bar{b}^{az}_{\text{ART},g=1}, \bar{b}^{el}_{\text{ART},g=1})$ and the uncertainty bars for both $\bar{u}^{az}_{\text{ART},g=1}$, $\bar{u}^{el}_{\text{ART},g=1}$ and $s^{az}_{\text{tot},g=1}$, $s^{el}_{\text{tot},g=1}$.

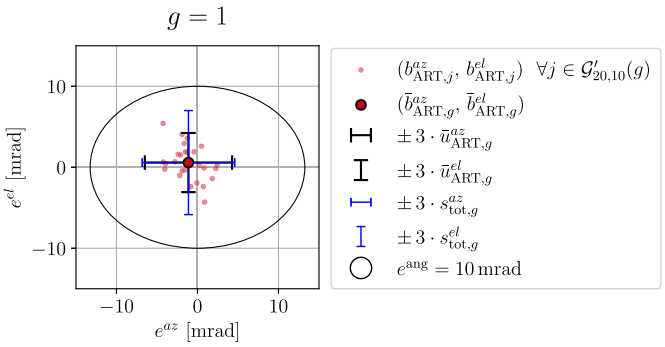


Fig. 21. Scatter plot of the orientation estimation errors for group $g = 1$ in the azimuth-elevation-plane for $R = 20$ and $D = 10$: The small circles show the errors for individual concentrators, while the large circle shows the mean error for group 1. The bars indicate the averaged group-wise sample standard deviations in both azimuth and elevation. The circle indicates the region in which the total orientation estimation error remains below 10 mrad.

The angular mean bias is $\bar{b}^{\mathrm{ang}}_{\mathrm{ART},g=1} = 1.0\,\mathrm{mrad}$. The overall angular dispersion is $s^{\mathrm{ang}}_{\mathrm{tot},g=1} = 2.6\,\mathrm{mrad}$, with a method uncertainty of $\bar{u}^{\mathrm{ang}}_{\mathrm{ART},g=1} = 1.8\,\mathrm{mrad}$. The residual uncertainty is determined to be $u^{\mathrm{ang}}_{\mathrm{res},g=1} = 1.2\,\mathrm{mrad}$. The overall error is $\mathrm{RMSE}^{\mathrm{ang}}_{g=1} = 2.7\,\mathrm{mrad}$.

Both $\bar{b}^{\mathrm{ang}}_{\mathrm{ART},g}$ and $u^{\mathrm{ang}}_{\mathrm{res},g}$ can also be observed in other groups (see Section 7.3) and likely arise from the way in which the perception noise model is fitted in Section 6.2: The noise model parameters are fitted using data selected from all groups, effectively averaging the statistics of different viewing conditions. In real conditions, the detection noise characteristics vary between groups. Hence, residual systematic errors remain in the real detections, resulting in a bias $\bar{b}^{\mathrm{ang}}_{\mathrm{ART},g}$ of the orientation errors. Variations in viewing perspectives and lighting within a group further contribute to a spread between concentrators in the same group.

This hypothesis is confirmed by experiments, conducted under controlled conditions in the simulation environment presented in [25]. The validation flight is first reproduced in simulation:

1. The simulated concentrators are aligned to their reference orientations, and for each image i, the simulated camera is placed according to the corresponding pose estimate $\hat{\boldsymbol{T}}^{\mathrm{GCS}}_{\mathrm{OCS}_i}$ obtained from Problem 1.
2. Corner detections are then generated by reprojecting the simulated concentrator corners onto the ICS for each $\hat{\boldsymbol{T}}^{\mathrm{GCS}}_{\mathrm{OCS}_i}$.
3. Every reprojected corner point is then perturbed by normally distributed noise with zero mean and covariance $\boldsymbol{\Sigma}_c$, as determined in Section 6.2.
4. In the final step, the method is run on the set of simulated images and the visibility graph $\boldsymbol{V}'_{\mathrm{Sim}}$ is obtained by considering only those simulated observations selected during IF, ROF and DBF on the real validation dataset.

Using $\boldsymbol{V}'_{\mathrm{Sim}}$, the metrics introduced in Section 7.2 are computed. The resulting angular mean biases $\bar{b}^{az}_{\mathrm{ART},g}$ and $\bar{b}^{el}_{\mathrm{ART},g}$ are close to zero for all groups, supporting the hypothesis that the biases $\bar{b}^{az}_{\mathrm{ART},g}$ and $\bar{b}^{el}_{\mathrm{ART},g}$ in the real data stem from remaining systematic biases due to an incorrectly fitted noise model. This is further supported by the fact that the simulated overall dispersion $s^{\mathrm{ang}}_{\mathrm{tot},g}$ and the method uncertainty $\bar{u}^{\mathrm{ang}}_{\mathrm{ART},g}$ lie closely together for all groups: In simulation, the corner point detection noise is zero-mean and hence, variations in perspective between different concentrators don't propagate into additional spread. Furthermore, the reference orientation in simulation is precisely known, i.e., $u^{\mathrm{ang}}_{\mathrm{ref}} = 0\,\mathrm{mrad}$.

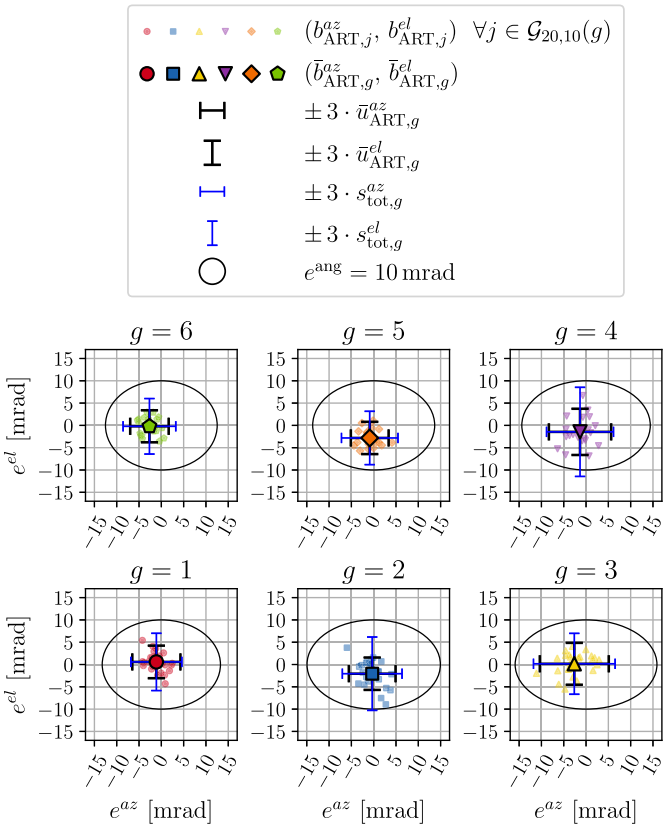


Fig. 22. Scatter plot of the orientation estimation errors for all groups $g = 1, \ldots, 6$ in the azimuth-elevation-plane for $R = 20$ and $D = 10$: The small circles show the errors for individual concentrators, while the large circles show the mean error for each group. The bars indicate the averaged group-wise sample standard deviation in both azimuth and elevation. The circle indicates the region in which the total orientation estimation error remains below 10 mrad.

7.3. Validation of all groups

This subsection presents the results for all groups $g = 1, \ldots, 6$. Fig. 22 shows all biases $(b^{az}_{\mathrm{ART},j}, b^{el}_{\mathrm{ART},j})$ as well as the mean bias $(\bar{b}^{az}_{\mathrm{ART},g},\ \bar{b}^{el}_{\mathrm{ART},g})$ and the uncertainty bars for both $\bar{u}^{az}_{\mathrm{ART},g}$, $\bar{u}^{el}_{\mathrm{ART},g}$ and $s^{az}_{\mathrm{tot},g}$, $s^{el}_{\mathrm{tot},g}$ for all groups $g = 1, \ldots, 6$. For every concentrator in every group, the method's estimates stay within the 10 mrad accuracy required for coarse calibration. A tabular overview of the component-wise metrics can be found in Appendix B.

In the remaining work, the results are discussed using the angular metrics as they summarize azimuth and elevation metrics. To obtain a performance measure across all groups, the group-wise metrics are averaged quadratically. E.g., the average angular bias is obtained as

$$\bar{b}^{\mathrm{ang}}_{\mathrm{ART,avg}} = \sqrt{\frac{1}{6}\sum_{g=1}^{6}\left(\bar{b}^{\mathrm{ang}}_{\mathrm{ART},g}\right)^2}\,. \tag{48}$$

The remaining metrics $s^{\mathrm{ang}}_{\mathrm{tot,avg}}$, $\bar{u}^{\mathrm{ang}}_{\mathrm{ART,avg}}$, $u^{\mathrm{ang}}_{\mathrm{res,avg}}$ and $\mathrm{RMSE}^{\mathrm{ang}}_{\mathrm{avg}}$ are computed analogously to Eq. (48). Note that each group is weighted equally despite the fact that the number of concentrators varies across different groups. This is because the varying number of group members stems from the flight route used to collect the validation data but does not reflect the importance of a given group. The resulting metrics are summarized in Table 1.

The smallest RMSE is obtained for group 1, where $\mathrm{RMSE}^{\mathrm{ang}}_{g=1} = 2.7\,\mathrm{mrad}$. The largest overall deviation can be observed for group 4, where $\mathrm{RMSE}^{\mathrm{ang}}_{g=4} = 4.1\,\mathrm{mrad}$. The largest method uncertainties

Table 1
Summary of the angular validation metrics for $R = 20$ and $D = 10$.

g	$\lvert\mathcal{G}(g)\rvert$	$\bar{b}^{\text{ang}}_{\text{ART},g}$ [mrad]	$s^{\text{ang}}_{\text{tot},g}$ [mrad]	$\bar{u}^{\text{ang}}_{\text{ART},g}$ [mrad]	$u^{\text{ang}}_{\text{res},g}$ [mrad]	$\text{RMSE}^{\text{ang}}_{g}$ [mrad]
1	27	1.0	2.6	1.8	1.2	2.7
2	29	2.1	3.1	1.7	2.2	3.7
3	30	1.7	3.0	2.3	1.3	3.4
4	21	1.7	3.8	2.4	2.6	4.1
5	19	2.9	2.5	1.6	1.4	3.8
6	19	2.1	2.6	1.7	1.4	3.3
avg		2.0	3.0	1.9	1.8	3.5

are obtained for group 3 and group 4, with $\bar{u}^{\text{ang}}_{\text{ART},g=3} = 2.3\,\text{mrad}$ and $\bar{u}^{\text{ang}}_{\text{ART},g=4} = 2.4\,\text{mrad}$, respectively. The quadratically averaged RMSE amounts to $\text{RMSE}^{\text{ang}}_{\text{avg}} = 3.5\,\text{mrad}$.

The increased uncertainty in group 3 and group 4 can be explained by the presence of low–ψ observations in these groups. As demonstrated in Section 5.1, observations with a lower ψ have a higher measurement uncertainty. Consequently, the average of such observations will also show a higher uncertainty. Section 7.4 contains a more detailed analysis of this effect and demonstrates that DBF reduces the effect of such low–ψ observations on the final estimate. However, the number of available observations with high ψ values is still intrinsically limited by the HFOV of the measurement camera, i.e., group 3 and group 4 still tend to contain lower ψ observations even after DBF, resulting in larger uncertainties $\bar{u}^{\text{ang}}_{\text{ART,avg}}$.

Nonetheless, the validation results show that it is possible to obtain the concentrator orientation in coarse calibration accuracy with $R = 20$ and $D = 10$, even for concentrators in group 3 and 4, whose normal vectors are closely aligned with the optical axis of the measurement camera. We hence conclude that a lawnmower pattern flight trajectory with a fixed camera orientation as it was used to acquire the validation data (see Section 6.1) is sufficient to achieve coarse calibration accuracy under any concentrator orientation contained in the validation dataset, given that the concentrator corner points can be detected in the images. The following subsection concludes the validation by describing the method's performance as a function of R and D.

7.4. Selection of R and D

This subsection investigates the influence of R and D on the validation results using the metrics introduced in the previous subsections. By conducting a parameter sweep, the method's performance is assessed as a function of available observations. The number of representative observations is varied as

$$R \in \{5, 10, 15, \dots, 50\}\,, \tag{49}$$

which corresponds to observation densities achievable at drone velocities ranging from approximately 3 to 30 m/s, assuming the configuration described in Section 6.1. Since $D \leq R$ by design, D is varied as

$$D \in \{5, 10, \dots, R\} \tag{50}$$

for each value of R. For every resulting (R, D) pair, the original visibility graph is processed to generate a corresponding filtered dataset, as demonstrated for $(R, D) = (20, 10)$ in Fig. 19. Note that not all concentrators meet the minimum observation requirement for all parameter combinations. In such cases, concentrators that are observed fewer than R times are excluded from that specific validation run, but may still contribute to others where the observation count is sufficient.

For every obtained filtered dataset, the metrics are computed as described in Sections 7.1 and 7.3. $\text{RMSE}^{\text{ang}}_{\text{avg}}$ summarizes the overall error across all validation groups and is hence a good indicator of the method's performance for a given (R, D). Fig. 23 shows $\text{RMSE}^{\text{ang}}_{\text{avg}}(R, D)$ for each

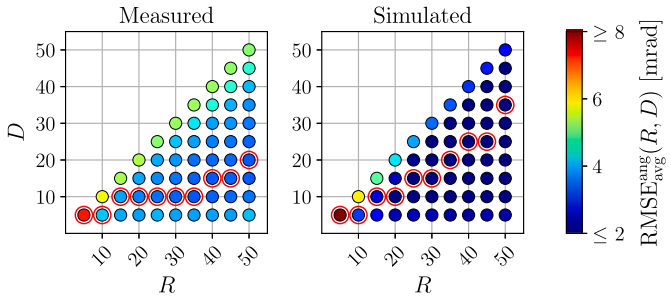


Fig. 23. $\text{RMSE}^{\text{ang}}_{\text{avg}}$ as a function of R and D for both measured data (left) and simulated data (right). The value D which minimizes $\text{RMSE}^{\text{ang}}_{\text{avg}}(R, D)$ for a given R is indicated by a red circle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

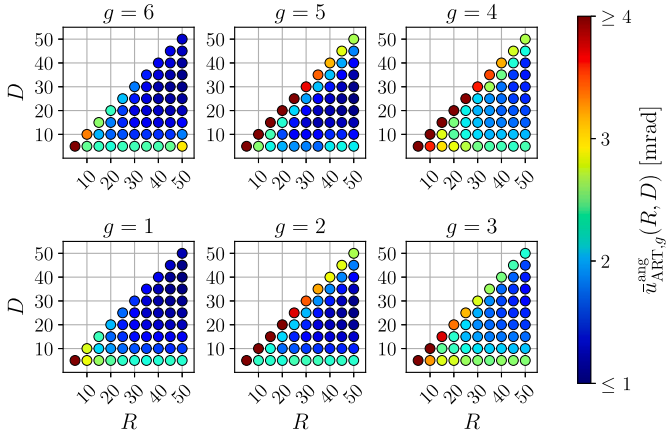


Fig. 24. Measured method uncertainty $\bar{u}^{\text{ang}}_{\text{ART},g}$ as a function of R and D, plotted separately for each group $g = 1, \dots, 6$.

of the 55 unique parameter combinations (R, D) for both measured and simulated validation data.

The red circles in Fig. 23 mark the optimal values D which minimize $\text{RMSE}^{\text{ang}}_{\text{avg}}(R, D)$ for a given R, i.e.,

$$D_{\text{opt}}(R) = \arg\min_D \left(\text{RMSE}^{\text{ang}}_{\text{avg}}(R, D) \right)\,. \tag{51}$$

For both measured and simulated data, $D_{\text{opt}}(R)$ tends to increase with larger R. While the worst performance is obtained along the top diagonal line where $D = R$, the method's performance also decreases for low D. Hence, $D_{\text{opt}}(R)$ lies in a "corridor" between $D = 5$ and $D = R$.

As already discussed in Section 7.2, the degraded performance for $D = R$ can be explained by an increased measurement uncertainty $\bar{u}^{\text{ang}}_{\text{ART,avg}}$ due to the presence of low–ψ observations: If $D = R$, all observations obtained after ROF are averaged for the final estimate, including those with low ψ values. Consequently, the average also exhibits a higher measurement uncertainty. This effect is particularly pronounced in groups 2, 3, 4 and 5, as demonstrated by Fig. 24, where $\bar{u}^{\text{ang}}_{\text{ART},g}$ is plotted over all parameter combinations (R, D) for the measured validation data.

Excluding low–ψ observations using DBF with $D < R$ results in a lower uncertainty, even though the number of averaged observations is reduced. However, an increased $\bar{u}^{\text{ang}}_{\text{ART},g}$ for $D = 5$ in all groups indicates that D should not be selected too low: In this case, only a few observations contribute to the average estimate for each concentrator, resulting in a larger uncertainty.

While the trajectories of $D_{\text{opt}}(R)$ show similar behaviors for both measured and simulated $\text{RMSE}^{\text{ang}}_{\text{avg}}$, $D_{\text{opt}}(R)$ increases at a lower rate for

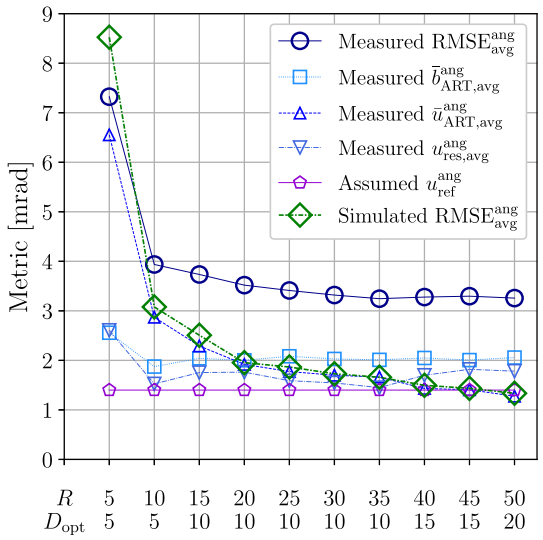


Fig. 25. Measured RMSE$^{ang}_{avg}$, $\bar{b}^{ang}_{ART,avg}$, $\bar{u}^{ang}_{ART,avg}$, $u^{ang}_{res,avg}$, u^{ang}_{ref}, as well as simulated RMSE$^{ang}_{avg}$ following the path of red circles in Fig. 23, i.e., choosing $D_{opt}(R)$ according to Eq. (51) for every R. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the measured data than for the simulated data. This is due to the fact that RMSE$^{ang}_{g}$ is composed not only of the method's uncertainty $\bar{u}^{ang}_{ART,g}$ but also of systematic biases $\bar{b}^{ang}_{ART,g}$ as well as the residual uncertainty $u^{ang}_{res,g}$. As stated in Section 7.2, these terms likely stem from a misfit of the perception noise model and hence can partially be attributed to the method's input and not the method itself. To demonstrate this effect, Fig. 25 shows the measured and simulated RMSE$^{ang}_{avg}$ over the optimal trajectory $(R, D_{opt}(R))$. In addition, it shows the measured terms $\bar{b}^{ang}_{ART,avg}$, $\bar{u}^{ang}_{ART,avg}$ and $s^{ang}_{tot,avg}$, into which the measured RMSE$^{ang}_{avg}$ can be decomposed.

While the largest contribution to RMSE$^{ang}_{avg}$ stems from the method's uncertainty $\bar{u}^{ang}_{ART,g}$ for $R \leq 15$, this uncertainty becomes low for $R \geq 20$ while the other noise terms remain roughly constant. The method's performance is limited by $\bar{b}^{ang}_{ART,avg}$ and $u^{ang}_{res,avg}$ for $R \geq 20$, which are suspected to stem from remaining systematic errors in the perception module.

This hypothesis is confirmed by the fact that the simulated RMSE$^{ang}_{avg}$ further decreases for large R, closely following the measured $\bar{u}^{ang}_{ART,avg}$. Since the simulation assumes zero-mean concentrator corner point detection, we conclude that $\bar{u}^{ang}_{ART,avg}$ serves as a good estimate of the achievable method performance in the case that the perception module outputs zero-mean detections, with the noise levels as described in Section 6.2. According to the simulation, RMSE$^{ang}_{avg}$ could be reduced to 1.5 mrad with $R \geq 40$ for the presented measurement setup.

On the other hand, the measured RMSE$^{ang}_{avg}$ only decreases slightly with $R \geq 20$, justifying the choice of $R = 20$ (and the corresponding $D_{opt}(20) = 10$) in the previous subsections. The perception module's performance is concluded to be the main limiting factor of the method's performance and improvements can be expected if both systematic and statistical errors in the perception module are reduced. However, other factors may limit the method's performance on real data, as the simulation assumes perfect knowledge of quantities that are subject to noise in real heliostat fields: For example, the real concentrator geometry may deviate from the assumed geometry due to facet canting or mounting errors. Furthermore, the assumed heliostat positions are only known up to cm-level accuracy (see [25]). Nonetheless, coarse calibration accuracy is demonstrated for all validated concentrators.

7.5. Flight time estimation and practical constraints

Assuming a rectangular heliostat field of length l_{field} (along the flight direction) and width w_{field} (perpendicular to the flight direction), the field is scanned using a lawnmower pattern consisting of N_{FL} parallel flight lines, as illustrated in Fig. 26. Each flight line spans approximately the length l_{field}. Neglecting the short turning segments between adjacent flight lines, the total flight path length is given by

$$l_{FP} = N_{FL} \cdot l_{field} \,. \tag{52}$$

The number of flight lines depends on the distance d_{FL} between adjacent flight lines:

$$N_{FL} = \frac{w_{field}}{d_{FL}} \,. \tag{53}$$

Assuming nadir configuration (i.e., camera facing toward the ground) and that the shorter sensor side is oriented perpendicular to the flight direction, the camera observes a ground width

$$w_{cam} = 2 \cdot h_{cam} \cdot \tan\left(\frac{VFOV_{cam}}{2}\right) \,, \tag{54}$$

where $VFOV_{cam}$ denotes the vertical field of view (VFOV) and h_{cam} denotes the flight height. To ensure that each concentrator is observed from N_{passes} adjacent flight lines, the distance between flight lines is chosen as

$$d_{FL} = \frac{w_{cam}}{N_{passes}} \,. \tag{55}$$

Substituting the above equations yields the total flight path length

$$l_{FP} = \frac{l_{field} \cdot w_{field} \cdot N_{passes}}{2 \cdot h_{cam} \cdot \tan\left(VFOV_{cam}/2\right)} \,. \tag{56}$$

The total flight time follows as

$$t_{FP} = \frac{l_{FP}}{v_{cam}} = \frac{l_{field} \cdot w_{field} \cdot N_{passes}}{2 \cdot h_{cam} \cdot \tan\left(VFOV_{cam}/2\right) \cdot v_{cam}} \,. \tag{57}$$

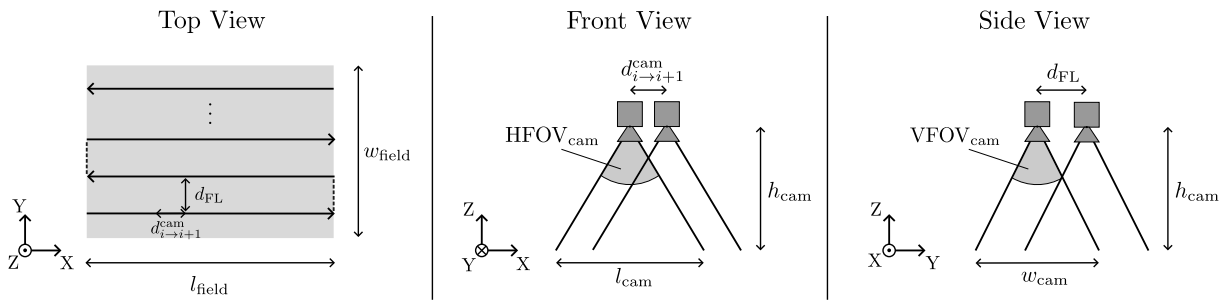


Fig. 26. Top view, front view and side view illustration of the lawnmower pattern and the implications on the length and width observed by the camera.

Expressing the time per field area $A_{\text{field}} = l_{\text{field}} \cdot w_{\text{field}}$ yields

$$\frac{t_{\text{FP}}}{A_{\text{field}}} = \frac{N_{\text{passes}}}{2 \cdot h_{\text{cam}} \cdot \tan\left(\text{VFOV}_{\text{cam}}/2\right) \cdot v_{\text{cam}}} . \tag{58}$$

Besides geometric coverage, a sufficient number of observations per concentrator is required. It is assumed that a concentrator is recorded in R_{FL} images along a single flight line. Since N_{passes} adjacent flight lines pass each concentrator, the total number of observations is

$$R = N_{\text{passes}} \cdot R_{\text{FL}} . \tag{59}$$

The observation count per flight line depends on the ground length observed by the camera

$$l_{\text{cam}} = 2 \cdot h_{\text{cam}} \cdot \tan\left(\frac{\text{HFOV}_{\text{cam}}}{2}\right) \tag{60}$$

and the distance traveled between two consecutive images

$$d_{i \to i+1}^{\text{cam}} = \frac{v_{\text{cam}}}{f_{\text{cam}}} , \tag{61}$$

where f_{cam} denotes the image acquisition rate. Approximating that a concentrator appears over a ground distance comparable to the ground length l_{cam}, the observation count per flight line becomes

$$R_{\text{FL}} = \frac{l_{\text{cam}}}{d_{i \to i+1}^{\text{cam}}} = \frac{2 \cdot h_{\text{cam}} \cdot \tan(\text{HFOV}_{\text{cam}}/2) \cdot f_{\text{cam}}}{v_{\text{cam}}} . \tag{62}$$

Using the parameters of the validation setup

$$h_{\text{cam}} = 35\,\text{m}, \quad \text{HFOV}_{\text{cam}} = 54.3°, \quad f_{\text{cam}} = 1\,\text{Hz},$$

Eq. (62) yields

$$R_{\text{FL}} \approx \frac{36\,\text{ms}^{-1}}{v_{\text{cam}}} . \tag{63}$$

The choice of v_{cam} must respect the physical limits of the UAV as well as constraints regarding the image quality, such as motion blur. In the present setup, $v_{\text{cam}} = 7.2\,\text{ms}^{-1}$ represents a reasonably fast, yet feasible choice that yields the integer-value $R_{\text{FL}} = 5$. In this case, $N_{\text{passes}} = 4$ can be chosen such that $R = N_{\text{passes}} \cdot R_{\text{FL}} = 20$, as identified in the previous subsection. More generally, v_{cam} and N_{passes} can be traded off to achieve a desired total observation count R: Flying slower increases R_{FL} (i.e., the number of images acquired per flight line), allowing a reduction in N_{passes} while maintaining the same R.

Inserting $v_{\text{cam}} = 7.2\,\text{ms}^{-1}$, $N_{\text{passes}} = 4$ and $\text{VFOV}_{\text{cam}} = 37.8°$ into Eq. (58) results in

$$\frac{t_{\text{FP}}}{A_{\text{field}}} = 23.18\,\text{ms}\,\text{m}^{-2} = 3.87\,\text{min}\,\text{ha}^{-1} = 6.44\,\text{h}\,\text{km}^{-2} . \tag{64}$$

Hence, an entire scan of the heliostat field at the STJ (where $A_{\text{field}} \approx 10\,\text{ha}$) can be completed in roughly 38.7 min. The flight execution and data evaluation are automated to a large extent. Hence, a single operator is sufficient to supervise the flight mission, where the degree of required human supervision depends on the applicable aviation regulations.

Alternatively, the scan time can be expressed relative to the total mirror aperture area A_{solar}. For the STJ, this mirror area amounts to approximately $17500\,\text{m}^2$, yielding a scan time per mirror area of

$$\frac{t_{\text{FP}}}{A_{\text{solar}}} \approx 133\,\text{ms}\,\text{m}^{-2} . \tag{65}$$

It should be noted that such a normalization depends on the ground coverage ratio (i.e., the ratio between mirror area and land area), which varies between heliostat fields. Consequently, expressing the scan time per unit land area provides a more generally transferable metric across different plant layouts and heliostat sizes. Note that the values given in Eqs. (64) and (65) should be interpreted as a baseline estimates of the required scan time. They are based on several simplifying assumptions.

First, turning maneuvers between adjacent flight lines were neglected. In practice, the UAV requires additional time to decelerate and accelerate again when transitioning from one flight line to the next. The additional time depends on the ratio between the turning time per line and the time required to traverse an entire flight line. Consequently, the actual scan time may exceed the baseline value depending on the specific field geometry and flight dynamics.

Second, the selected flight height in the validation setup was chosen to suit the heliostat dimensions, the field layout and the camera IOR. For larger heliostats, it may be necessary to fly at a greater altitude in order to record at least three heliostats within a single image, as required for the camera pose estimation. For a fixed camera field of view, increasing the flight height enlarges both the ground length and the width observed by a single image. Compared to a heliostat field with the same total land area but smaller sized heliostats, the distance between adjacent flight lines can hence be chosen larger due to the larger ground width, reducing the number of required flight lines to cover the total area and thereby reducing the scan time. In addition, the ground length increases the number of observations per flight line and may allow higher flight velocities while maintaining a desired observation count, provided that UAV speed limits and motion-blur constraints are respected. However, increasing the flight height reduces the ground sampling distance, which may necessitate a higher observation count per concentrator to preserve measurement accuracy. Moreover, heliostat spacing is typically not homogeneous but varies with the heliostat–tower distance to avoid shading. An optimal flight route may therefore require non-uniform distances between flight lines adapted to the local field geometry. The systematic optimization of flight height, velocity and flight line spacing as a function of heliostat size and layout lies beyond the scope of this work.

Third, the derivation assumes nadir viewing for mathematical simplicity. In practice we observed that nadir images often result in ground reflections being visible on the concentrator surfaces, which can complicate edge detection depending on the concentrator orientation. For this reason, the camera is pitched away from nadir in the validation experiments. Note that off-nadir viewing increases the ground area covered by each image and may allow both larger spacing between flight lines and higher flight velocities, thereby reducing measurement time. As for larger flight heights, the resulting decreased ground sampling distance may necessitate a higher observation count per concentrator to preserve measurement accuracy.

The presented estimates should therefore be regarded as a site-specific order-of-magnitude approximation. The system's transfer to other heliostat fields requires refinement using plant-specific geometric, operational, and regulatory parameters, ideally within a dedicated technoeconomic assessment.

8. Conclusion

In this work, we have presented a novel approach for the airborne estimation of solar concentrator orientation. As an input, it requires the image coordinates of identified concentrator corner points of each concentrator for which the orientation is to be estimated. In addition, it requires the heliostat positions, a model of the design kinematic system as well as the design concentrator geometry, i.e., quantities that are typically well-known. Since the method processes images individually and does not rely on image sequences, it is suited for the application in real-time, laying the foundation for autonomous solar tower power plant condition monitoring systems. While the presented method also estimates the camera pose, a detailed analysis of the camera pose accuracy lies beyond the scope of this work. However, first experiments show that this accuracy lies in the range of RTK accuracy. Hence, the

proposed method could potentially replace hardware systems for camera localization. This is especially relevant in remote areas where RTK correction signals may not be available, or in cases where the UAV is not equipped with an RTK system.

The method's development process was supported by a dedicated simulation environment [25], allowing the analysis of the effect of noise in different components of the algorithm on the final result. During development, simulation-based findings were transferred to real data successfully, showing that a simulation environment can help in the development process of airborne condition monitoring systems for solar tower power plants.

We validated the algorithm using measurement data collected at the solar tower test facility in Jülich, Germany and demonstrated the accuracy on 146 validated concentrators to be $\mathrm{RMSE}_{\mathrm{avg}}^{\mathrm{ang}} = 3.5\,\mathrm{mrad}$, given that a number of 20 observations per concentrator is used. While simulations predicted that $\mathrm{RMSE}_{\mathrm{avg}}^{\mathrm{ang}}$ could be reduced further by increasing the number of observations, the error was only slightly reduced when increasing the number of observations for the real measurement data. Based on simulation experiments and the decomposition of $\mathrm{RMSE}_{\mathrm{avg}}^{\mathrm{ang}}$ in different uncertainty terms, we suspect systematic biases in the perception module to limit the method's performance for a large number of observations. Nevertheless, the achieved accuracy qualifies the presented method as a coarse calibration system, showing that it can be used as an upstream component to fine calibration systems for heliostats.

While the method was validated by means of a static heliostat field, this static assumption is not strictly required. Field operators typically maintain a model of the expected concentrator poses over time, based on solar position data and kinematic parameters. This allows for the comparison of estimated concentrator orientations against their time-varying reference values. Hence, tracking errors can still be evaluated by analyzing these deviations over time. Future work could address the additional complexity introduced by such a dynamic reference model.

CRediT authorship contribution statement

Alexander Schnerring: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Rafal Broda:** Writing – review & editing, Software, Methodology, Conceptualization. **Michael Nieslony:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Niels Algner:** Writing – review & editing, Resources, Methodology. **Eduardo Saez Martinez:** Writing – review & editing, Software, Methodology. **Marc Röger:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Sonja Kallio:** Writing – review & editing, Supervision. **Robert Pitz-Paal:** Supervision, Funding acquisition.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT (GPT 5, OpenAI) to improve the readability and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

Financial support from the Ministry of Economic Affairs, Industry, Climate Action and Energy of the State of North Rhine-Westphalia (5hine, contract 005-2108-0068) is gratefully acknowledged. We also thank CSP Services for their support during the measurement campaign. Furthermore, we thank our German Aerospace Center (DLR) colleagues Oliver Kaufhold, Moritz Wirger, Marcel Sibum, Max Pargmann, Felix Göhring and Peter Schwarzbözl at the STJ.

Appendix A. Nomenclature

See Tables A.1–A.4.

Table A.1
Overview of the coordinate systems and estimated quantities (see Section 4.2).

Symbol	Meaning
GCS	Global coordinate system (fixed)
OCS_i	Observer coordinate system for image i
CCS_j	Concentrator coordinate system for concentrator j
$\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$	Estimated camera pose for image i
$\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$	Estimated pose of concentrator j, based on image i, expressed w.r.t. OCS_i
$\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)$	Estimated pose of concentrator j, based on image i, transformed to the GCS using $\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$
$\hat{\boldsymbol{n}}_j^{\mathrm{GCS}}$	Estimated normal vector of concentrator j, expressed w.r.t. GCS

Table A.2
Summary of the involved quantities for concentrator j in Problem 1 (see Section 4.2).

Symbol	Quantity	ICS_i	GCS	CCS_j
$\boldsymbol{C}_j$	Concentrator corner points	detected in		known in
$\boldsymbol{o}_j$	Heliostat origin		known in	
$\boldsymbol{p}_j$	Concentrator passpoint	detected in	estimated in	known in
$\boldsymbol{n}_j$	Concentrator normal vector		estimated in	
$\boldsymbol{K}_j$	Kinematic model		known in	

Table A.3
Overview of the variables used to process the raw validation data (see Section 6).

Symbol	Meaning
g	Validation group index
τ_{px}	Pixel threshold for reprojected concentrator corner points during IF step
R	Number of selected representative observations during ROF step
D	Number of observations selected based on descriptor ψ during DBF step
$\mathcal{G}_{R,D}(g)$	Set of all concentrators in each validation group g after applying IF, ROF and DBF
$\mathcal{I}'_{R,D}(j)$	Set of all images observing concentrator j after applying IF, ROF and DBF

Table A.4
Overview of the validation metrics computed in Section 7.

Symbol	Meaning	Validation level
$\hat{az}_j(i)$, $\hat{el}_j(i)$	Azimuth/elevation angle estimate for concentrator j, computed from a single observation i	Observation
az_j^{ref}, el_j^{ref}	Azimuth/elevation angle reference value for concentrator j	Concentrator
$az_{\mathrm{ART},j}$, $el_{\mathrm{ART},j}$	ARTSCORE-C azimuth/elevation angle estimate for concentrator j	
$b^{az}_{\mathrm{ART},j}$, $b^{el}_{\mathrm{ART},j}$, $b^{\mathrm{ang}}_{\mathrm{ART},j}$	Azimuth/elevation/angular bias of ARTSCORE-C estimate for concentrator j	
$u^{az}_{\mathrm{ART},j}$, $u^{el}_{\mathrm{ART},j}$, $u^{\mathrm{ang}}_{\mathrm{ART},j}$	Azimuth/elevation/angular uncertainty of ARTSCORE-C estimate for concentrator j	
$\bar{b}^{az}_{\mathrm{ART},g}$, $\bar{b}^{el}_{\mathrm{ART},g}$, $\bar{b}^{\mathrm{ang}}_{\mathrm{ART},g}$	Azimuth/elevation/angular mean bias of ARTSCORE-C estimate for group g, averaged over systematic errors $b^{az}_{\mathrm{ART},j}$, $b^{el}_{\mathrm{ART},j}$, $b^{\mathrm{ang}}_{\mathrm{ART},j}$ for all $j \in \mathcal{G}(g)$	Group
$\bar{u}^{az}_{\mathrm{ART},g}$, $\bar{u}^{el}_{\mathrm{ART},g}$, $\bar{u}^{\mathrm{ang}}_{\mathrm{ART},g}$	Azimuth/elevation/angular uncertainty of ARTSCORE-C estimate for group g, averaged over uncertainties $u^{az}_{\mathrm{ART},j}$, $u^{el}_{\mathrm{ART},j}$, $u^{\mathrm{ang}}_{\mathrm{ART},j}$ for all $j \in \mathcal{G}(g)$	
$s^{az}_{\mathrm{tot},g}$, $s^{el}_{\mathrm{tot},g}$, $s^{\mathrm{ang}}_{\mathrm{tot},g}$	Azimuth/elevation/angular overall dispersion of group g, estimated from all ARTSCORE-C estimation errors for all $j \in \mathcal{G}(g)$	
$u^{\mathrm{ang}}_{\mathrm{res},g}$	Residual angular uncertainty of group g	
$\mathrm{RMSE}^{\mathrm{ang}}_g$	Angular root mean squared error of group g	
$u^{\mathrm{ang}}_{\mathrm{ref}}$	Reference angular uncertainty	All Groups
$\bar{b}^{az}_{\mathrm{ART},g}\,Avg$, $\bar{b}^{el}_{\mathrm{ART,avg}}$, $\bar{b}^{\mathrm{ang}}_{\mathrm{ART,avg}}$	Azimuth/elevation/angular bias of ARTSCORE-C estimate for all groups, averaged over systematic errors $\bar{b}^{az}_{\mathrm{ART},g}$, $\bar{b}^{el}_{\mathrm{ART},g}$, $\bar{b}^{\mathrm{ang}}_{\mathrm{ART},g}$ for $g = 1, \ldots, 6$	
$\bar{u}^{az}_{\mathrm{ART,avg}}$, $\bar{u}^{el}_{\mathrm{ART,avg}}$, $\bar{u}^{\mathrm{ang}}_{\mathrm{ART,avg}}$	Azimuth/elevation/angular uncertainty of ARTSCORE-C estimate for all groups, averaged over uncertainties $\bar{u}^{az}_{\mathrm{ART},g}$, $\bar{u}^{el}_{\mathrm{ART},g}$, $\bar{u}^{\mathrm{ang}}_{\mathrm{ART},g}$ for $g = 1, \ldots, 6$	
$s^{az}_{\mathrm{tot,avg}}$, $s^{el}_{\mathrm{tot,avg}}$, $s^{\mathrm{ang}}_{\mathrm{tot,avg}}$	Azimuth/elevation/angular overall dispersion for all groups, averaged over $s^{az}_{\mathrm{tot},g}$, $s^{el}_{\mathrm{tot},g}$, $s^{\mathrm{ang}}_{\mathrm{tot},g}$ for $g = 1, \ldots, 6$	
$\mathrm{RMSE}^{\mathrm{ang}}_{\mathrm{avg}}$	Angular root mean squared error of all groups, averaged over $\mathrm{RMSE}^{\mathrm{ang}}_g$ for $g = 1, \ldots, 6$	

Appendix B. Supplementary Material

B.1. Supplementary Material for Section 6.2

Table B.1 lists the means, standard deviations and correlation coefficients of the fitted bivariate normal distributions shown in Fig. 15, characterizing the noise in the perception module. Note that the noise is fitted separately for every corner $c \in \{\mathrm{UL}, \mathrm{UR}, \mathrm{LL}, \mathrm{LR}\}$, with UL = upper left, UR = upper right, LL = lower left, LR = lower right.

Table B.1
Numeric values for the perception module error distributions shown in Fig. 15.

c	$\hat{\mu}_c$, X [px]	$\hat{\mu}_c$, Y [px]	$\hat{\sigma}_c$, X [px]	$\hat{\sigma}_c$, Y [px]	$\hat{\rho}$
UL	−0.83	0.99	1.00	0.72	−0.10
UR	−0.31	1.24	1.04	0.99	−0.31
LL	0.00	0.29	0.58	0.65	−0.07
LR	−0.37	−0.50	0.84	1.13	−0.01

B.2. Supplementary Material for Section 7

Table B.2 summarizes the validation metrics in azimuth and elevation for $R = 20$ and $D = 10$. For the average biases $\bar{b}^{az}_{\mathrm{ART,avg}}$ and $\bar{b}^{el}_{\mathrm{ART,avg}}$, the arithmetic mean is computed. For the average uncertainties, the quadratic mean is computed.

Table B.2
Summary of the validation metrics for $R = 20$ and $D = 10$.

g	$\lvert\mathcal{G}(g)\rvert$	$\bar{b}^{az}_{\mathrm{ART},g}$ [mrad]	$\bar{b}^{el}_{\mathrm{ART},g}$ [mrad]	$\bar{u}^{az}_{\mathrm{ART},g}$ [mrad]	$\bar{u}^{el}_{\mathrm{ART},g}$ [mrad]	$s^{az}_{\mathrm{tot},g}$ [mrad]	$s^{el}_{\mathrm{tot},g}$ [mrad]
1	27	−1.1	0.6	1.8	1.2	1.9	2.1
2	29	−0.3	−2.1	1.7	1.2	2.2	2.7
3	30	−2.6	0.2	2.6	1.6	3.0	2.3
4	21	−1.3	−1.5	2.3	1.7	2.5	3.3
5	19	−0.9	−2.8	1.4	1.2	2.1	2.0
6	19	−2.7	−0.2	1.4	1.2	2.0	2.1
avg		−1.5	−1.0	1.9	1.4	2.3	2.5

Appendix C. Monte-Carlo experiment

The second Monte-Carlo experiment is designed to cover the domain of interest for the validation data, collected using the measurement setup described in Section 6.1. To ensure that the sampling process covers all relevant relative poses, a sampling scheme similar to Collins and Bartoli [30] is chosen: For the translational part, $J = 5000$ random points are sampled in the ICS and the rays $\boldsymbol{w}_j$ are computed and projected from the camera focal point into the OCS. The translation component of the concentrator pose is $\mathrm{Trans}\left(\boldsymbol{T}^{\mathrm{OCS}}_{\mathrm{CCS}_j}\right) = d_j \cdot \boldsymbol{w}_j / |\boldsymbol{w}_j|$, where the distance d_j between the projected point and the camera focal point is uniformly drawn from the distribution $d_j \sim \mathcal{U}(30\,\mathrm{m}, 60\,\mathrm{m})$. This way, only those poses are sampled where the concentrator is visible in the image. For $\mathrm{Rot}\left(\boldsymbol{T}^{\mathrm{OCS}}_{\mathrm{CCS}_j}\right)$, the concentrator normal vectors are first aligned with their respective rays $\boldsymbol{w}_j$ to ensure they are facing the camera. After this alignment, each concentrator is rotated with an angle of $\psi_j \sim \mathcal{U}(0°, 60°)$ around a rotation vector $\boldsymbol{r}_j$ on the unit circle in the CCS XY-plane, i.e., $\boldsymbol{r}_j = (\cos(\theta_j)\ \sin(\theta_j)\ 0)^T$ with $\theta_j \sim \mathcal{U}(0, 2\pi)$. In the last step, each concentrator is rotated around its optical axis by an angle $\phi_j \sim \mathcal{U}(0, 2\pi)$. Fig. C.1 shows a schematic of the sampling process for one concentrator.

As in the first Monte-Carlo experiment, four independent error vectors $(e^{\mathrm{ICS}}_{\mathrm{X}}, e^{\mathrm{ICS}}_{\mathrm{Y}})^T$, are sampled $K = 1000$ times according to the bivariate normal distribution described in Eq. (24) for each relative perspective. The P4P algorithm is applied to every noisy concentrator projection, resulting in the rotation estimates $\hat{\boldsymbol{R}}^{\mathrm{OCS}}_{\mathrm{CCS}_{j,k}}$ for every concentrator $j = 1, \ldots, 5000$ and $k = 1, \ldots, 1000$.

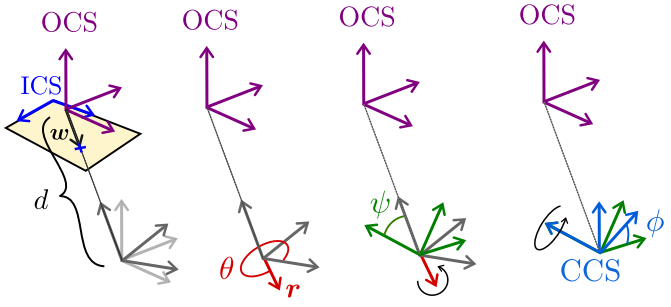


Fig. C.1. Schematic overview of the sampling process in the second Monte-Carlo experiment.

Appendix D. Algorithms

D.1. Algorithms for Section 4

Algorithm 1 Computing passpoint from corner points [16].

Input:
Detected corner points: $\boldsymbol{C}_j^{\mathrm{ICS}_i}$
Output:
Passpoint: $\boldsymbol{p}_j^{\mathrm{ICS}_i}$

$\boldsymbol{d}_1 \leftarrow [\boldsymbol{C}_j^{\mathrm{ICS}_i}]_{\mathrm{UR}} - [\boldsymbol{C}_j^{\mathrm{ICS}_i}]_{\mathrm{LL}}$
$\boldsymbol{d}_2 \leftarrow [\boldsymbol{C}_j^{\mathrm{ICS}_i}]_{\mathrm{UL}} - [\boldsymbol{C}_j^{\mathrm{ICS}_i}]_{\mathrm{LR}}$
$\Delta \leftarrow \boldsymbol{d}_1 \times \boldsymbol{d}_2$ ▷ 2D cross product (scalar)
$\boldsymbol{p}_j^{\mathrm{ICS}_i} \leftarrow [\boldsymbol{C}_j^{\mathrm{ICS}_i}]_{\mathrm{LL}} + \frac{\left([\boldsymbol{C}_j^{\mathrm{ICS}_i}]_{\mathrm{LR}} - [\boldsymbol{C}_j^{\mathrm{ICS}_i}]_{\mathrm{LL}}\right) \times \boldsymbol{d}_2}{\Delta} \cdot \boldsymbol{d}_1$
return $\boldsymbol{p}_j^{\mathrm{ICS}_i}$

Algorithm 2 Solution to Problem 1.

Input:
Detected corner points: $\{\boldsymbol{C}_j^{\mathrm{ICS}_i}\ \forall j \in \mathcal{C}(i)\}$
Detected passpoints: $\{\boldsymbol{p}_j^{\mathrm{ICS}_i}\ \forall j \in \mathcal{C}(i)\}$
A-priori Knowledge:
Heliostat origins: $\{\boldsymbol{o}_j^{\mathrm{GCS}}\ \forall j \in \mathcal{C}(i)\}$
Kinematic backward pass: $\boldsymbol{K}^{\mathrm{b}}$
Kinematic design parametrization: $\boldsymbol{k}$
Concentrator geometry: $\{\boldsymbol{C}_j^{\mathrm{CCS}_j}\ \forall j \in \mathcal{C}(i)\}$
Passpoints w.r.t. CCS_j: $\{\boldsymbol{p}_j^{\mathrm{CCS}_j}\ \forall j \in \mathcal{C}(i)\}$
Output:
Concentrator orientation estimates:
$\{\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)\ \forall j \in \mathcal{C}(i)\}$
Camera pose estimate: $\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$

Initialization:
$\boldsymbol{P}_i^{\mathrm{ICS}_i} \leftarrow \{\boldsymbol{p}_j^{\mathrm{ICS}_i}\ \forall j \in \mathcal{C}(i)\}$
$\boldsymbol{O}_i^{\mathrm{GCS}} \leftarrow \{\boldsymbol{o}_j^{\mathrm{GCS}}\ \forall j \in \mathcal{C}(i)\}$
$\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}} \leftarrow \mathrm{PN}_i\mathrm{P}(\boldsymbol{P}_i^{\mathrm{ICS}_i}, \boldsymbol{O}_i^{\mathrm{GCS}})$
for $j \in \mathcal{C}(i)$ **do**
  $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i} \leftarrow \mathrm{Rot}\left(\mathrm{P4P}^{-1}(\boldsymbol{C}_j^{\mathrm{ICS}_i}, \boldsymbol{C}_j^{\mathrm{CCS}_j})\right)$
  $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) \leftarrow \mathrm{Rot}\left(\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}\right) \cdot \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$
end for

Passpoint Update:
while $\{\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)\ \forall j \in \mathcal{C}(i)\}$ not converged **do**
  for $j \in \mathcal{C}(i)$ **do**
    $\hat{\boldsymbol{n}}_j^{\mathrm{GCS}}(i) \leftarrow \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}} \cdot (0\ 0\ 1)^T$
    $\hat{\boldsymbol{p}}_j^{\mathrm{GCS}}(i) \leftarrow \boldsymbol{K}^{\mathrm{b}}(\hat{\boldsymbol{n}}_j^{\mathrm{GCS}}(i) \mid \boldsymbol{o}_j^{\mathrm{GCS}}, \boldsymbol{k}) \cdot \boldsymbol{p}_j^{\mathrm{CCS}_j}$
  end for
  $\hat{\boldsymbol{P}}_i^{\mathrm{GCS}} \leftarrow \{\hat{\boldsymbol{p}}_j^{\mathrm{GCS}}(i)\ \forall j \in \mathcal{C}(i)\}$
  $\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}} \leftarrow \mathrm{PN}_i\mathrm{P}(\boldsymbol{P}_i^{\mathrm{ICS}_i}, \boldsymbol{P}_i^{\mathrm{GCS}})$
  for $j \in \mathcal{C}(i)$ **do**
    $\hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) \leftarrow \mathrm{Rot}\left(\hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}\right) \cdot \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{OCS}_i}$
  end for
end while
$\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) \leftarrow \begin{pmatrix} \hat{\boldsymbol{R}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i) & \hat{\boldsymbol{p}}_j^{\mathrm{GCS}}(i) \\ \boldsymbol{0} & 1 \end{pmatrix}$
return $\{\hat{\boldsymbol{T}}_{\mathrm{CCS}_j}^{\mathrm{GCS}}(i)\ \forall j \in \mathcal{C}(i)\}, \hat{\boldsymbol{T}}_{\mathrm{OCS}_i}^{\mathrm{GCS}}$

Data availability

Data will be made available on request.

References

[1] S.A. Jones, K.W. Stone, Analysis of Solar Two heliostat tracking error sources, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States), 1999, https://www.osti.gov/biblio/3312.
[2] J.C. Sattler, M. Röger, P. Schwarzbözl, R. Buck, A. Macke, C. Raeder, J. Göttsche, Review of heliostat calibration and tracking control methods, Solar Energy 207 (2020) 110–132, https://doi.org/10.1016/j.solener.2020.06.030
[3] R.A. Mitchell, G. Zhu, A non-intrusive optical (NIO) approach to characterize heliostats in utility-scale power tower plants: methodology and in-situ validation, Solar Energy 209 (2020) 431–445, https://doi.org/10.1016/j.solener.2020.09.004
[4] J. Yellowhair, P.A. Apostolopoulos, D.E. Small, D. Novick, M. Mann, Development of an aerial imaging system for heliostat canting assessments, AIP Conf. Proc. 2445 (1) (2022) 120024, https://doi.org/10.1063/5.0087057
[5] J.J. Krauth, C. Happich, N. Algner, R. Broda, A. Kämpgen, A. Schnerring, S. Ulmer, M. Röger, HelioPoint – a fast airborne calibration method for heliostat fields, J. Sol. Energy Eng. 146 (6) (2024) 061005, https://doi.org/10.1115/1.4065868
[6] W. Jessen, M. Röger, C. Prahl, R. Pitz-Paal, A two-stage method for measuring the heliostat offset, AIP Conf. Proc. 2445 (1) (2022) 070005, https://doi.org/10.1063/5.0087036
[7] K.W. Stone, Automatic heliostat track alignment method, U.S. Patent 4, 564, 275 Jan, 1986, https://patents.google.com/patent/US4564275A.
[8] S.J. Lombard, W.J. Smit, Practical challenges to calibrate a heliostat with a multi-copter, AIP Conf. Proc. 2445 (1) (2022) 050005, https://doi.org/10.1063/5.0085695
[9] A. Macario Barros, M. Michel, Y. Moline, G. Corre, F. Carrel, A comprehensive survey of visual SLAM algorithms, Robotics 11 (1) (2022), https://doi.org/10.3390/robotics11010024
[10] R. Broda, A. Schnerring, D. Schnaus, M. Nieslony, J.J. Krauth, M. Röger, S. Kallio, R. Triebel, R. Pitz-Paal, Bridging the sim2real gap: training deep neural networks for heliostat detection with purely synthetic data, Sol. Energy 300 (2025) 113728, https://doi.org/10.1016/j.solener.2025.113728
[11] G. Zhu, C. Augustine, R. Mitchell, M. Muller, P. Kurup, A. Zolan, S. Yellapantula, R. Brost, K. Armijo, J. Sment, R. Schaller, M. Gordon, M. Collins, J. Coventry, J. Pye, M. Cholette, G. Picotti, M. Arjomandi, M. Emes, D. Potter, M. Rae, Heliocon: a roadmap for advanced heliostat technologies for concentrating solar power, Sol. Energy 264 (2023) 111917, https://doi.org/10.1016/j.solener.2023.111917
[12] M. Röger, T. Schlichting, K. Blume, Guidelines for heliostat testing, in: G. Zhu, M. Röger, Z. Wang (Eds.), Advances in Solar Energy: Heliostat Systems Design, Implementation, and Operation, vol. 12671, International Society for Optics and Photonics, SPIE, 2023, https://doi.org/10.1117/12.2682675, p. 1267106.
[13] M. Röger, et al., SolarPaces guideline for heliostat performance testing, version 1.0, 2023, https://doi.org/10.5281/zenodo.15365475
[14] E. Marchand, H. Uchiyama, F. Spindler, Pose estimation for augmented reality: a hands-on survey, IEEE Trans. Vis. Comput. Graph. 22 (12) (2016) 2633–2651, https://doi.org/10.1109/TVCG.2015.2513408
[15] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, Elsevier, New York, 1976, https://www.zib.de/userpage/groetschel/teaching/WS1314/BondyMurtyGTWA.pdf.
[16] F. Antonio, Iv.6 - faster line segment intersection, In D. Kirk (Ed.), Graphics Gems III (IBM Version), Morgan Kaufmann, San Francisco, 1992, pp. 199–202, https://doi.org/10.1016/B978-0-08-050755-2.50045-2
[17] M. Röger, C. Prahl, S. Ulmer, Heliostat shape and orientation by edge detection, J. Sol. Energy Eng. 132 (2) (2010) 021002, https://doi.org/10.1115/1.4001400
[18] G. Schweighofer, A. Pinz, Robust pose estimation from a planar target, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2024–2030, https://doi.org/10.1109/TPAMI.2006.252
[19] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2
[20] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nat. Methods 17 (2020) 261–272, https://doi.org/10.1038/s41592-019-0686-2
[21] G. Bradski, The OpenCV library, Dr. Dobb's J. Softw. Tools (2000).
[22] G. Terzakis, M. Lourakis, A consistently fast and globally optimal solution to the perspective-n-point problem, in: European Conference on Computer Vision, Springer International Publishing, 2020, pp. 478–494, https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123460460.pdf.
[23] K. Madsen, H. Nielsen, T. Ole, Methods for Non-linear Least Squares Problems, second ed., 2004, https://www2.imm.dtu.dk/pubdb/edoc/imm3215.pdf.
[24] J. MacQueen, Some methods for classification and analysis of multivariate observations, Proc. 5th Berkeley Symp. Math. Stat. Probab, Vol. 1: Stat, Univ. Calif. Press, Berkeley (1967) 281–297, http://projecteuclid.org/euclid.bsmsp/1200512992.

[25] A. Schnerring, R. Broda, A. Winter, M. Nieslony, J.J. Krauth, M. Röger, S. Kallio, R. Pitz-Paal, A simulation environment for UAV-based real-time condition monitoring of solar tower power plants, Sol. Energy 300 (2025) 113803, https://doi.org/10.1016/j.solener.2025.113803

[26] T.D. Barfoot, Matrix Lie groups, Cambridge University Press, 2017, pp. 205–284. https://doi.org/10.1017/9781316671528

[27] Evaluation of measurement data – guide to the expression of uncertainty in measurement, Joint Committee for Guides in Metrology, JCGM 100:2008. https://doi.org/10.59161/JCGM100-2008E

[28] M. Pargmann, M. Leibauer, V. Nettelroth, D. Maldonado Quinto, R. Pitz-Paal, Questioning the reliability of open-loop calibration methods: introducing a robust data sampling for year-round high accuracy, Sol. Energy 286 (2025) 113094, https://doi.org/10.1016/j.solener.2024.113094

[29] G. Johnston, On the analysis of surface error distributions on concentrated solar collectors, J. Sol. Energy Eng. 117 (4) (Oct 1995), https://doi.org/10.1115/1.2847843

[30] T. Collins, A. Bartoli, Infinitesimal plane-based pose estimation, Int. J. Comput. Vis. 109 (2014) 252–286, https://doi.org/10.1007/s11263-014-0725-5