

Diese Arbeit wurde vorgelegt am Deutschen Zentrum für Luft- und Raumfahrt

The present work was submitted to the German Aerospace Center

## Masterarbeit

---

Entwicklung und Evaluierung eines Bildverarbeitungsverfahrens auf einem mobilen Kamera-Sensorsystem zur Bestimmung der initialen Heliostatausrichtung

Felix Nienaber

---

„Entwicklung und Evaluierung eines Bildverarbeitungsverfahrens auf einem mobilen Kamera-Sensorsystem zur Bestimmung der initialen Helio-  
statausrichtung“ (Masterarbeit)

vorgelegt am 19. Dezember 2016 von

Felix Nienaber, Matrikelnummer: 309346

betreut durch Dipl.-Ing (FH) Oliver Kaufhold

RWTH Aachen University  
Deutsches Zentrum für Luft und Raumfahrt  
Univ.-Prof. Dr.-Ing. B. Hoffschmidt  
Linder Höhe  
D-51147 Köln  
Telefax +49 2203-601-4712  
Telefon +49 2203-601-3631  
[www.dlr.de](http://www.dlr.de)  
[info@dlr.de](mailto:info@dlr.de)

---

# Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit, bis auf die vom Betreuer gegebene Hilfestellung, selbstständig verfasst und keine anderen, als die im Literaturverzeichnis aufgeführten Quellen verwendet habe. Alle Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, habe ich als solche gekennzeichnet.

Aachen, 19. Dezember 2016

---

# Zusammenfassung

**Schlüsselwörter: CSP, CST, Heliostaten, OpenCV, Bilderkennung**

In dieser Masterarbeit wurde ein mobiles System bestehend aus Kamera und Einplatinencomputer entwickelt, mit dessen Hilfe die Bestimmung der Ausrichtung der Heliostate am Solarturm Jülich in Azimut- und Elevationsrichtung in deren Referenzposition theoretisch auf  $\leq 0,5^\circ$  genau möglich ist. Zunächst erfolgte ein Vergleich zwischen möglichen bildverarbeitenden und nicht-bildverarbeitenden Systemen hinsichtlich möglicher Genauigkeit, Umsetzbarkeit und Kosten. Diese Arbeit umfasst ebenso den Vergleich verschiedener möglicher Bildverarbeitungsverfahren (wie etwa Verfahren mit Breitband-Deskriptoren wie ORB, und einfachen Template-Match-Methoden) und die prototypische Umsetzung des Erfolg versprechendsten Systems. Mit der Beendigung der Arbeit wurde ein, für die beschriebene Anwendung geeignetes, Prototypensystem ausgearbeitet, auf dessen Basis das endgültige System in Zukunft aufgebaut werden kann. Das System konnte noch nicht statistisch aussagekräftig untersucht werden, aber die ersten Ergebnisse sind mit Genauigkeiten von  $0,97^\circ$  in Azimuts- ( $\sigma = 1,611^\circ$ ) und  $-3,2^\circ$  in Elevationsrichtung ( $\sigma = 2,367^\circ$ ) für den ersten Prototyp angemessen. Weitere Untersuchungen und Anpassungen des Systems sind vor dem Einsatz noch notwendig.



---

# Abstract

**Keywords:** CSP, CST, heliostats, OpenCV, image recognition

In this master thesis a mobile system, consisting of a camera and a single board computer, was designed and built, with which the alignment of heliostats at the Jülich solar tower can quickly be assessed in elevation as well as in azimuth with a respective theoretical accuracy of  $\leq 0,5^\circ$ . Initially, a comparison of image processing and non image processing systems ensues. The reachable accuracy, feasibility and cost are compared. This thesis also includes the comparison of different possible image processing methods (e.g. broadband descriptors like ORB, and simple template-matching-methods) and the prototypical implementation of the most promising system. Upon completion of the thesis, a prototype fit for the mentioned utilization was developed, on which the final system can be built in the future. The system could not be statistically evaluated, but the first results were promising, with accuracies of  $0,97^\circ$  ( $\sigma = 1,611^\circ$ ) in azimuth- and  $-3,2^\circ$  ( $\sigma = 2,367^\circ$ ) in elevation alignment which are appropriate for a prototype. Further evaluation and adaptation of the system are necessary before implementation.

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Methoden zur Heliostaten-Ausrichtung</b>	<b>4</b>
2.1	Nicht-optische Systeme . . . . .	4
2.1.1	Magnetkompass . . . . .	5
2.1.2	GPS . . . . .	5
2.1.3	Akustisch . . . . .	6
2.2	Optische Systeme . . . . .	6
2.2.1	Aktuelles System: Kamera im Feld . . . . .	7
2.2.2	Laser . . . . .	8
2.2.3	Tachymetrie . . . . .	8
2.2.4	Range Imaging . . . . .	9
2.2.5	Structured Light . . . . .	10
2.2.6	Drone . . . . .	10
2.2.7	Kamera am Heliostat . . . . .	11
<b>3</b>	<b>Bilderkennung</b>	<b>12</b>
3.1	Template Matching . . . . .	12
3.1.1	Erstellen des Referenzbildes . . . . .	13
3.1.2	Auswertung der <i>TM</i> -Methode . . . . .	17
3.2	Breitband-Deskriptoren . . . . .	18
3.3	Konturen . . . . .	20
<b>4</b>	<b>Aufbau des ausgewählten Systems</b>	<b>22</b>
4.1	Aufbau des Systems . . . . .	23
4.2	Bestimmung der Kameraposition am Heliostat . . . . .	24
4.3	Bestimmung der Kamera y-Achse und Verdrehung anhand des Bildes	30
4.4	Bestimmung der Kamera x- und z-Achsen anhand der bekannten Kamera y-Achse und Verdrehung . . . . .	38
4.5	Bestimmung der Winkel aus gemessenen Achsen . . . . .	41

<b>5</b>	<b>Bewertung des Algorithmus</b>	<b>45</b>
5.1	Hardwarefehler . . . . .	45
5.1.1	Verzerrung des Bildes durch die Kameralinse . . . . .	46
5.1.2	Positionierung der Kamera auf der Halterung . . . . .	46
5.1.3	Positionierung der Halterung am Spiegel . . . . .	48
5.1.4	Position und Ausrichtung des Spiegels . . . . .	49
5.2	Softwarefehler . . . . .	50
5.2.1	Genauigkeit der Bilderkennung . . . . .	51
5.2.2	Vorgehen zur Bestimmung der Testwerte . . . . .	52
5.2.3	Genauigkeit des Solvers zur Bildmittenerkennung . . . . .	53
5.2.4	Genauigkeit des Solvers zur Winkelbestimmung . . . . .	54
5.2.4.1	Genauigkeit der x-Achse . . . . .	55
5.2.4.2	Genauigkeit der y-Achse . . . . .	58
5.2.4.3	Genauigkeit der z-Achse . . . . .	60
5.3	Gesamtfehler des Systems . . . . .	63
5.4	Versuchsreihe unter realen Bedingungen . . . . .	64
5.4.1	Messungen in Referenzposition . . . . .	65
5.4.2	Messungen während der Kalibrierung . . . . .	66
<b>6</b>	<b>Zusammenfassung &amp; Ausblick</b>	<b>69</b>
	<b>Literaturverzeichnis</b>	<b>I</b>
<b>A</b>	<b>Anhang</b>	<b>IV</b>

---

# Abbildungsverzeichnis

1.1	Aufbau Heliostat . . . . .	1
2.1	Aktuelles Kalibrationsverfahren . . . . .	7
3.1	Turm-KOS . . . . .	14
3.2	Bildebene des Referenzbildes zur $TM$ . . . . .	14
3.3	Funktionierende und nicht-funktionierende $TM$ -Methode . . . . .	18
3.4	Funktionierendes $ORB$ . . . . .	19
3.5	Nicht-funktionierendes $ORB$ . . . . .	20
3.6	Konturenerkennung . . . . .	21
4.1	Umgesetztes System . . . . .	23
4.2	Darstellung $H_{yaw}$ und $\alpha$ . . . . .	25
4.3	Kamera-KOS . . . . .	25
4.4	Elevationswinkel $\varepsilon$ . . . . .	26
4.5	Azimuths- und Elevations-KOS . . . . .	27
4.6	Spiegelabmessungen . . . . .	28
4.7	Halterungsabmessungen . . . . .	28
4.8	Vergleich Turmansichten . . . . .	31
4.9	Verdrehung des Bildes . . . . .	32
4.10	Projektion der Bildebene . . . . .	33
4.11	Projektion der Bildebene und Kamera-Kugel . . . . .	34
4.12	Pixel-Abstände . . . . .	35
4.13	Abstände projizierter Punkt . . . . .	35
4.14	Durchstoßpunkte Kugeln . . . . .	36
4.15	Schnittpunkte der Kugeln . . . . .	37
4.16	Kamera- und Hilfs-KOS . . . . .	38
4.17	Schnittgerade Kamera- und Hilfs-KOS . . . . .	40
5.1	Verzerrte Aufnahme der Kamera . . . . .	47
5.2	Entzerrte Aufnahme der Kamera . . . . .	48
5.3	Verbesserung durch zweite Iteration . . . . .	50
5.4	Verbesserung durch erste Iteration . . . . .	51
5.5	Abweichungen der Bildmitten . . . . .	55

5.6	$H_{yaw}$ der betrachteten Heliostaten . . . . .	56
5.7	Einstellung der Rasten . . . . .	57
5.8	$\alpha$ aus x . . . . .	58
5.9	$\varepsilon$ aus x . . . . .	59
5.10	$\alpha$ aus y . . . . .	60
5.11	$\varepsilon$ aus y . . . . .	61
5.12	$\alpha$ aus z . . . . .	62
5.13	$\varepsilon$ aus z . . . . .	63
5.14	Messung während Kalibrierung . . . . .	67
6.1	Pixel-Karte $30^\circ$ . . . . .	70
6.2	Pixel-Karte $3^\circ$ . . . . .	71
A.1	Solveriterationen $\alpha$ aus x . . . . .	IV
A.2	Solveriterationen $\alpha$ aus x Zoom . . . . .	V
A.3	Solveriterationen $\alpha$ aus x weiterer Zoom . . . . .	VI
A.4	Solveriterationen $\alpha$ aus y . . . . .	VII
A.5	Solveriterationen $\alpha$ aus y Zoom . . . . .	VIII
A.6	Solveriterationen $\alpha$ aus y weiterer Zoom . . . . .	IX
A.7	Solveriterationen $\alpha$ aus z . . . . .	X
A.8	Solveriterationen $\alpha$ aus z Zoom . . . . .	XI
A.9	Solveriterationen $\alpha$ aus z weiterer Zoom . . . . .	XII
A.10	Solveriterationen $\varepsilon$ aus x . . . . .	XIII
A.11	Solveriterationen $\varepsilon$ aus x zoom . . . . .	XIV
A.12	Solveriterationen $\varepsilon$ aus y . . . . .	XV
A.13	Solveriterationen $\varepsilon$ aus y zoom . . . . .	XVI
A.14	Solveriterationen $\varepsilon$ aus z . . . . .	XVII
A.15	Solveriterationen $\varepsilon$ aus z zoom . . . . .	XVIII
A.16	Programmablauf . . . . .	XXI

---

# Tabellenverzeichnis

4.1	Längenangaben der Kamerhalterung und des Heliostaten . . . . .	27
5.1	x-Achse Statistiken . . . . .	58
5.2	y-Achse Statistiken . . . . .	60
5.3	z-Achse Statistiken . . . . .	62
5.4	Auswertung Messung in Referenzposition . . . . .	66
5.5	Auswertung Messung während Kalibrierung . . . . .	68
A.1	x-Achse Statistiken: $\alpha = H_{yaw} - 15^\circ, \varepsilon = F_{Pitch} - 5^\circ$ . . . . .	XIX
A.2	y-Achse Statistiken: $\alpha = H_{yaw} - 15^\circ, \varepsilon = F_{Pitch} - 5^\circ$ . . . . .	XIX
A.3	z-Achse Statistiken: $\alpha = H_{yaw} - 15^\circ, \varepsilon = F_{Pitch} - 5^\circ$ . . . . .	XIX
A.4	x-Achse Statistiken: $\alpha = H_{yaw}, \varepsilon = F_{Pitch}$ . . . . .	XX
A.5	y-Achse Statistiken: $\alpha = H_{yaw}, \varepsilon = F_{Pitch}$ . . . . .	XX
A.6	z-Achse Statistiken: $\alpha = H_{yaw}, \varepsilon = F_{Pitch}$ . . . . .	XX

---

# Abkürzungsverzeichnis

## Abkürzungen

Symbol	Definition	Einheit
<i>DSP</i>	Durchstoßpunkt	
<i>FOV</i>	Field Of View - Blickwinkel	°
<i>ORB</i>	Oriented FAST and Rotated BRIEF	
<i>SIFT</i>	Scale-Invariant Feature Transform	
<i>TM</i>	Template Matching	

## Griechische Formelzeichen

Symbol	Definition	Einheit
$\alpha$	Winkel in Azimutsrichtung	°
$\phi$	Skalierungsfaktor der Geradengleichung	
$\varepsilon$	Winkel in Elevationsrichtung	°

## Indizes

Symbol	Definition	Einheit
$\Delta P$	Positionsunterschied	
B	Bildebene	
H	Heliostatposition	
Hilf	Hilfs-Koordinatensystem	
Kamera	Kamera-Koordinatensystem	
Ost	Östlich	

<b>Symbol</b>	<b>Definition</b>	<b>Einheit</b>
P	Eckpunkt des Turmes	
P'	Projizierter Eckpunkt des Turmes	
Pixel	Pixel-Koordinatensystem	
Raste	Rasteneinstellung der Kamerahalterung	
Ref	Referenzbild	
Spiegel	Spiegel-Größe	m
T	Target-Mitte	
Turm	Turm-Koordinatensystem	
x	x-Achsen-Richtung	
y	y-Achsen-Richtung	
z	z-Achsen-Richtung	

## Lateinische Formelzeichen

<b>Symbol</b>	<b>Definition</b>	<b>Einheit</b>
$F_y$	Abstand Elevationsachse und Azimutsachse	mm
$F_{Pitch}$	Referenz Elevationswinkel	°
$G_y$	Abstand Spiegelfront und Elevationsachse	mm
$h_{Linse}$	Höhe der Linse	mm
$H_{yaw}$	Referenz Azimutswinkel	°
$l_{Kipp}$	Länge nach Kippung	mm
$l_{SK}$	Abstand Spiegelfront und Halterungs-Kippachse	mm
a	Abstand	m
b	Breite	m
HP	Gerade zwischen Heliostat und Punkt	
n	Normalenvektor	
r	Radius	m



---

# 1 Einleitung

Solarturmkraftwerke gewinnen in den heutigen Strommärkten immer mehr an Bedeutung, da diese sowohl nachhaltigen Strom als auch Wärme erzeugen können. In Solarturmkraftwerken fokussieren tausende Spiegel Sonnenlicht auf einen Punkt, in welchem ein Wärmeträgermedium erwärmt und anschließend zur Dampferzeugung verwendet wird. Die Spiegel sind auf sogenannten Heliostaten montiert, welche verfahrbar sind, um die kontinuierliche Reflexion der Sonnenstrahlen auf den Turm zu ermöglichen.

Die am Solarturm in Jülich aufgestellten Heliostaten sind horizontal und vertikal verfahrbar. In Abbildung 1.1 ist der Aufbau der Heliostatkinematik dargestellt. In der horizontalen Ebene verdreht der Azimutsmotor und in der vertikalen Ebene der Elevationsmotor den Heliostaten.

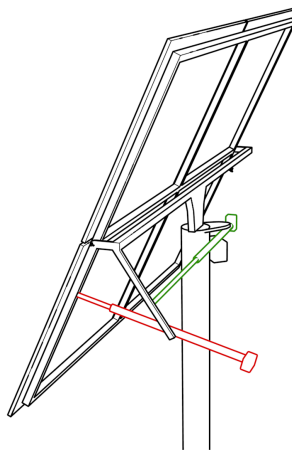


Abbildung 1.1: Aufbau eines Heliostaten. In grün dargestellt ist der Azimutsmotor, in rot der Elevationsmotor

Um die Positionsgenauigkeit der Reflexionen zu bewahren, muss eine Untersuchung der Heliostatenausrichtung in regelmäßigen Abständen und nach Reparaturarbeiten an den Heliostaten erfolgen. Das momentan am Solarturm in Jülich verwendete System zur Heliostatenausrichtung basiert auf dem Patent von Stone (Stone, 1986), bei welchem die auszurichtenden Heliostaten Sonnenlicht in einen vorgegebenen Bereich, das sogenannte Target, reflektieren. Mit Hilfe eines Kamerasystems kann der Steuerungscomputer des Heliostatenfeldes die Position der Reflexion auf dem Target erkennen und anhand des bekannten Sonnenstandes und der Position des Heliostaten auf dessen Ausrichtung zurückschließen (siehe Stone, 1986). Dieses System liefert hohe Genauigkeiten, hat aber auch einige Nachteile. Diese Methode kann nur mit direkt einfallenden Sonnenstrahlen funktionieren. Auch muss der Heliostat den vorgegebenen Bereich am Turm treffen. Um dies am Solarturm in Jülich im ersten Versuch zu erreichen, muss die Ausrichtung des Heliostaten zuvor bereits bis auf mindestens  $0,72^\circ$  bekannt sein. Ist dies nicht der Fall, so muss der Bediener den Heliostat so lange von Hand verfahren, bis die reflektierten Sonnenstrahlen am Turm sichtbar sind.

Um die Reflexion der Sonnenstrahlen auf das Target ohne weiteres Verfahren zu ermöglichen, wird in dieser Masterarbeit ein System entwickelt, welches die Ausrichtung der Heliostaten in Referenzposition bestimmt. Die geforderte Genauigkeit dieses Systems entspricht den  $0,5^\circ$  in Azimut und Elevation, um von allen Positionen im Heliostatenfeld den Turm direkt treffen zu können und zukünftige Erweiterungen des Heliostatenfeldes abzudecken. Dies geschieht mit Hilfe eines optischen Systems, bei welchem der Bediener eine Digitalkamera, die neben sichtbaren auch für nahinfrarotes Licht sensibel ist, am zu untersuchenden Heliostaten anbringt. Die Kamera erstellt eine Aufnahme zweier am Turm montierten Infrarotstrahlern. Anhand der Position der Infrarotstrahler im erstellten Foto ist die Ausrichtung der Kamera und dadurch die Ausrichtung des Heliostaten berechenbar.

Zunächst erfolgt eine Beschreibung verschiedener Methoden zur Bestimmung der Heliostatausrichtung. Im darauf folgenden Kapitel werden Methoden zur Bildererkennung genauer betrachtet und verglichen. Anschließend erfolgt die Erläuterung der Ausrichtungsbestimmung. Das letzte Kapitel schließt mit einer Auswertung der letzt-

endlich gewählten Methode sowohl anhand von computergenerierten Datenpunkten, als auch durch Versuche im Feld ab.

---

## **2 Methoden zur Heliostaten-Ausrichtung**

Es gibt verschiedene Technologien, welche die Ausrichtung eines Heliostaten bestimmen können. Die Ausrichtung eines Heliostaten wird anhand dessen Azimuts- und Elevationswinkel festgelegt. Die Azimutsrichtung entspricht einer Drehung um eine senkrechte und die Elevationsrichtung einer Drehung um eine horizontale Achse. Die untersuchten Technologien lassen sich in optische und nicht-optische Systeme unterscheiden. Im folgenden Kapitel erfolgt eine nähere Betrachtung einiger möglicher Technologien.

### **2.1 Nicht-optische Systeme**

Nicht-optische Systeme sind unabhängig von Umgebungsbedingungen, welche optische Systeme beeinflussen können. Dies sind beispielsweise Sonnenschein, Sichtverhältnisse, Schattenwurf und Sichtblockaden. Diese Systeme sind dadurch flexibler im Bezug auf die Uhrzeiten, Wetterbedingungen und Positionen, in denen sie zum Einsatz kommen können.

### 2.1.1 Magnetkompass

In einer vorangehenden Bachelorarbeit am DLR versuchte Piet van der Meulen (van der Meulen, 2015) die Heliostatausrichtung mit Hilfe eines Magnetkompasses zu bestimmen. Der Versuchsaufbau bestand dabei aus einem Magnetkompass zur Bestimmung der Azimutsausrichtung und einem Schwerefeldsensor zur Bestimmung der Elevationsausrichtung. Die Versuchsreihen zeigten dabei, dass die Sensorik die Elevationsausrichtung bis auf  $\pm 0,4^\circ$  mit einer Standardabweichung von  $0,17^\circ$  genau bestimmen konnte und somit innerhalb der vorgegeben Genauigkeit lag (siehe van der Meulen, 2015, S. 42). In der Azimutsausrichtung kam es allerdings durch den Einfluss der Metallstrukturen des Heliostatenfeldes auf den Magnetkompass zu Abweichungen in den Messergebnissen zwischen den einzelnen Heliostaten. So lag dort der Mittelwert der Abweichung zwischen Messung und realer Ausrichtung in der besten Messreihe bei  $-0,225^\circ$  bei einer Standardabweichung von  $1,367^\circ$  (siehe van der Meulen, 2015, S. 50). Der Magnetfeldsensor ist demnach nicht für diese Anwendung geeignet, weshalb das System in dieser Masterarbeit nicht weiter ausgearbeitet wurde.

### 2.1.2 GPS

Die gewünschte Genauigkeit der Heliostatausrichtung zu erhalten, ist mit Hilfe eines GPS-Systems möglich. Herkömmliche GPS-Systeme sind bis auf 7,8 m genau (siehe US Department of Defense and GPS Navstar, 2008, S. 22). RTK-GPS-Systeme (*real-time kinematic GPS*) hingegen weisen eine höhere Genauigkeit auf, da bei diesen eine zusätzliche Basisstation in der Nähe des zu messenden Punktes hilft, atmosphärische Signalstörungen herauszufiltern. Ein Eigenbau RTK-GPS-System kann die Position eines Messpunktes in Referenz zu einer zweiten festgesetzten GPS-Antenne bis auf 1 cm genau bestimmen (siehe Takasu and Yasuda, 2009, S. 6). Da zwei Eckpunkte des Heliostaten vermessen werden müssen, addieren sich die Fehler zu 2 cm, was in Azimutsrichtung bei der Heliostatenbreite von 3,4 m einer Genauigkeit von  $\Delta\alpha = \arcsin\left(\frac{2\text{cm}}{3,4\text{m}}\right) = 0,32^\circ$  entspricht. Professionelle RTK-Systeme erreichen eine Genauigkeit von 3 mm (siehe Leica Geosystems AG, 2015, S. 3). An zwei Mess-

punkten entspräche dies dann einer gesamten Genauigkeit von  $0,1^\circ$ . Für die Elevationsmessung ergibt sich mit dem professionellen RTK-System eine Genauigkeit von  $\Delta\alpha = \arcsin\left(\frac{6\text{mm}}{2,5\text{m}}\right) = 0,14^\circ$ . Für ein Eigenbau RTK-System liegt die Genauigkeit bei maximal  $0,46^\circ$  und ist somit noch innerhalb der Anforderungen. Aufgrund des vergleichsweise hohen Kostenaufwands wurde dieses System allerdings nicht umgesetzt.

### 2.1.3 Akustisch

Ein akustischer Sensor kann seinen Abstand zu einer akustisch reflektierenden Fläche bis auf 3 mm genau in einem maximalen Abstand von 6 m bestimmen (siehe Baumer AG, 2015). Dies entspricht der Genauigkeit des professionellen GPS-Systems von  $0,1^\circ$ . Auch misst dieses System bei entsprechendem Aufbau sowohl die Azimuts- als auch Elevationsausrichtung. Allerdings kann die Messung in diesem System nur mit einer separat aufgestellten Messvorrichtung geschehen, welche die Sensoren auf den Heliostaten ausrichtet. Die Positionierung und Ausrichtung einer solchen mobilen Messvorrichtung wirft allerdings wiederum die selben Probleme auf, welche das akustische System lösen sollte, weshalb dieses System auch verworfen wurde.

## 2.2 Optische Systeme

Optische Systeme können höhere Genauigkeiten erreichen als nicht-optische Systeme, da Licht beispielsweise nicht durch elektromagnetische Felder abgelenkt werden kann und sich somit in den am Solarturm Jülich betrachteten Größenordnungen immer gerade ausbreitet. Optische Systeme verwenden Sensoren, welche licht-sensitiv sind. Diese können sowohl einfache Bauteile zur Helligkeitsmessung, als auch komplexe Kamerasysteme sein.

### 2.2.1 Aktuelles System: Kamera im Feld

Das aktuell am Solarturm Jülich verwendete System zur Heliostatausrichtung verwendet eine im Feld angebrachte Kamera, welche auf das dafür vorgesehene, am Turm angebrachte, Target ausgerichtet ist. Ein automatisches System steuert einzelne Heliostaten so, dass diese nacheinander Sonnenstrahlen auf das Target reflektieren, wie in Abbildung 2.1 dargestellt. Anhand der Heliostatposition, der Position der reflektierten Sonnenstrahlen und dem Sonnenstand errechnet das System die Ausrichtung des Heliostaten zum Zeitpunkt der Messung. Anhand mehrerer solcher Messpunkte kann das Geometriemodell des Heliostaten, welches den Zusammenhang zwischen Motorpositionen und der Ausrichtung des Heliostaten darstellt, berechnet werden (siehe Stone, 1986).

Ein Nachteil dieses Systems ist allerdings, dass es nur bei ausreichendem Sonnenschein funktioniert. Weiterhin muss der Heliostat bereits so genau ausgerichtet sein, dass er die Sonnenstrahlen auf das Target reflektieren kann, um eine automatische Ausrichtungsbestimmung zu ermöglichen. Ist dies nicht der Fall, so muss der Bediener des Systems den Heliostaten von Hand verfahren, bis die Reflexion durch die Kamera sichtbar wird. Bei schlechter initialer Ausrichtung des Heliostaten kann dieses Verfahren mehrere Minuten in Anspruch nehmen.

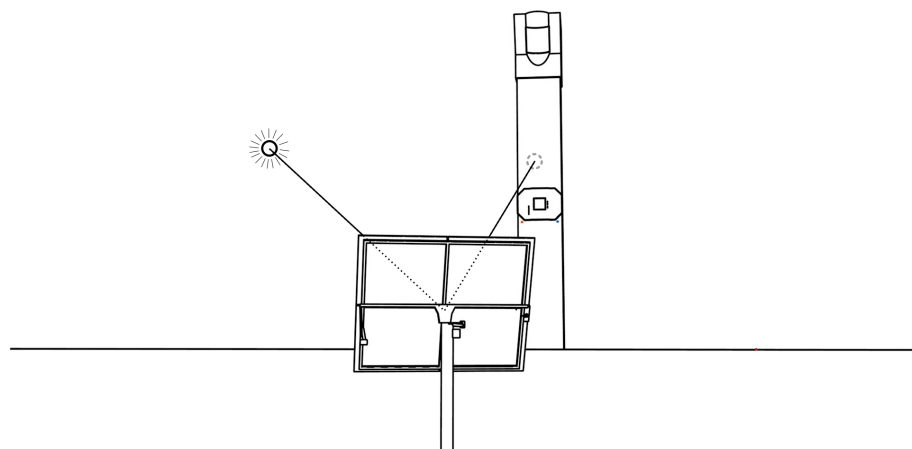


Abbildung 2.1: Aktuelles Kalibrationsverfahren, in welchem ein Heliostat Sonnenstrahlen auf das Target am dem Turm reflektiert

### 2.2.2 Laser

Eine weitere Möglichkeit ist ein System, bei dem ein Laser an einem Heliostaten angebracht wird, welcher einen fest montierten Reflektor anstrahlt und das reflektierte Licht in einem Sensor auffängt, um zu registrieren, wann der Laserstrahl auf den Reflektor trifft. Um die geforderte Genauigkeit von  $0,5^\circ$  zu erhalten, darf für die Entfernung von 350 m im hintersten Feld, ein Reflektor mit einem maximalen Durchmesser von 3 m verwendet werden. Im vorderen Bereich des Feldes ist ein maximaler Durchmesser von 0,3 m erforderlich. Ein Vorteil dieses Systems besteht in seiner kompakten Bauweise. Ein mobiles System könnte dabei lediglich aus einem Einplatinencomputer, einem Lichtsensor zur Reflexionsregistrierung und einem Laser-Pointer bestehen. Probleme treten in diesem System allerdings dadurch auf, dass eine hohe Blendgefahr für den Bediener besteht. Auch ist für dieses System notwendig, dass der Heliostat so ausgerichtet wird, dass der Laser-Pointer auf den Reflektor strahlt. Dies ist zu vermeiden, da die Ausrichtungsbestimmung in Referenzposition geschehen soll. Ein Laser-Messverfahren ohne Heliostatbewegung wäre durch einen Laser-Pointer, welcher sich automatisch durch Schrittmotoren auf den Reflektor ausrichtet, realisierbar. Da durch die Schrittmotoren weitere Fertigungs- und Messungenauigkeiten auftreten, ist allerdings von einem verfahrenbaren Laser abzuraten.

### 2.2.3 Tachymetrie

Ein Tachymeter eignet sich auch für diese Anwendung. Diese bestimmen mit Hilfe eines Laserstrahls die Entfernung zu einem reflektierenden Material. Die Entfernungsmessung geschieht dabei entweder über eine Laufzeitmessung oder die Phasenverschiebung zwischen ausgesandten und reflektierten Strahlen (siehe Lang, 2012, S. 2). Nach Herstellerangaben erreichen Tachymeter bei Abstandsmessungen eine Genauigkeit von  $10 \mu\text{m}$  (siehe Hexagon Metrology, 2009, S. 15). Dadurch beträgt die theoretische Genauigkeit der Ausrichtungsmessung von unter  $0,1^\circ$ , ohne Berücksichtigung der Genauigkeit der Positionierung des Tachymeters. Allerdings ist die wirklich gemessene Abweichung deutlich größer, da winkelabhängig Abweichungen von ca. 1 cm



( $\cong 0,3^\circ$  Abweichung) bei einer Messung ohne angebrachte Reflektoren zu erwarten sind. Ist eine entsprechende Reflektorfolie angebracht, so erhöht sich die Genauigkeit der Messung auf etwa 5 mm ( $\cong 0,15^\circ$  Abweichung) (siehe Lang, 2012). Problematisch ist hierbei wiederum die Positionsbestimmung, da je nach Abstand zwischen Tachymeter und Heliostat die Genauigkeit der Messungen dadurch beeinflusst wird. Auch ist die Messung nur möglich, wenn die zu untersuchende Oberfläche die Laserstrahlen nach dem Auftreffen auf den Heliostaten wieder in Richtung des Tachymeters zurück reflektiert. Dies ist nur möglich, wenn die nicht spiegelnde Rückseite des Heliostaten vermessen wird, oder zusätzliche Retroreflektoren auf der Spiegelfläche angebracht werden. Die maximale Reichweite eines Tachymeters beträgt 320 m (siehe Hexagon Metrology, 2009, S. 15). Somit wäre es am Solarturm Jülich möglich, das gesamte Heliostatenfeld von einem Punkt aus zu untersuchen. Dieser kann dabei so gewählt werden, dass der minimale Abstand zwischen Tachymeter und Heliostat bei 15 m liegt, um den Einfluss der Positionierung des Tachymeters auf das Messergebnis zu verringern. Bestimmt der Bediener in diesem Abstand die Position des Tachymeters mit Hilfe eines herkömmlichen GPS-Systems bis auf 7,8 m genau, so ergibt sich ein maximaler Fehler von  $\Delta\alpha = \arctan\left(\frac{7,8\text{ m}}{15\text{ m}}\right) = 27,5^\circ$  in der Messung. Erfolgt die Positionsbestimmung anhand eines professionellen RTK-GPS-Systems mit 3 mm Genauigkeit, so verringert sich dieser Fehler auf  $\Delta\alpha = 0,011^\circ$  (siehe Abschnitt 2.1.2). Für die Anwendung am Solarturm in Jülich wäre ein solches System generell geeignet, solange die Ergebnisse der Abstandsmessung automatisch ausgegeben und weiter verarbeitet werden können. Ist dies nicht der Fall, so erhöht sich der Bedieneaufwand um ein vielfaches. Auch eignet sich ein Tachymeter zur Vermessung vieler Heliostaten in einem kurzen Zeitraum, wenn das Tachymeter außerhalb des Feldes in einer erhöhten Position angebracht ist.

### 2.2.4 Range Imaging

Range Imaging Systeme, wie etwa die X-Box Kinect Kamera der Firma Microsoft, senden Laser-Strahlen aus, und nehmen das auf die Umgebung projizierte Muster mit einer Kamera auf und werten dieses aus. So kann das System Tiefeninformationen der Umgebung gewinnen. In 3,5 m Entfernung zeigt die Kinect Kamera noch eine

Genauigkeit von etwa 4 mm (siehe Yang et al., 2015, S. 8) und ist somit mit der Genauigkeit eines Tachymeters vergleichbar. Allerdings eignet sich die Kinect, ebenso wie Tachymeter, nicht für die Messung von spiegelnden Gegenständen. Wird die Kinect Kamera verwendet, so kann ein Range Imaging System kostengünstig umgesetzt werden. Ein Nachteil dieser Technologie ist wiederum, dass die Positionsbestimmung der Kamera die Genauigkeit des Systems stark beeinflussen kann.

### 2.2.5 Structured Light

Das Structured Light Verfahren verwendet ein veränderliches Muster, welches ein Beamer auf eine Fläche, wie etwa das Target des Solarturmes, wirft. Die Spiegelung dieses Musters im Heliostaten wird durch eine Kamera aufgenommen und die Verzerrung des Musters untersucht. Dieses System findet bereits Anwendung in der Untersuchung der Spiegeloberfläche, mit einer Genauigkeit im Mikrometer-Bereich (siehe Xiao et al., 2012). Allerdings ist dieses Verfahren für die Messung der Heliostatausrichtung ungeeignet, da der Heliostat verfahren werden muss, um das gespiegelte Muster von einer fest installierten Kamera aus sehen zu können. Wird eine mobile Kamera verwendet, so kann die Messung auch in Referenzposition erfolgen, was wiederum eine Positions- und Ausrichtungsmessung der Kamera erforderlich macht.

### 2.2.6 Drone

Eine autonom fliegende Drone hätte die Möglichkeit, viele Heliostaten in einer Messreihe zu untersuchen. Für linienfokussierende Solarkraftwerke wurde durch das DLR bereits ein Dronensystem entwickelt, welches die Spiegeloberfläche vermisst (siehe Prah et al., 2013). Theoretisch kann dieses System auch zur Positionsbestimmung der Ecken einzelner Heliostate verwendet werden, da dies bereits einen Teil der Funktionalität darstellt. Allerdings kann hierbei nur die Position der Ecken in der horizontalen Ebene ermittelt werden, sodass die Ausrichtungsbestimmung in Elevationsrichtung an Genauigkeit verliert. Auch müsste der zu untersuchende Heliostat in einer fast waagerechten Position ausgerichtet sein, um die Eckenerkennung zu er-

möglichen und eine angemessene Genauigkeit des Systems zu garantieren. Da dieses System somit sowohl eine Verdrehung des Heliostaten außerhalb seiner Referenzposition benötigt, als einen großen Messaufwand für einzelne Heliostaten aufweist, ist es für die geforderte Anwendung nicht geeignet.

### **2.2.7 Kamera am Heliostat**

Die vielversprechendste Methode zur Ausrichtungsbestimmung ist ein auf dem Heliostat angebrachtes Kamerasystem, welches durch Bilderkennungssoftware seine Ausrichtung ermitteln kann. Die Kamera ist dabei an einer Halterung befestigt, welche eine genaue und einfache Positionierung am Heliostaten ermöglicht. Die Bilderkennung untersucht die Fotos auf bekannte Merkmale, wie etwa Gebäude oder extra für diesen Zweck angebrachte Markierungen, und kann anhand der Position dieser Merkmale im Bild die Kameraausrichtung bestimmen. Open Source Methoden zur Bilderkennung, welche auch von Einplatinencomputern verwendet werden können, sind frei verfügbar und alle Funktionen sind ausführlich dokumentiert. Dadurch ist eine kostengünstige Umsetzung dieses Systems möglich.

Im Folgenden wird diese Methode näher untersucht. Zunächst erfolgt ein Vergleich verschiedener möglicher Bilderkennungsmethoden und anschließend die Erläuterung und Auswertung des letztendlich umgesetzten Systems.

---

## 3 Bilderkennung

Bei der Bilderverarbeitung gibt es verschiedene Möglichkeiten, ein gesuchtes Objekt im Bild zu finden und zuzuordnen. Die während dieser Masterarbeit untersuchten Methoden beschränken sich allerdings auf die transponierende Methode, Breitband-Deskriptoren und einfache Konturen.

### 3.1 Template Matching

Bei der *TM*-Methode (Template Matching - Schablonenabgleich) setzt ein Algorithmus das gesuchte Referenzbild auf jeden Punkt im gemachten Foto. Das Referenzbild ist in diesem Fall ein automatisch generiertes Bild, in welchem die gesuchten sichtbaren Kanten des Turmes abgebildet sind. Der Algorithmus vergleicht die Intensitäten beider Bilder in den überlagerten Punkten und sucht die maximale Übereinstimmung in allen Positionen des Referenzbildes auf dem gemachten Foto (siehe Mordvintsev and K., 2013f). Es sind unterschiedliche Berechnungsmethoden zur Bestimmung der größten Überlagerung verfügbar, welche beispielsweise die quadratischen Abweichungen berechnen und dadurch unterschiedliche Ergebnisse erzeugen (siehe OpenCV.org Foundation, 2016).

Zur testweisen Anwendung dieser Methode in der Heliostatenausrichtung wurde die Form des Turmes mit den im Bild gefundenen Kanten verglichen. Die Kanten des gemachten Fotos ermittelte die Methode zunächst mit Hilfe der Canny-Edge-Detection. Nun vergleicht die *TM* das Kanten-Foto mit dem speziell generierten Referenzbild. Zur Erstellung des Referenzbildes ist es nötig, die projizierte Form des Turmes aus Sicht des Heliostaten zu bestimmen.

### 3.1.1 Erstellen des Referenzbildes

Das Erstellen des Referenzbildes geschieht anhand von Vektorrechnungen mit Hilfe der bekannten Geometrien und Positionen. Hierbei ist zur Vereinfachung der Berechnung davon auszugehen, dass die Kamera auf den Mittelpunkt des Targets zeigt. Das Target besteht aus am Turm angebrachten Markierungen, welche die Ausrichtungsbestimmung der Heliostaten im Betrieb, mit Hilfe des in Abschnitt 2.2.1 vorgestellten Verfahrens, ermöglichen. Der Vektor, welcher von der Target-Mitte auf die Heliostatposition zeigt, ist als Normalenvektor der Bildebene anzunehmen. Die Bildebene ist abschließend in ein Pixelkoordinatensystem zu überführen, um auf die Form des Turmes in der Darstellung zu kommen. Die Koordinaten aller verwendeten Punkte sind zunächst im Turm-KOS (Koordinatensystem) gegeben. Der Ursprung dieses Koordinatensystems liegt mittig unter dem Hauptreciever des Solarturmes, auf Höhe des Bodens, wie es in Abbildung 3.1 zu sehen ist.

Der Normalenvektor  $\vec{n}$  errechnet sich anhand der Heliostatposition  $\vec{x}_H$  und der Position der Target-Mitte  $\vec{x}_T$ . Als Heliostatposition ist in diesem Fall die Position des Azimuts-Koordinatensystems, wie es in Abbildung 4.5 dargestellt ist, festgelegt. Somit lautet die Gleichung des Normalenvektors

$$\vec{n} = \begin{pmatrix} x_H - x_T \\ y_H - y_T \\ z_H - z_T \end{pmatrix}. \quad (3.1)$$

Die Ebene, welche durch die Target-Mitte geht und deren Normalenvektor  $\vec{n}$  ist, ergibt sich durch die Gleichung

$$(\vec{x} - \vec{x}_T) * \vec{n} = 0. \quad (3.2)$$

Ausformuliert ergibt sich somit die Ebenengleichung der in Abbildung 3.2 dargestellten Bildebene zu

$$(x - x_T) * (x_H - x_T) + (y - y_T) * (y_H - y_T) + (z - z_T) * (z_H - z_T) = 0. \quad (3.3)$$

Die allgemeine Geradengleichung einer Gerade  $\overrightarrow{HP}$  zwischen dem Heliostaten und

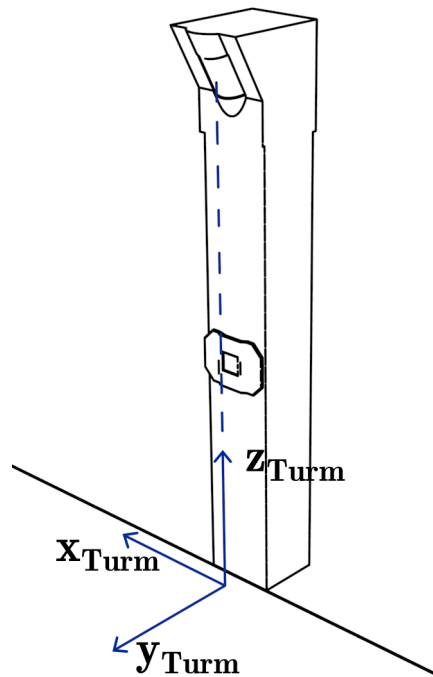


Abbildung 3.1: TurmKOS, dessen Ursprung auf Bodenhöhe unterhalb des Hauptreievers liegt

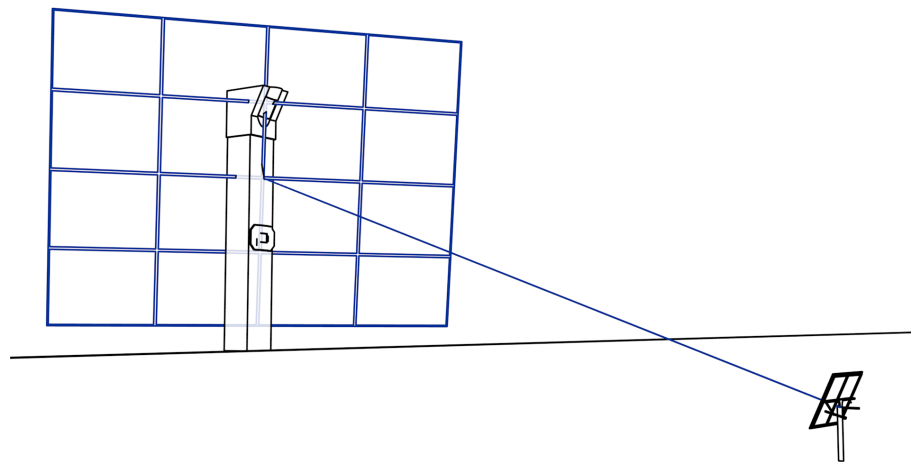


Abbildung 3.2: In blau dargestellt ist die projizierte Bildebene, in welche die Eckpunkte des Turmes projiziert werden. Die verbindende Linie zwischen der Bildebene und dem Heliostaten ist der Normalenvektor der Bildebene

den gegebenen Eckpunkten des Turmes  $\overrightarrow{x_{P,i}}$ , welche in die Bildebene gelegt werden sollen, ergibt sich mit dem Skalierungsfaktor  $\phi$  zu

$$\overrightarrow{HP} = \begin{pmatrix} x_H \\ y_H \\ z_H \end{pmatrix} + \phi \begin{pmatrix} x_{P,i} - x_H \\ y_{P,i} - y_H \\ z_{P,i} - z_H \end{pmatrix}. \quad (3.4)$$

In die Ebenengleichung eingesetzt und nach  $\phi$  umgeformt ergibt sich der endgültige Skalierungsfaktor  $\phi_{P'}$ , welcher in die Geradengleichung eingesetzt den Durchstoßpunkt der Geradengleichung in der Bildebene ergibt, zu

$$\phi_{P'} = \frac{-((x_H - x_T)^2 + (y_H - y_T)^2 + (z_H - z_T)^2)}{(x_H - x_T) * (x_p - x_H) + (y_H - y_T) * (y_p - y_H) + (z_H - z_T) * (z_p - z_H)}. \quad (3.5)$$

In die Geradengleichung eingesetzt ergibt sich somit der die Position des projizierten Punktes  $P,i'$  in der Bildebene zu

$$\overrightarrow{x_{P,i'}} = \begin{pmatrix} x_H \\ y_H \\ z_H \end{pmatrix} + \phi_{P'} \begin{pmatrix} x_{P,i} - x_H \\ y_{P,i} - y_H \\ z_{P,i} - z_H \end{pmatrix}. \quad (3.6)$$

Um nun auf die genauen Punkte in dem Koordinatensystem der Bildebene zu kommen, ist das Koordinatensystem des Turmes in die Target-Mitte zu verschieben, und so zu drehen, dass die y-Achse auf den Heliostaten zeigt und die x-Achse parallel zur ursprünglichen x-y-Ebene bleibt. Dies entspricht einer Drehung um die z-Achse um den negativen Azimutsunterschied  $\alpha_{\Delta P}$ , gefolgt von einer Drehung um die neue x-Achse um den negativen Elevationsunterschied  $\varepsilon_{\Delta P}$ . Diese Winkel entsprechen nicht der Ausrichtung des Heliostaten, sondern ergeben sich aus dem Positionsunterschied  $\Delta P$  zwischen der Target-Mitte und der zuvor verwendeten Heliostaten-Pylon Position zu

$$\alpha_{\Delta P} = -\arctan\left(\frac{x_H - x_T}{y_H - y_T}\right) \quad (3.7)$$

$$\varepsilon_{\Delta P} = \arctan\left(\frac{z_H - z_T}{\sqrt{(x_H - x_T)^2 + (y_H - y_T)^2}}\right). \quad (3.8)$$

Die Transformationsgleichung ergibt die Koordinaten der projizierten Punkte  $\vec{x}_{B,i}$  im Koordinatensystem der Bildebene somit zu

$$\begin{pmatrix} x_{B,i} \\ y_{B,i} \\ z_{B,i} \end{pmatrix} = \begin{pmatrix} \cos(\alpha_{\Delta P}) & -\sin(\alpha_{\Delta P}) & 0 \\ \cos(\varepsilon_{\Delta P}) \sin(\alpha_{\Delta P}) & \cos(\varepsilon_{\Delta P}) \cos(\alpha_{\Delta P}) & -\sin(\varepsilon_{\Delta P}) \\ \sin(\varepsilon_{\Delta P}) \sin(\alpha_{\Delta P}) & \sin(\varepsilon_{\Delta P}) \cos(\alpha_{\Delta P}) & \cos(\varepsilon_{\Delta P}) \end{pmatrix} * \begin{pmatrix} x_{P,i'} - x_T \\ y_{P,i'} - y_T \\ z_{P,i'} - z_T \end{pmatrix}. \quad (3.9)$$

Hier entspricht nun der Punkt (0,0,0) in der Bildebene dem Mittelpunkt des Targets. Da die x-z-Ebene des Bildebenen-KOS der virtuellen Bildebene entspricht, gilt für alle korrekt projizierten und transformierten Eckpunkte

$$y_{B,i} = 0. \quad (3.10)$$

Um die Turmumrisse darstellen zu können, sind alle Eckpunkte des Turmes zuerst durch Gleichung 3.6 in die virtuelle Bildebene zu projizieren, und anschließend durch Gleichung 3.9 in das Koordinatensystem der Bildebene zu überführen.

Zur Berechnung der Pixelwerte aller Eckpunkte des Vergleichsbildes anhand der Koordinaten in der Bildebene, sind zunächst die Höhe und Breite des Vergleichsbildes zu bestimmen. Dazu ist zunächst der Abstand  $a_{H,T}$  des Heliostaten zur Targetmitte zu bestimmen, durch den Zusammenhang

$$a_{H,T} = \sqrt{(x_H - x_T)^2 + (y_H - y_T)^2 + (z_H - z_T)^2}. \quad (3.11)$$

Nun sind die Winkel zwischen dem Vektor vom Heliostat auf die äußersten projizierten Punkte und der Target-Mitte für positive und negative x-Richtung,  $\alpha_1$  und  $\alpha_2$ , durch die maximalen und minimalen auftretenden x-Werte aller betrachteten Punkte  $\vec{x}_B$  zu bestimmen. Die Winkel ergeben sich durch

$$\alpha_1 = \arctan \left( \frac{MAX(x_B)}{a_{H,T}} \right) \quad (3.12)$$

$$\alpha_2 = \arctan \left( \frac{MIN(x_B)}{a_{H,T}} \right) \quad (3.13)$$



Mit der Breite des Fotos in Pixeln  $b_{Bild,Pixel}$  und der Größe des horizontalen  $FOVs$  (Field Of View - Blickwinkel)  $\angle FOV$  ist die Breite des Referenzbildes  $b_{Ref,Pixel}$  berechenbar durch

$$b_{Ref,Pixel} = \frac{b_{Bild,Pixel}}{2} * \left( \frac{\tan(\alpha_1)}{\angle FOV/2} + \frac{\tan(\alpha_2)}{\angle FOV/2} \right). \quad (3.14)$$

Die Höhe des Referenzbildes  $h_{Ref,Pixel}$  errechnet sich aus dem Verhältnis der auftretenden Unterschiede in x- und z-Richtung

$$h_{Ref,Pixel} = \frac{MAX(z_B) - MIN(z_B)}{MAX(x_B) - MIN(x_B)} * b_{Ref,Pixel}. \quad (3.15)$$

Die Pixelkoordinaten  $(x_{Pixel,i}, z_{Pixel,i})$  eines einzelnen Bildpunktes  $\vec{x}_{B,i}$  ergeben sich durch

$$x_{Pixel,i} = \frac{b_{Ref,Pixel}}{MAX(x_B) - MIN(x_B)} * (MAX(x_B) - x_{B,i}) \quad (3.16)$$

$$z_{Pixel,i} = \frac{h_{Ref,Pixel}}{MAX(z_B) - MIN(z_B)} * (MAX(z_B) - z_{B,i}). \quad (3.17)$$

Zur Berechnung des Vergleichsbildes sind die Pixel-Positionen für alle wichtigen Eckpunkte des Turmes vorzugeben. Anhand dieser kann die Bilderkennungssoftware ein Referenzbild aller sichtbaren Turmkanten erstellen, welches mit dem auf Kanten reduzierten Vergleichsbild abzugleichen ist.

#### 3.1.2 Auswertung der *TM*-Methode

Durch die Reduktion des Bildes auf die vorhandenen Kanten, ist die Untersuchung durch die *TM*-Methode erst möglich. Zum Beispiel durch Canny Edge Detection wird das Bild in eine schemenhafte Version umgerechnet. Diese verwendet eine Kombination aus Sobel-Operatoren und Schwellenwertbildung, um die möglichen Kanten des

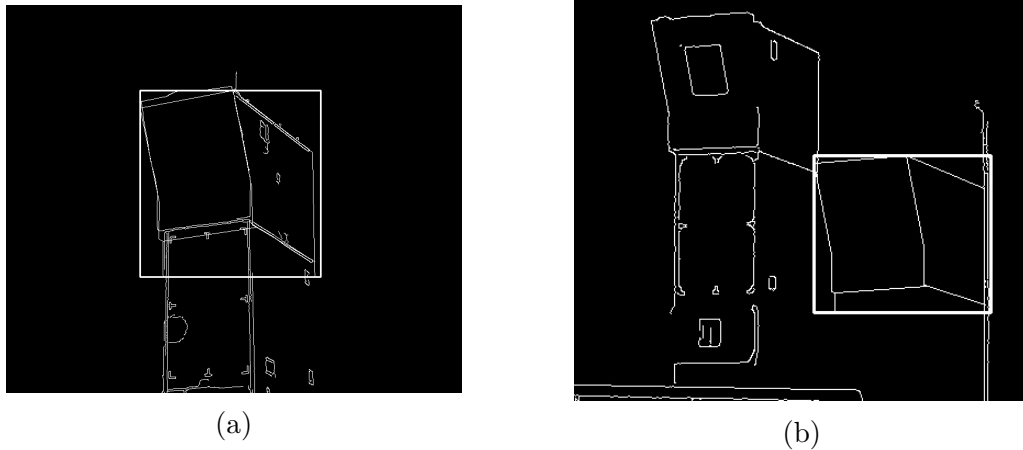


Abbildung 3.3: Beispiele der *TM-Methode*, (a) mit funktionierender Erkennung, (b) mit fehlerhafter Erkennung. Im weißen Rechteck befinden sich jeweils die durch den Algorithmus berechneten Referenzbilder. In (a) wurde dies korrekt dem Turmumriss zugeordnet, während in (b) eine im Feld stehende Antenne fälschlicherweise durch das System dem Turm zugeordnet wurde

Bildes so weit zu reduzieren, dass es diese mit einer Breite von einem Pixel darstellen kann (siehe Mordvintsev and K., 2013d).

Aufgrund der Unzuverlässigkeit gegenüber Verdrehungen und wechselnden Lichtbedingungen (welche nicht immer alle Kanten für die Kamera sichtbar machen konnten), ist diese Methode nicht für die in dieser Masterarbeit geforderte Anwendung geeignet. So ist beispielsweise in Abbildung 3.3b zu sehen, dass schon bei geringen Unterschieden zwischen Referenz- und Vergleichsbild andere im Bild gefundene Kanten fälschlicherweise dem Turm zugeordnet werden.

## 3.2 Breitband-Deskriptoren

Breitband-Deskriptoren wie *SIFT* (Scale-Invariant Feature Transform) und *ORB* (Oriented FAST and Rotated BRIEF) haben eine zur *TM-Methode* vollkommen unterschiedliche Funktionsweise. Breitband-Deskriptor-Methoden vergleichen viele markante Stellen, sogenannte *features*, des Referenzbildes und gemachten Fotos und

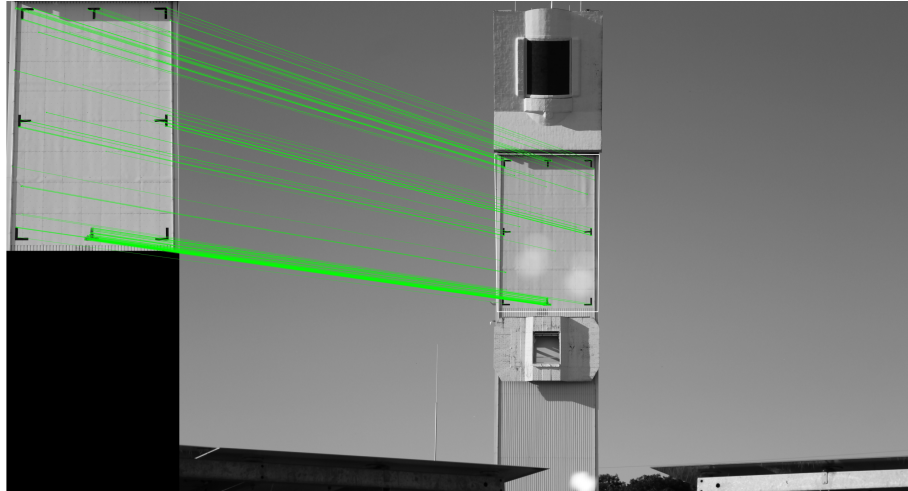


Abbildung 3.4: Funktionierendes Beispiel der *ORB* mit angemessener Genauigkeit. Der vergrößerte Bildausschnitt wurde aus diesem Bild entnommen und durch die Bilderkennung korrekt zugeordnet

geben die plausibelste Kombination dieser *features* als Abbildung des Referenzbildes im Foto aus. Diese Methoden funktionieren sehr gut bei Objekten mit vielen Markanten Stellen, wie etwa speziell dafür entwickelten Positionsmarkern. Die *ORB*-Methode kann dabei besser Verkippungen und Verdrehungen berücksichtigen, als es die *SIFT*-Methode zulässt. Des Weiteren ist die *ORB*-Methode Open-Source, während die *SIFT*-Methode patentiert, und somit kostenpflichtig bei der kommerziellen Verwendung ist (siehe Mordvintsev and K., 2013c,b)

Für die Aufgabenstellung dieser Masterarbeit sind Breitband-Deskriptoren allerdings ungeeignet. Bei großen Entfernungen zum Turm sind nur noch wenige der *features* sichtbar, welche im gegebenen Referenzbild vorhanden sind. Ebenso sorgen unterschiedliche Lichtbedingungen dafür, dass die Turm-*features*, wie beispielsweise die markierten Eckpunkte des Targets, teilweise im Schatten liegen. Durch solche Bedingungen ändert sich die Anzahl und Position der *features*, welche die Eckpunkte als solche kennzeichnen sollten. In Abbildung 3.4 ist eine 100%ige Erkennung des Referenzbildes im Vergleichsbild zu sehen, während in Abbildung 3.5 sich zu wenige der im Referenzbild gefundenen *features* im Vergleichsbild wiederfinden, sodass eine falsche Zuordnung erfolgt.

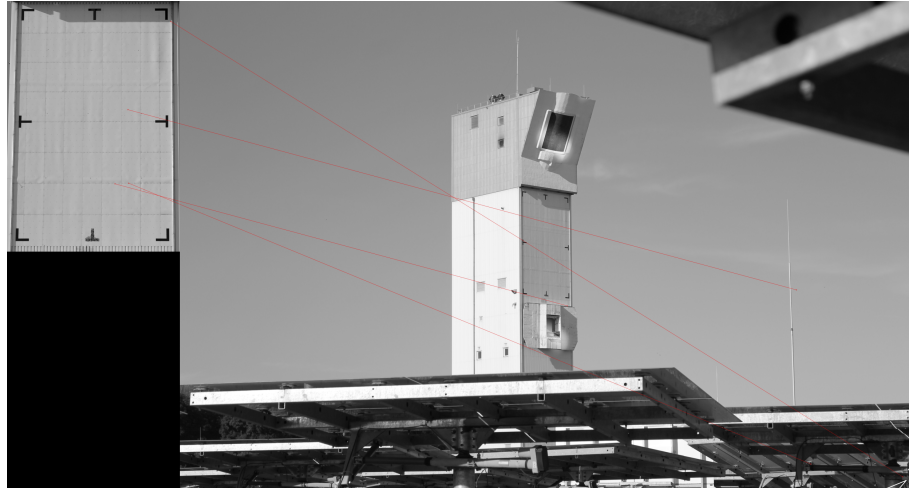


Abbildung 3.5: Nicht-funktionierendes Beispiel der *ORB*. Einzelne *features* des Referenzbildes wurden zusammenhangslosen Punkten im Bild zugeordnet, wodurch die Bilderkennung fehlschlug.

### 3.3 Konturen

Die Konturerkennung sucht zusammenhängende Pixelgruppen der selben Helligkeit (siehe Mordvintsev and K., 2013e). In diesem Fall berechnet die Methode zunächst eine binarisierte Version des zu untersuchenden Fotos, in welchem die Methode alle Punkte, deren Helligkeit über einem vorgegebenen Grenzwert liegt, als weiße Pixel, und alle darunter als schwarze Pixel festsetzt. Um kleine Pixelgruppen zu vernachlässigen, erodiert die Methode zunächst das binäre Foto und erweitert es anschließend wieder. Diese Vorgänge verringern und vergrößern die Umrisse der im Bild vorhandenen Pixelgruppen. Durch die Erosion verschwinden kleine Pixelgruppen und können somit bei der Erweiterung nicht wiederhergestellt. In Abbildung 3.6 ist das Ergebnis einer solchen Konturerkennung dargestellt, bei dem zusätzlich alle Konturenpaare markiert wurden, welche den selben Abstand aufweisen, wie die auf der Forschungsebene angebrachten Strahler, welche in Kapitel 4 näher beschrieben sind.

Im endgültigen System wird die Konturerkennung verwendet, da diese auch auf einem Einplatinencomputer in wenigen Sekunden die zu untersuchenden acht Megapixel Bilder verarbeiten, und die Positionen der verwendeten Infrarotstrahler aus-

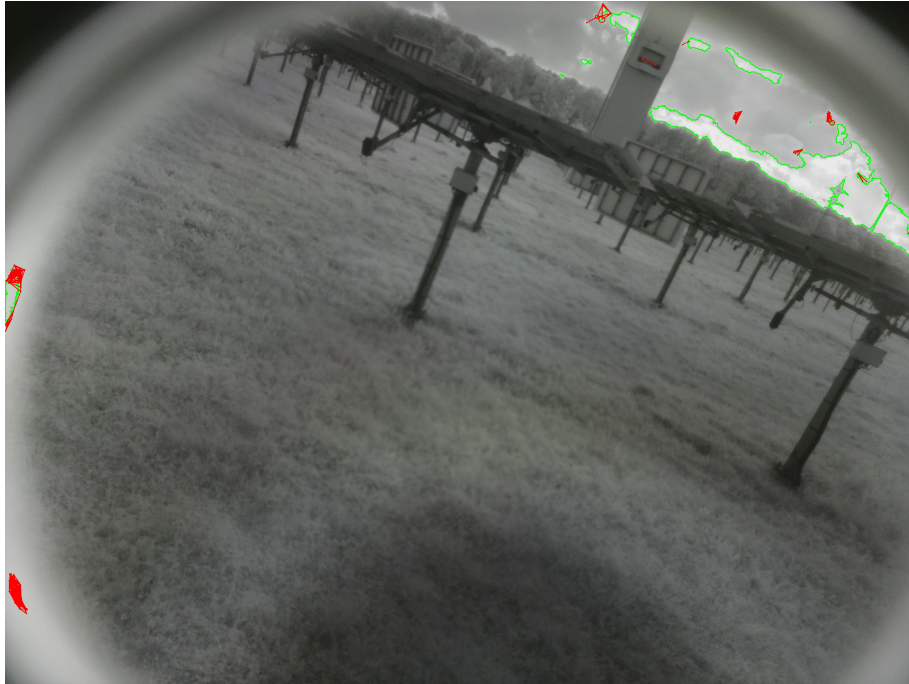


Abbildung 3.6: Beispiel der Konturenerkennung. In grün dargestellt sind die erkannten Konturen. In rot dargestellt sind alle Konturenpaare, welche den selben Abstand aufweisen, wie die auf der Forschungsebene angebrachten Strahler. Die an der Forschungsebene angebrachten Strahler sind im Bild auch erkannt und somit auch mit einer roten Linie verbunden worden

geben kann. Im folgenden Kapitel wird näher erläutert, wie anhand der bekannten Pixel Position der Strahler auf die Ausrichtung der Kamera, und anschließend auf die Ausrichtung des Heliostaten rückgeschlossen werden kann.

---

## 4 Aufbau des ausgewählten Systems

Im endgültigen System kommt eine infrarotsensitive Digitalkamera mit Bandpassfilter zum Einsatz, welche ein Foto von testweise befestigten IR-Strahlern (Infrarot-Strahlern) am Turm macht. Diese IR-Strahler strahlen Licht mit einer Wellenlänge von 940 nm aus und wurden ursprünglich zur Ausleuchtung von Überwachungskameras verkauft. Der Bandpassfilter ist dabei so gewählt, dass er lediglich einen  $940 \pm 5$  nm Wellenlängenbereich durchlässt, welcher den Abstrahlbereich der IR-Strahler abdeckt. Der Bandpassfilter verringert die Bildinformationen soweit, dass lediglich die Infrarotstrahler und direkte oder reflektierte Sonnenstrahlen zu sehen sind. Bewölkungsabhängig kann es also dazu kommen, dass die Strahler die hellsten Punkte im Bild darstellen, oder dass aufgrund starken Sonneneinfalls viele weitere helle Bildpunkte sichtbar sind, welche auch als mögliche Strahler erkannt werden. Aufgrund der bekannten Positionen der Strahler und der Heliostatposition, ist die Größe der Strahler im Bild sowie deren Abstand in Pixeln mit Hilfe des in Abschnitt 3.1.1 vorgestellten Berechnungs-Algorithmus berechenbar. Mit diesen zusätzlichen Informationen kann die Konturenerkennung die Anzahl der möglichen Strahlerpaare im Bild so weit reduzieren, dass im Idealfall lediglich die wirklichen Strahler verbleiben. Ist dies nicht der Fall, so kann der Bediener über den Touchscreen den Bereich des Bildes auswählen, in welchem sich die gesuchten Strahler befinden. So können direkt zwei Punkte im Bild ermittelt werden, deren Position in Relation zur Kameraposition bekannt sind. In Abbildung 4.1 ist das Kamerasystem, inklusive eines Touchscreens zur Bedienung, an einem Heliostat angebracht, dargestellt. Alle Messungen dieses Systems finden statt, wenn der Heliostat in seine Referenzposition gefahren ist. Am Ende der Messung gibt der verwendete Algorithmus die gemessene Ausrichtung des Heliostaten in Azimuts- und Elevationsrichtung aus.

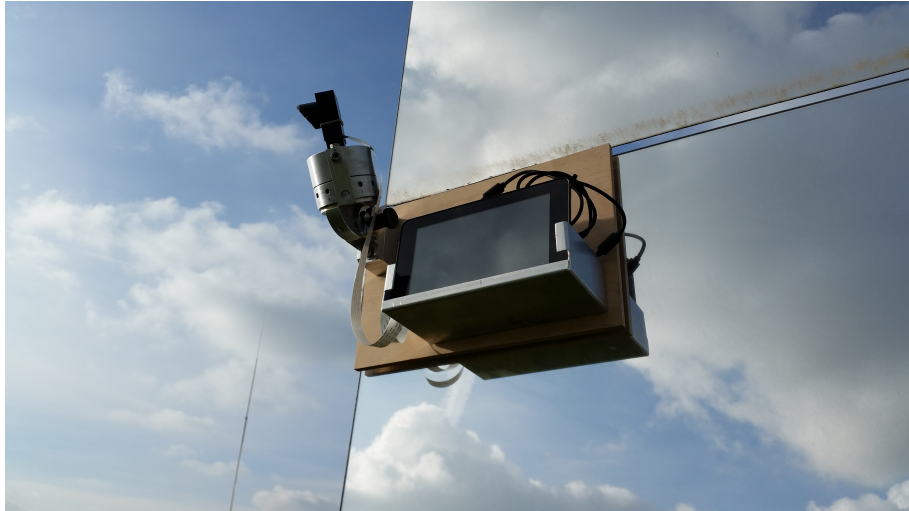


Abbildung 4.1: Foto des umgesetzten und an einem Heliostat angebrachten Systems

### 4.1 Aufbau des Systems

Das System besteht aus einem Einplatinencomputer, in diesem Fall ein Raspberry Pi 3, einer Raspi IR Kamera V2, einem 940nm Filter, einem Touchscreen zur Bedienung des Computers und einer selbst gebauten Halterung, welche das System am Heliostat befestigt. Die Halterung hat eine Größe von 20 mal 30 cm und ist dabei so aufgebaut, dass die Kamera verdrehbar ist, um aus jeder Position im Feld den Turm im Kamerasichtfeld zu haben. Dies wird durch eine drehbare Halterung ermöglicht, welche die Azimutsausrichtung der Kamera durch vorgegebene Rastpositionen in 30° Schritten einstellen kann. Am Turm sind zwei IR-Strahler angebracht, welche dauerhaft strahlen und deren Wellenlänge für die Kamera sichtbar ist.

Das System fotografiert die Umgebung und ein selbstgeschriebenes Programm auf dem Raspberry Pi berechnet mit Hilfe der Bilderkennung und anhand der im Folgenden erklärten Formeln die Kameraausrichtung. Anhand der bekannten Heliostatgeometrie und der Rasteneinstellung lässt sich aus der Kameraausrichtung auf die gesuchte Heliostatausrichtung zurückschließen.

### 4.2 Bestimmung der Kameraposition am Heliostat

Im ersten Schritt dieser Berechnung verwendet der Algorithmus die Elevations- und Azimutswinkel in der idealen Referenzposition des zu untersuchenden Heliostaten. Durch diese Ausgangswinkel ist es möglich, die Position der Kamera genauer zu bestimmen, und auch dem Bediener die Rasteneinstellung der Kamerahalterung vorzugeben. In weiteren Algorithmus-Iterationen beginnt die Berechnung an dieser Stelle erneut, allerdings berechnet der Algorithmus die Kameraposition dann anhand der berechneten Azimuts- und Elevationswinkel der vorherigen Iteration, um die Genauigkeit des Systems zu erhöhen.

Den Heliostaten des Feldes sind gruppenweise verschiedene  $H_{yaw}$  Werte vorgegeben, um die Heliostaten positionsbasiert auf den Turm auszurichten. Der Referenz-Azimutswinkel  $H_{yaw}$  ist gegeben und stellt den Winkel zwischen Norden und der Spiegelnormalen in mathematisch negativer Drehrichtung dar. In Abbildung 5.6 ist dargestellt, welchen  $H_{yaw}$  die einzelnen Heliostaten im Feld aufweisen. Der Azimutswinkel  $\alpha$  des Heliostaten ist im Folgenden als mathematisch positiver Winkel zwischen Spiegeloberfläche und Norden definiert, um einen Azimutswinkel von  $0^\circ$  bei Nord-Süd Ausrichtung der Kamera zu erhalten. Somit gilt die Beziehung  $\alpha = -(H_{yaw} - 90^\circ)$  (siehe Abbildung 4.2).

Um die Ausrichtung der Kamera auf die Strahler zu ermöglichen, berechnet der Algorithmus abhängig von der Heliostatposition und dessen  $H_{yaw}$  die nötige Azimutsverdrechung der Kamera. Dies geschieht, indem der Algorithmus den Azimutswinkel  $\alpha_{Raste}$  der Kamera ermittelt, bei welchem diese genau auf die Strahler ausgerichtet wäre und diesen dem Bediener über den Touchscreen mitteilt. Um den Rastenwinkel zu erhalten, ist von dem idealen Kamerawinkel der Referenz-Azimutswinkel des Heliostaten abzuziehen und abschließend der nächstgelegene Rastenwinkel auszuwählen. Die Verdrehung der Kamera anhand der Raste ist in Abbildung 4.3 verdeutlicht.

Der Referenz-Elevationswinkel  $F_{pitch}$  ist der Winkel in negativer Drehrichtung zwischen der Senkrechten und der Spiegeloberfläche in Referenzposition. Die Definition



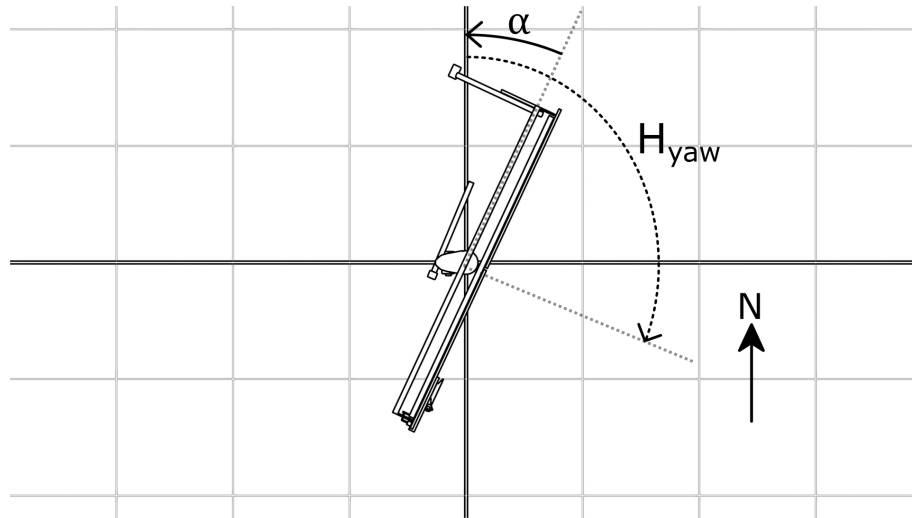


Abbildung 4.2: Zusammenhang von  $H_{yaw}$  und  $\alpha$  eines Heliostaten. Auf dem Turm-Server ist  $H_{yaw}$  entgegen der mathematisch positiven Drehrichtung definiert

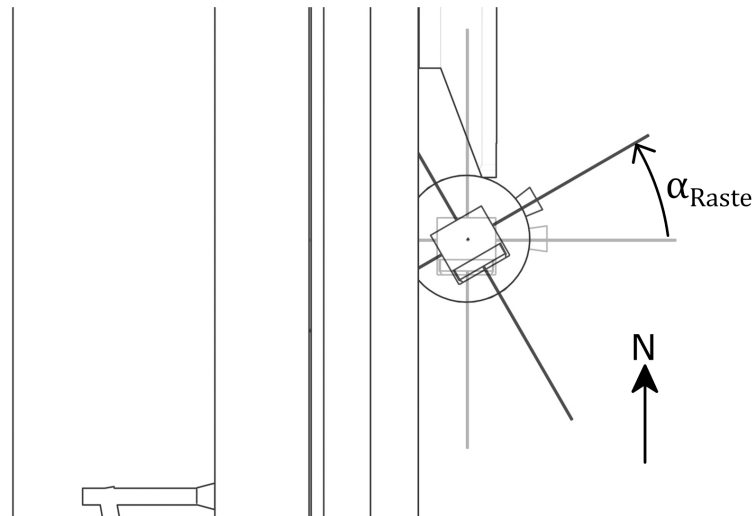


Abbildung 4.3: Kamera KOS von oben, inklusive  $\alpha_{Raste}$ , welches die Verdrehung der Kamera zu ihrer Halterung beschreibt

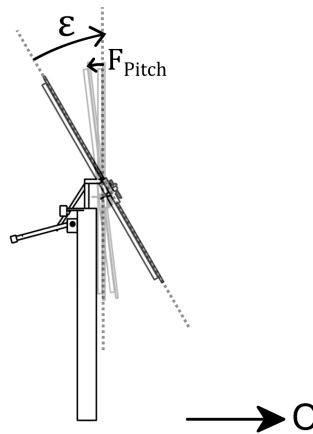


Abbildung 4.4: Darstellung des Elevationswinkels  $\varepsilon$  mit dessen positiver Winkeldefinition.  $F_{Pitch}$  ist im System in negativer Drehrichtung definiert, und ist der negative Elevationswinkel, den der Heliostat in seiner Referenzposition einnimmt

des Elevationswinkels  $\varepsilon$  in verdrehter Position ist der Winkel in positiver Drehrichtung zwischen der Senkrechten und der Spiegeloberfläche in verdrehter Position (siehe Abbildung 4.4).

Um die Berechnung der Winkelbestimmung zu verbessern, ist die Kameraposition auf dem Heliostaten und deren Ausrichtung anhand der bekannten Geometrie der Halterung durch den Algorithmus zu bestimmen. Zunächst ist die Position der Kamera am Heliostaten im Elevationskoordinatensystem des Heliostaten zu bestimmen. Bei diesem liegt der Ursprung in der Mitte der Elevationsachse, wobei die x-Achse in Richtung der Spiegelnormalen zeigt, die y-Achse entlang der Elevationsachse und die z-Achse senkrecht nach oben. Da an dieser Stelle der Azimutswinkel zu Null gesetzt ist, ist die Spiegelnormale noch nach Osten ausgerichtet und die Elevationsachse nach Norden. Das Elevations- und das anschließend verwendete Azimutskoordinatensystem sind in Abbildung 4.5 dargestellt.

Zur Berechnung der Kameraposition sind die in Abbildungen 4.6 und 4.7 dargestellten und in Tabelle 4.1 aufgelisteten Größenangaben der Halterung und des Heliostaten notwendig.

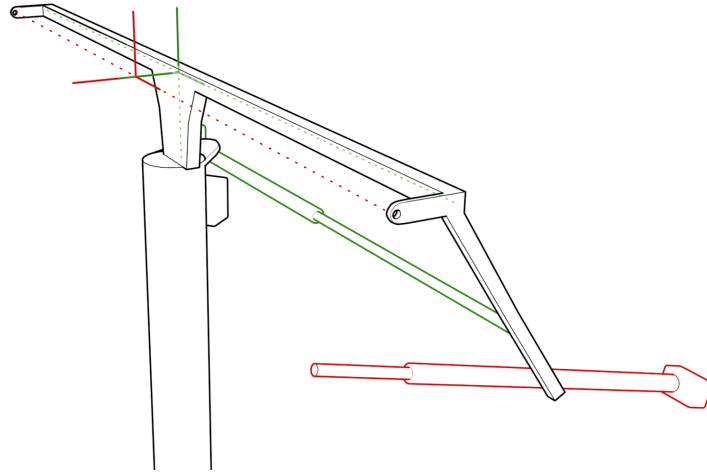


Abbildung 4.5: Darstellung des Elevations- und des Azimuts-Koordinatensystems. Elevation in rot und Azimut in grün.

Längenbezeichnung	Symbol	Größe
Abstand Elevationsachse und Azimutachse	$F_y$	95 mm
Abstand Spiegelfront und Elevationsachse	$G_y$	125 mm
Abstand Spiegelfront und Halterungs-Kippachse	$l_{SK}$	81 mm
Kippungswinkel	$\varepsilon_{Raste}$	30°
Länge nach Kippung	$l_{Kipp}$	80 mm
Höhe Linse	$h_{Linse}$	37 mm
Breite Spiegel	$b_{Spiegel}$	3,4 m
Position auf Halter in y	$l_y$	31 mm
Position auf Halter in z	$l_z$	61 mm
Referenz Azimutswinkel	$H_{yaw}$	variabel
Referenz Elevationswinkel	$F_{Pitch}$	variabel
Rastenwinkel der Kamerahalterung	$\alpha_{Raste}$	variabel

Tabelle 4.1: Längenbezeichnungen und -angaben der Kamerahalterung und des Heliostaten (siehe Abbildungen 4.6 und 4.7)

Die Position der Kamera im Elevations-KOS  $\vec{x}_{Kamera,ElevationsKOS}$  ergibt sich anhand der Gleichungen:

$$x_{Kamera,ElevationsKOS} = G_y + l_{SK} + \sin(\varepsilon_{Raste}) * (l_{Kipp} + h_{Linse}) \quad (4.1)$$

#### 4. Aufbau des ausgewählten Systems

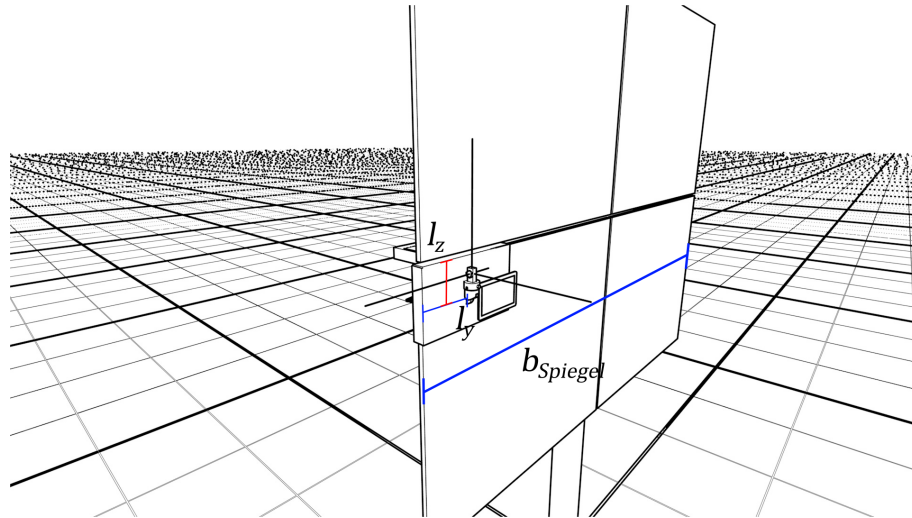


Abbildung 4.6: Abmessungen  $l_y$ ,  $l_z$  und  $b_{Spiegel}$  des Heliostaten

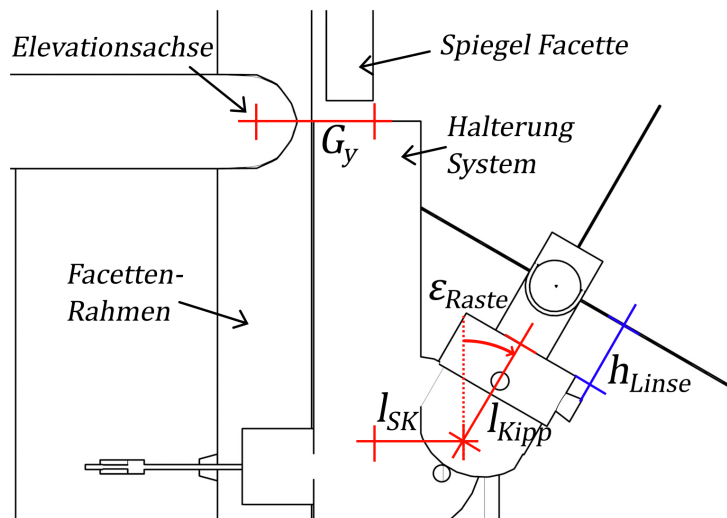


Abbildung 4.7: Abmessung  $G_y$  des Heliostaten und  $l_{SK}$ ,  $\epsilon_{Raste}$ ,  $l_{Kipp}$ ,  $h_{Linse}$  der Halterung

$$y_{Kamera,ElevationsKOS} = -b_{Spiegel}/2 + l_y \quad (4.2)$$

$$z_{Kamera,ElevationsKOS} = -l_z + \cos(\varepsilon_{Raste}) * (l_{Kipp} + h_{Linse}) \quad (4.3)$$

Nun ist die Position berechenbar, welche die Kamera im Elevations-KOS nach einer Drehung um  $F_{Pitch}$  hat. Dafür wird zunächst der Abstand der Kamera zum Elevations-KOS Ursprung  $a_{EleKos,Kamera}$  benötigt, um die Drehung um diesen Punkt auf die x und z Achse aufteilen zu können. Der Winkel  $\varepsilon_0$  ist der Winkel zwischen der horizontalen Ebene und der Kameraposition um den Ursprung des Elevations-KOS. Wird der Heliostat um den Winkel  $F_{Pitch}$  gedreht, so ergibt sich der neue Winkel  $\varepsilon_1$  durch

$$a_{EleKos,Kamera} = \sqrt{(x_{Kamera,ElevationsKOS})^2 + (z_{Kamera,ElevationsKOS})^2} \quad (4.4)$$

$$\varepsilon_0 = -\arctan(z_{Kamera,ElevationsKOS}/x_{Kamera,ElevationsKOS}) \quad (4.5)$$

$$\varepsilon_1 = \varepsilon_0 - F_{Pitch}. \quad (4.6)$$

Dadurch ergibt sich die neue Position der Kamera  $\vec{x}_\varepsilon$  zu

$$\begin{pmatrix} x_\varepsilon \\ y_\varepsilon \\ z_\varepsilon \end{pmatrix} = \begin{pmatrix} \cos(\varepsilon_1) * a_{EleKos,Kamera} \\ 1 * y_{Kamera,ElevationsKOS} \\ \sin(\varepsilon_1) * a_{EleKos,Kamera} \end{pmatrix} \quad (4.7)$$

An dieser Stelle wird das Elevations- in das Azimuts-KOS, welches im Drehpunkt der Azimuts-Achse liegt, überführt.

$$\begin{pmatrix} x_{Azimut-KOS} \\ y_{Azimut-KOS} \\ z_{Azimut-KOS} \end{pmatrix} = \begin{pmatrix} x_\varepsilon \\ y_\varepsilon \\ z_\varepsilon \end{pmatrix} + \begin{pmatrix} F_y \\ 0 \\ 0 \end{pmatrix} \quad (4.8)$$

Nun wird mit der selben Vorgehensweise in Azimutsrichtung um den Winkel  $-(H_{yaw} - 90^\circ)$  gedreht.

$$a_{AziKOS,Kamera} = \sqrt{(x_{Azimut-KOS})^2 + (y_{Azimut-KOS})^2} \quad (4.9)$$

$$\alpha_0 = \arctan(y_{Azimut-KOS}/x_{Azimut-KOS}) \quad (4.10)$$

$$\alpha_1 = \alpha_0 - (H_{yaw} - 90^\circ) \quad (4.11)$$

Dadurch ergibt sich die neue Position der Kamera zu

$$\begin{pmatrix} x_\alpha \\ y_\alpha \\ z_\alpha \end{pmatrix} = \begin{pmatrix} \cos(\alpha_1) & * & a_{AziKOS,Kamera} \\ \sin(\alpha_1) & * & a_{AziKOS,Kamera} \\ 1 & * & z_{Azimut-KOS} \end{pmatrix} \quad (4.12)$$

Somit ist die Position der Kamera im Turm-KOS gegeben durch die Position des Azimuts-KOS  $\overrightarrow{x_{PositionAziKOS}}$ , welche der zuvor verwendeten Position des Heliostat-Pylons entspricht,

$$\begin{pmatrix} x_{Kamera,Turm-KOS} \\ y_{Kamera,Turm-KOS} \\ z_{Kamera,Turm-KOS} \end{pmatrix} = \begin{pmatrix} x_{PositionAziKOS} \\ y_{PositionAziKOS} \\ z_{PositionAziKOS} \end{pmatrix} + \begin{pmatrix} x_\alpha \\ y_\alpha \\ z_\alpha \end{pmatrix}. \quad (4.13)$$

### 4.3 Bestimmung der Kamera y-Achse und Verdrehung anhand des Bildes

Wie in Abbildung 4.8 zu sehen ist, ändert sich aus unterschiedlichen Heliostatpositionen, auch im Falle der horizontalen Ausrichtung der Kamera, der Winkel zwischen den Strahlern und der Bild x-Achse. Aus diesem Grund ist  $\lambda_{StrahlerHorizontaleKamera}$  für den Fall der horizontalen Kameraausrichtung zu bestimmen, um eventuelle Verdrehungen des Bildes bestimmen zu können. Die Bestimmung der Strahlerpositionen im horizontalen Vergleichsbild geschieht dabei anhand der in Abschnitt 3.1.1 herge-

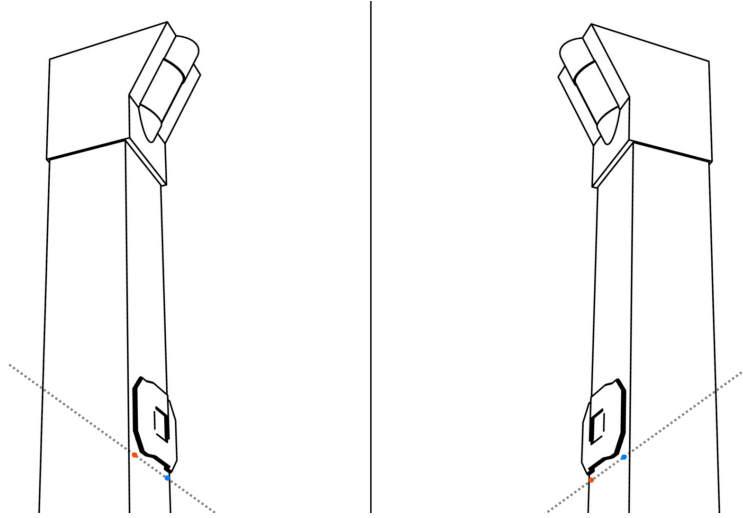


Abbildung 4.8: Links der Turm aus östlicher Ansicht, rechts aus dem Westen des Feldes

leiteten Formeln, wobei nicht die Target-Mitte als Bildmittelpunkt zu verwenden ist, sondern die Bildmitte der im Referenzpunkt des Heliostaten stehenden Kamera.

Die Berechnung der einzelnen  $\lambda$  Winkel geschieht dabei wie Folgt:

$$\lambda_i = \arctan \left( \frac{-(z_{Pixel,Westen,i} - z_{Pixel,Osten,i})}{x_{Pixel,Westen,i} - x_{Pixel,Osten,i}} \right) \quad (4.14)$$

Aus der Differenz zwischen den Winkeln aus gemessenen und idealen Strahlerpositionen, wie in Abbildung 4.9 dargestellt, lässt sich somit die Bildverdrehung ermitteln.

$$\lambda = \lambda_{StrahlerGemessen} - \lambda_{StrahlerHorizontaleKamera} \quad (4.15)$$

Für Verdrehungen des Bildes entgegen des Uhrzeigersinns ist  $\lambda$  negativ. Dies ergibt sich daraus, dass eine Drehung im Bild gegen den Uhrzeigersinn einer Drehung der Kamera im Uhrzeigersinn entspricht.

Um anhand der Strahlerpositionen im Bild auf die Ausrichtung der Kamera zu kommen, legt der Algorithmus die Ausrichtung des Kamera-KOSs im Raum fest. Zusammengefasst sind die folgenden Schritte zur Bestimmung der y-Achse der Kamera notwendig:

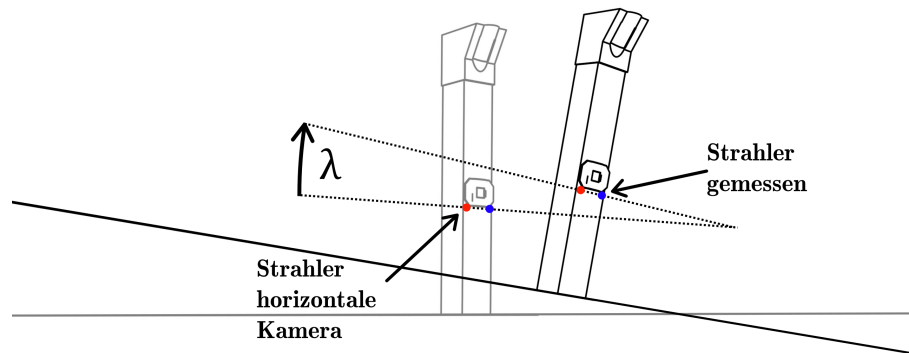


Abbildung 4.9: Verdrehung des Bildes. Orange dargestellt ist der östlich angebrachte Strahler, blau der westliche

1. Festlegen der Kugeloberfläche, auf welcher der Bildmittelpunkt der projizierten Bildebene liegen wird, siehe Abbildungen 4.10 und 4.11
2. Bestimmen der Pixel Positionen der Strahler im Bild und deren Abstand zur Bildmitte
3. Bestimmen der Durchstoßpunkte der Bildebene anhand der Geraden zwischen Strahlern und Kamera
4. Aufstellen der Kugelgleichungen um beide Durchstoßpunkte und um die Kamera
5. Bestimmen der zwei möglichen Schnittpunkte der drei Kugeln
6. Auswählen des korrekten Schnittpunktes und Bestimmung der resultierenden y-Achse.

Im Detail erfolgt die Berechnung wie folgt:

Im ersten Schritt stellt der Algorithmus eine Kugelgleichung mit dem Radius von einem Meter und dem Mittelpunkt in der Kameraposition auf. Auf diese wird die Bildebene projiziert, anhand derer die Ausrichtung festgelegt werden kann. Diese



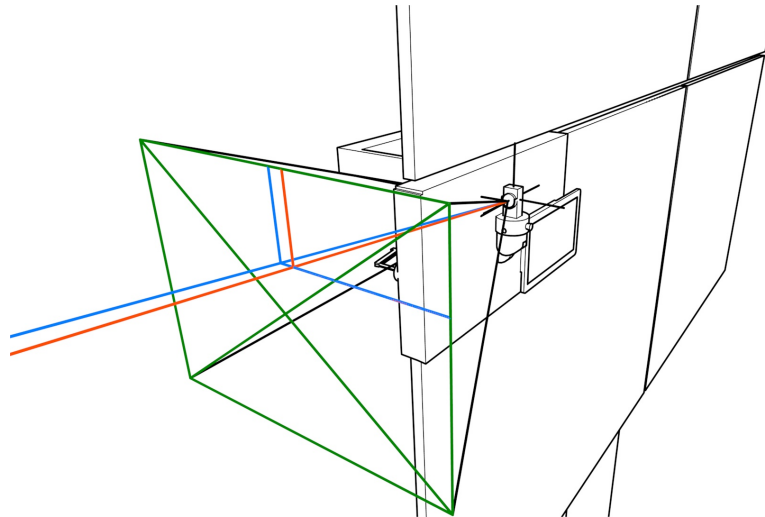


Abbildung 4.10: Projektion der Bildebene. Orange und blau dargestellt sind jeweils die verbindenden Vektoren zwischen den Strahlern und der Kamera. Grün dargestellt ist die projizierte Bildebene

Bildebene, inklusive der verbindenden Vektoren zwischen Strahlern und Kamera, ist in Abbildung 4.10 dargestellt. Der Mittelpunkt der projizierten Bildebene liegt auf der Kugeloberfläche und wird anhand der Strahlerpositionen festgelegt (siehe Abbildung 4.11)

$$(x - x_{Kamera})^2 + (y - y_{Kamera})^2 + (z - z_{Kamera})^2 - 1 m^2 = 0 \quad (4.16)$$

Im zweiten Schritt ermittelt der Algorithmus den Abstand zwischen den Pixel-Positionen der Strahler und der Mitte des gemachten Fotos (siehe Abbildung 4.12). Diese werden mit Hilfe des Faktors  $mpx$ , welcher die Breite eines Pixels in Metern in der virtuellen Bildebene darstellt, in Meter umgerechnet. Für diese Berechnun-

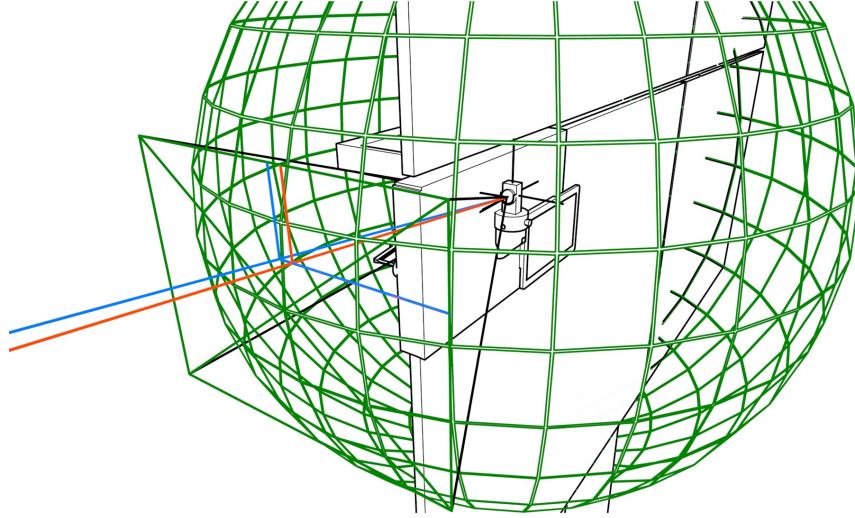


Abbildung 4.11: Projektion der Bildebene und Kugel um Kamera. Die Bildebene ist definitionsgemäß immer tangential zur Kugeloberfläche, wobei der Bildmittelpunkt der berührende Punkt ist

gen werden die Pixel-Breite des Bildes  $b_{Bild}$  und die Pixel-Höhe  $h_{Bild}$  benötigt. Der Abstand zwischen den östlichen Strahlerpixeln und der Bildmitte ergibt sich durch

$$mpx = \tan(\angle FOV/2) * \frac{1m}{b_{Bild}/2} \quad (4.17)$$

$$\Delta x_{ost} = (x_{PixelStrahlerOst} - b_{Bild}/2) * mpx \quad (4.18)$$

$$\Delta z_{ost} = (z_{PixelStrahlerOst} - h_{Bild}/2) * mpx \quad (4.19)$$

$$r_{Ost} = \sqrt{(\Delta x_{ost})^2 + (\Delta z_{ost})^2}. \quad (4.20)$$

Im dritten Schritt wird anhand der Pixelabstände zwischen den Strahlern und der Bildmitte die Position der *DSP* (Durchstoßpunkte) im Raum ermittelt. In diesen *DSP* schneiden sich die virtuelle Bildebene und die Geraden, welche die Strahler und die Kamera verbinden (siehe Abbildung 4.13). Zur Positionsbestimmung der *DSP* wird zunächst der Abstand dieser zur Kamera benötigt. Dies geschieht für den östlichen Punkt anhand der Gleichung

$$a_{OstKamera} = \sqrt{r_{Ost}^2 + a_{KameraBm}^2} = \sqrt{r_{Ost}^2 + 1}. \quad (4.21)$$

#### 4. Aufbau des ausgewählten Systems

---

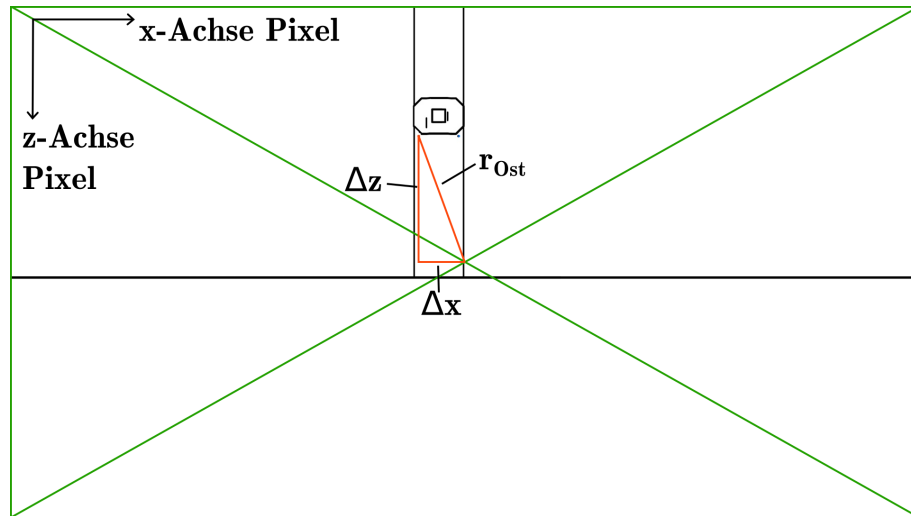


Abbildung 4.12: Bestimmung der Pixel-Abstände anhand des Fotos. Die Ausrichtung der Pixel-Achsen ergibt sich aus der durch OpenCV vorgegebenen Pixel-Achsen

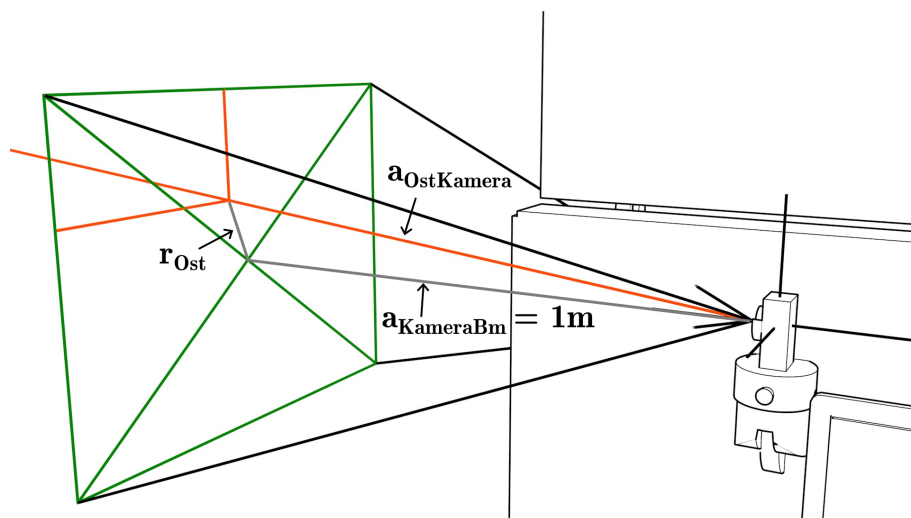


Abbildung 4.13: Abstände zwischen östlichem projiziertem Punkt und Kamera

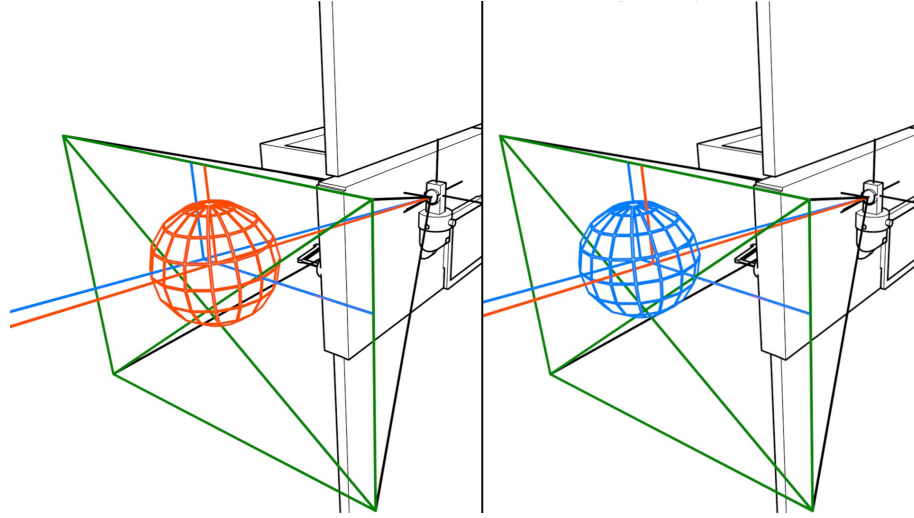


Abbildung 4.14: Links die Kugel um den östlichen Durchstoßpunkt, rechts um den westlichen

Anhand der Geradengleichungen, welche die Kamera und die Strahler verbinden, kann zusammen mit dem Abstand der *DSP* und der Kamera auf die Position der *DSP* im Raum geschlossen werden. Für den östlichen Strahler geschieht dies durch die Gleichung

$$\begin{pmatrix} x_{DSP,Ost} \\ y_{DSP,Ost} \\ z_{DSP,Ost} \end{pmatrix} = \begin{pmatrix} x_{Kamera} \\ y_{Kamera} \\ z_{Kamera} \end{pmatrix} + \frac{a_{OstKamera}}{|\vec{x}_{Strahler,Ost} - \vec{x}_{Kamera}|} * \begin{pmatrix} x_{Strahler,Ost} - x_{Kamera} \\ y_{Strahler,Ost} - y_{Kamera} \\ z_{Strahler,Ost} - z_{Kamera} \end{pmatrix}. \quad (4.22)$$

Die vorangegangenen Berechnungen werden sowohl für den östlichen als auch für den westlichen *DSP* durchgeführt.

Im vierten Schritt stellt der Algorithmus jeweils eine Kugelgleichung für jeden der *DSP* auf, mit dem Abstand zum Bildmittelpunkt als Radius und der Position des *DSP* als Mittelpunkt (siehe Abbildung 4.14).

$$(x - x_{DSP,Ost})^2 + (y - y_{DSP,Ost})^2 + (z - z_{DSP,Ost})^2 - r_{Ost}^2 = 0 \quad (4.23)$$

$$(x - x_{DSP,West})^2 + (y - y_{DSP,West})^2 + (z - z_{DSP,West})^2 - r_{West}^2 = 0 \quad (4.24)$$

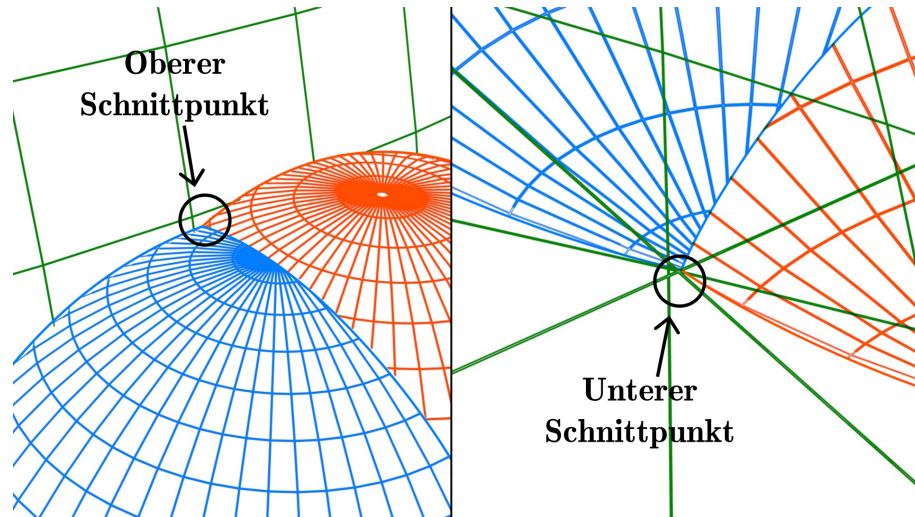


Abbildung 4.15: Links der obere Schnittpunkt der 3 Kugeln, rechts der untere, welcher in diesem Fall dem Bildmittelpunkt entspricht

Im fünften Schritt sucht ein Solver die Schnittpunkte zwischen den drei aufgestellten Kugelgleichungen. Diese weisen zwei Lösungen auf, da zwei sich schneidende Kugeln einen Schnittkreis ergeben und eine dritte schneidende Kugel den Schnittkreis in zwei Punkten schneidet. Die Positionen der zwei möglichen Schnittpunkte sind in Abbildung 4.15 dargestellt. Um herauszufinden, welche Lösung die Richtige ist, ist für beide Punkte die resultierende  $y$ -Achse der Kamera anhand der Vorgehensweise aus dem sechsten Schritt zu berechnen. Diese wird anschließend in eine Methode eingesetzt, welche die Pixelpositionen der Strahler im Bild dieser  $y$ -Achse berechnet. Diese entspricht in ihrer Funktionsweise der in Abschnitt 5.2.2 beschriebenen Methode. Diejenige  $y$ -Achse, deren Strahler-Pixel die geringere Abweichung zu den gemessenen aufzeigen, ist die  $y$ -Achse der Kamera.

Im sechsten Schritt wird die  $y$ -Achse der Kamera dadurch festgelegt, dass diese als Einheitsvektor von dem Bildmittelpunkt aus auf die Kameraposition zeigt. Der Ursprung der resultierenden  $y$ -Achse liegt in einem Hilfs-Koordinatensystem, dessen Ursprung in der Kamera liegt und dessen Achsen parallel zu denen des Turm-KOS

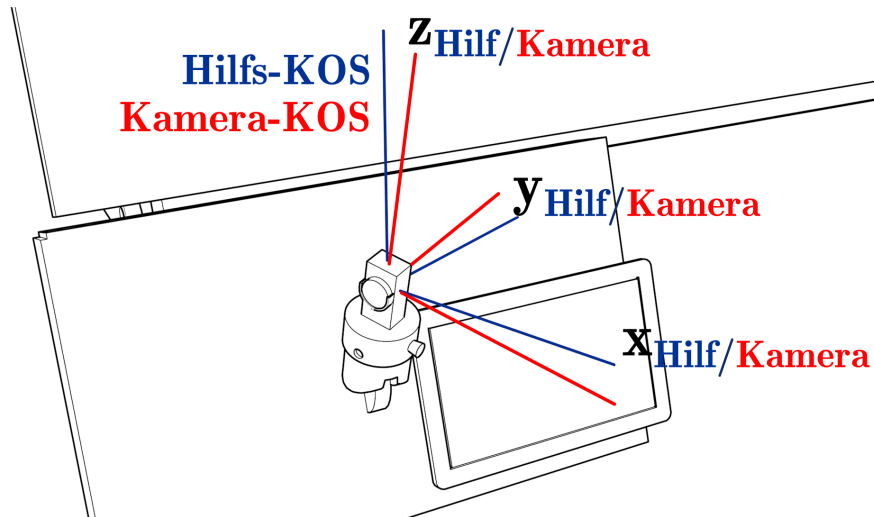


Abbildung 4.16: Das Kamera-Koordinatensystem ist in rot dargestellt und das Hilfs-KOS, dessen Ursprung in der Kamera liegt und dessen Achsen parallel zu denen des Turm-KOS verlaufen, ist in blau dargestellt

verlaufen. Sowohl das Kamera-KOS als auch das Hilfs-KOS sind in Abbildung 4.16 dargestellt. Somit ergibt sich  $\vec{y}$  zu

$$\vec{y} = \begin{pmatrix} x_{Kamera} - x_{Schnittpunkt} \\ y_{Kamera} - y_{Schnittpunkt} \\ z_{Kamera} - z_{Schnittpunkt} \end{pmatrix}. \quad (4.25)$$

Die x- und z-Achsen des Kamera-KOS werden im folgenden Kapitel berechnet.

## 4.4 Bestimmung der Kamera x- und z-Achsen anhand der bekannten Kamera y-Achse und Verdrehung

Im Kamera-KOS zeigt die y-Achse vom Turm weg, also aus der Bildebene der Kamera heraus, die x-Achse zeigt in die Bildbreite, und die z-Achse in die Bildhöhe. Mit Hilfe der Verdrehung des Bildes sind an dieser Stelle die x- und z-Achsen der Kamera berechenbar.

Um die x- und z-Achsen berechnen zu können, ist eine Ebenengleichung aufzustellen, welche durch den Ursprung des Hilfs-KOS geht, und deren Normalenvektor die y-Achse ist (siehe Abbildung 4.17). Diese bildet die x-z-Ebene des Kamera-KOS. Die Schnittgerade dieser Ebene und der x-y-Ebene des Hilfs-KOS wird anschließend um den Verdrehungswinkel  $-\lambda$  um die y-Achse gedreht, um die Kamera x-Achse zu erhalten. Eine Drehung um  $-\lambda - 90^\circ$  ergibt die z-Achse. Das Hilfs-KOS entspricht in seiner Achsausrichtung dem Turm-KOS, wobei der Ursprung in der Kamera liegt, um die Berechnung der Heliostatverdrehung anhand des Kamera-KOS zu vereinfachen. Dies ist anhand von Abbildung 4.16 ersichtlich. Die folgenden Berechnungen finden im Hilfs-KOS statt.

Die Ebenengleichung der x-z-Ebene des Kamera-KOS im Hilfs-KOS ergibt sich anhand der Kamera-y-Achse  $\vec{y}$  durch

$$\begin{pmatrix} x_{HilfsKOS} \\ y_{HilfsKOS} \\ z_{HilfsKOS} \end{pmatrix} * \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = 0 \quad (4.26)$$

und die Ebenengleichung der x-y-Ebene des Hilfs-KOS ist

$$\begin{pmatrix} x_{HilfsKOS} \\ y_{HilfsKOS} \\ z_{HilfsKOS} \end{pmatrix} * \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0. \quad (4.27)$$

Werden diese beiden Gleichungen gleichgesetzt und umgeformt, ergibt sich so die Geradengleichung der Schnittgeraden in der x-y-Ebene des Hilfs-KOS zu

$$y_{HilfsKOS} = -\frac{y_1}{y_2} * x_{HilfsKOS}. \quad (4.28)$$

Der normierte Vektor der Schnittgeraden  $\vec{s}_{Hilf}$  im Raum ergibt sich somit zu

$$\vec{s}_{Hilf} = \begin{pmatrix} \frac{1}{\sqrt{1+(y_1/y_2)^2}} \\ \frac{-y_1/y_2}{\sqrt{1+(y_1/y_2)^2}} \\ 0 \end{pmatrix}. \quad (4.29)$$

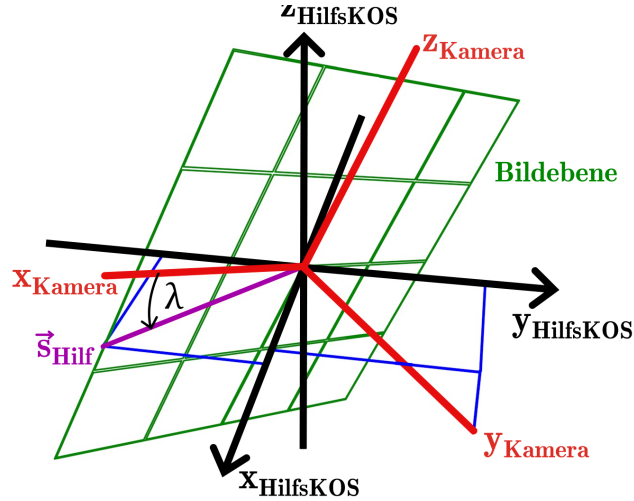


Abbildung 4.17: Visualisierung der Schnittgeraden  $\vec{s}_{Hilf}$  der Hilfs x-y-Ebene und der Kamera x-z-Ebene

Die x-Achse ergibt sich durch die Drehung von  $\vec{s}_{Hilf}$  um die y-Achse der Kamera um den Winkel  $-\lambda$ . Zur Realisierung der Drehung eines Vektors um einen beliebigen Drehachsenvektor, in diesem Fall die y-Achse des Kamera-KOS, ist das Koordinatensystem zunächst so zu drehen, dass die Drehachse in der z-Achse liegt. Anschließend erfolgt die gewünschte Drehung um die z-Achse, gefolgt von der ersten Drehung in umgekehrter Richtung und Reihenfolge, sodass die Drehachse wieder in ihre Ausgangsposition gelangt, während alle anderen Vektoren sich um diese herum gedreht haben.

Im ersten Schritt ist das Koordinatensystem so um die x-Achse zu drehen, dass die Drehachse im Anschluss in der y-z-Ebene liegt. Diese ergibt sich anhand der Drehmatrix

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{y_3}{\sqrt{y_2^2 + y_3^2}} & -\frac{y_2}{\sqrt{y_2^2 + y_3^2}} \\ 0 & \frac{y_2}{\sqrt{y_2^2 + y_3^2}} & \frac{y_3}{\sqrt{y_2^2 + y_3^2}} \end{pmatrix} \quad (4.30)$$

Im zweiten Schritt erfolgt die Drehung, welche die Drehachse in die z-Achse legt, durch

$$R_y = \begin{pmatrix} \sqrt{y_2^2 + y_3^2} & 0 & -y_1 \\ 0 & 1 & 0 \\ y_1 & 0 & \sqrt{y_2^2 + y_3^2} \end{pmatrix} \quad (4.31)$$



Nun findet die Drehung um den gewünschten Winkel statt. In diesem Fall um  $-\lambda$ , mit Hilfe der Drehmatrix

$$R_z = \begin{pmatrix} \cos(\lambda) & -\sin(\lambda) & 0 \\ \sin(\lambda) & \cos(\lambda) & 0 \\ 0 & 0 & 1. \end{pmatrix} \quad (4.32)$$

Die gesamte Drehmatrix, welche die Drehung eines Vektors um den Winkel  $-\lambda$  um die y-Achse der Kamera durchführt, ergibt sich somit zu:

$$R_{ges} = R_x^{-1} \cdot R_y^{-1} \cdot R_z \cdot R_y \cdot R_x \quad (4.33)$$

Mit den Winkeln  $-\lambda$  und  $-\lambda-90^\circ$  können somit die x- und z-Achsen aus der Schnittgeraden  $\vec{s}_{Hilf}$  bestimmt werden. Mit dem somit vollkommen bestimmten Kamera-KOS ist anhand der im folgenden Kapitel erläuterten Vorgehensweise die Heliostausrichtung zu bestimmen.

## 4.5 Bestimmung der Winkel aus gemessenen Achsen

Der Azimuts- und Elevationswinkel des Heliostaten sind anhand des Kamera-KOS bestimmbar. Dies ergibt sich daraus, dass das Hilfs-KOS durch einfache Drehmatrizen, welche sich aus der Verdrehung der Kamera und des Heliostaten zusammensetzen, in das Kamera-KOS umgeformt werden kann. Durch die bekannte Form der Matrizen kann somit direkt vom Kamera-KOS auf die Ausrichtung des Heliostaten geschlossen werden. Im Detail ergibt sich die gesamte Drehmatrix wie folgt:

1. Drehung der Koordinatenachsen des Hilfs-KOS um dessen z-Achse um  $\alpha_{Raste}$
2. Drehung des neuen KOS um die y-Achse des Hilfs-KOS um  $-F_{Pitch} + \varepsilon_{Raste}$
3. Drehung des neuen KOS um die z-Achse des Hilfs-KOS um  $-(H_{yaw} - 90^\circ)$ , um das Kamera-KOS zu erhalten.

Die zugehörigen Drehmatrizen ergibt sich mit  $\varepsilon = -F_{Pitch} + \varepsilon_{Raste}$  und  $\alpha_H = -(H_{yaw} - 90^\circ)$  dementsprechend zu

$$R_{Raste} = \begin{pmatrix} \cos(\alpha_{Raste}) & -\sin(\alpha_{Raste}) & 0 \\ \sin(\alpha_{Raste}) & \cos(\alpha_{Raste}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.34)$$

$$R_{Elevation} = \begin{pmatrix} \cos(\varepsilon) & 0 & \sin(\varepsilon) \\ 0 & 1 & 0 \\ -\sin(\varepsilon) & 0 & \cos(\varepsilon) \end{pmatrix} \quad (4.35)$$

$$R_{Azimut} = \begin{pmatrix} \cos(\alpha_H) & -\sin(\alpha_H) & 0 \\ \sin(\alpha_H) & \cos(\alpha_H) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.36)$$

$$R_{ges} = R_{Azimut} * R_{Elevation} * R_{Raste} \quad (4.37)$$

Die Achsen des Kamera-KOS im Hilfs-KOS sind direkt aus der Drehmatrix ablesbar. Die erste Spalte von  $R_{ges}$  entspricht der x-, die zweite der y- und die dritte der z-Achse des Kamera-KOS.

Aus Gleichung 4.37 ergeben sich die Achsen des Kamera-KOS in Abhängigkeit von  $\alpha_H$ ,  $\varepsilon$  und  $\alpha_{Raste}$  zu

$$\vec{x}_{Kamera} = \begin{pmatrix} \cos(\alpha_H) \cos(\varepsilon) \cos(\alpha_{Raste}) - \sin(\alpha_H) \sin(\alpha_{Raste}) \\ \sin(\alpha_H) \cos(\varepsilon) \cos(\alpha_{Raste}) + \cos(\alpha_H) \sin(\alpha_{Raste}) \\ -\sin(\varepsilon) \cos(\alpha_{Raste}) \end{pmatrix}. \quad (4.38)$$

Für  $\alpha_{Raste} = 0$  Vereinfacht sich dies zu

$$\vec{x}_{Kamera} = \begin{pmatrix} \cos(\alpha_H) \cos(\varepsilon) \\ \sin(\alpha_H) \cos(\varepsilon) \\ -\sin(\varepsilon) \end{pmatrix}. \quad (4.39)$$

Ebenso ergibt sich die y-Achse zu

$$\vec{y}_{Kamera} = \begin{pmatrix} -\cos(\alpha_H) \cos(\varepsilon) \sin(\alpha_{Raste}) - \sin(\alpha_H) \cos(\alpha_{Raste}) \\ -\sin(\alpha_H) \cos(\varepsilon) \sin(\alpha_{Raste}) + \cos(\alpha_H) \cos(\alpha_{Raste}) \\ \sin(\varepsilon) \sin(\alpha_{Raste}) \end{pmatrix}, \quad (4.40)$$

welches sich im Falle von  $\alpha_{Raste} = 0$  zu

$$\vec{y}_{Kamera} = \begin{pmatrix} -\sin(\alpha_H) \\ \cos(\alpha_H) \\ 0 \end{pmatrix} \quad (4.41)$$

vereinfacht. Die z-Achse enthält generell keine Abhängigkeit von  $\alpha_{Raste}$ , da dieser Winkel um die z-Achse dreht. Die z-Achse ergibt sich somit zu

$$\vec{z}_{Kamera} = \begin{pmatrix} \cos(\alpha_H) \sin(\varepsilon) \\ \sin(\alpha_H) \sin(\varepsilon) \\ \cos(\varepsilon) \end{pmatrix}. \quad (4.42)$$

Der aus den Achsen bestimmte  $\alpha_H$ -Winkel entspricht der Azimutsverdrehung des Heliostaten, während vom  $\varepsilon$ -Winkel noch  $\varepsilon_{Raste}$  abgezogen werden muss, um die Elevationsverdrehung zu erhalten.

Die Bildverdrehung wurde bereits anhand des gemachten Bildes und eines berechneten horizontalen Vergleichsbildes ermittelt. Anhand des bekannten Kamera-KOS, kann die Bildverdrehung allerdings auch bestimmt werden. Diese Berechnung ist in der Regel nicht notwendig, da die Bildverdrehung zur Berechnung des Kamera-KOS bereits vorhanden sein muss. Allerdings kann diese durchgeführt werden, um Rechenfehler ausschließen zu können. Dazu ist der Winkel zwischen der x-Achse der Kamera und der Schnittgeraden, welche sich anhand der y-Achse durch Gleichung 4.29 ergibt, zu berechnen. Dieser Winkel ergibt sich durch:

$$\vec{x}_{Kamera} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \vec{y}_{Kamera} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \quad (4.43)$$

$$\lambda = \arccos \left( \frac{\vec{x}_{\text{Kamera}} \times \vec{s}}{|\vec{x}| * |\vec{s}|} \right) \quad (4.44)$$

$$\lambda = \arccos \left( x_1 * \frac{1}{\sqrt{1 + (y_1/y_2)^2}} + x_2 * \frac{-y_1/y_2}{\sqrt{1 + (y_1/y_2)^2}} + x_3 * 0 \right). \quad (4.45)$$

Eine Darstellung des Ablaufes dieser Berechnungen im endgültigen System erfolgt im Anhang in Abbildung A.16. Im folgenden Kapitel wird untersucht, welche Fehlerquellen während der Durchführung dieser Messung und Berechnung zu Ungenauigkeiten führen können.

---

## 5 Bewertung des Algorithmus

Die Fehlerquellen des in dieser Arbeit erstellten Systems lassen sich in zwei Kategorien einteilen. Zunächst können Fehler in der Montage und der Positionierung der Kamera auftreten. Diese gehören zur Kategorie der Hardwarefehler. Des Weiteren entstehen Fehler in der Bilderkennung und Winkelbestimmung, welche in die Kategorie der Softwarefehler fallen. Im Laufe dieses Kapitels werden diese möglichen Fehlerquellen bewertet und eine Fehlerabschätzung durchgeführt. Daraus ergibt sich eine Formel, welche eine Einordnung der einzelnen Einflussgrößen der Fehler im Bezug auf die Gesamtgenauigkeit ermöglicht. Abschließend werden die Herangehensweisen und Ergebnisse der Versuchsreihen erläutert, mit denen das System unter realen Bedingungen getestet wurde.

### 5.1 Hardwarefehler

Fehler, welche im untersuchten System bereits vor der Bildverarbeitung auftreten, gehören zur Kategorie der Hardwarefehler und führen zu Ungenauigkeiten in der Winkelbestimmung. Im Folgenden Abschnitt erfolgt eine genauere Betrachtung dieser Einflüsse deren Auswirkungen auf die gemessenen Winkel.

### 5.1.1 Verzerrung des Bildes durch die Kameralinse

Aufgrund optischer Verzerrungen durch die Kameralinse und den davor gesetzten Filter, ist das gemachte Foto keine realistische Darstellung der Umgebung. Um diese Verzerrungen auf dem zu untersuchenden Foto zu eliminieren, können anhand der Kamerakalibrierungsmethode von OpenCV (siehe Mordvintsev and K., 2013a) die optischen Eigenschaften der Kamera untersucht werden. Sind diese bekannt, kann der Algorithmus das gemachte Foto entzerren. In Abbildung 5.1 ist das verzerrte Kalibrierungsmuster dargestellt. Um die Größe und Position der Verzerrungen zu verdeutlichen, sind die Kanten des entzerrten Schachmusters, wie es in Abbildung 5.2 dargestellt ist, überlagert. Die in diesem Fall maximal auftretende Bildverzerrung beträgt 24 Pixel Abweichung zwischen der ver- und entzerrten Pixelposition im Bild. Dies entspricht einer maximalen Abweichung der aus den ver- und entzerrten Bildern bestimmten Winkeln von  $0,48^\circ$ . In Gleichung 5.1 ist verdeutlicht, dass die Verzerrung direkt Proportional in die Abweichung des Ergebnisses mit geht.

$$F_{\angle Verzerrung} \propto \Delta \angle Verzerrung \leq 0,48^\circ = 8,4 \text{ mrad} \quad (5.1)$$

### 5.1.2 Positionierung der Kamera auf der Halterung

Die Kamera kann sowohl in ihrer Position als auch Ausrichtung falsch auf der Halterung angebracht sein. Die Ausrichtung geht dabei direkt proportional in die Abweichung des Ergebnisses mit ein, wie in Gleichung 5.2 zu sehen ist. Abweichungen von mehreren Grad sind dadurch bei falscher Befestigung der Kamera auf der Halterung möglich.

$$F_{\angle Kamera} \propto \Delta \angle Kamera \quad (5.2)$$

Die Auswirkung der Kameraposition auf das Ergebnis ist abhängig von der Entfernung des Heliostaten zu den Strahlern. Die größtmögliche Auswirkung durch die Positionsabweichung tritt bei dem Heliostaten mit dem geringsten Abstand zu den Strahlern auf, da der Abstand invers proportional in die Gleichung geht. Dies ist

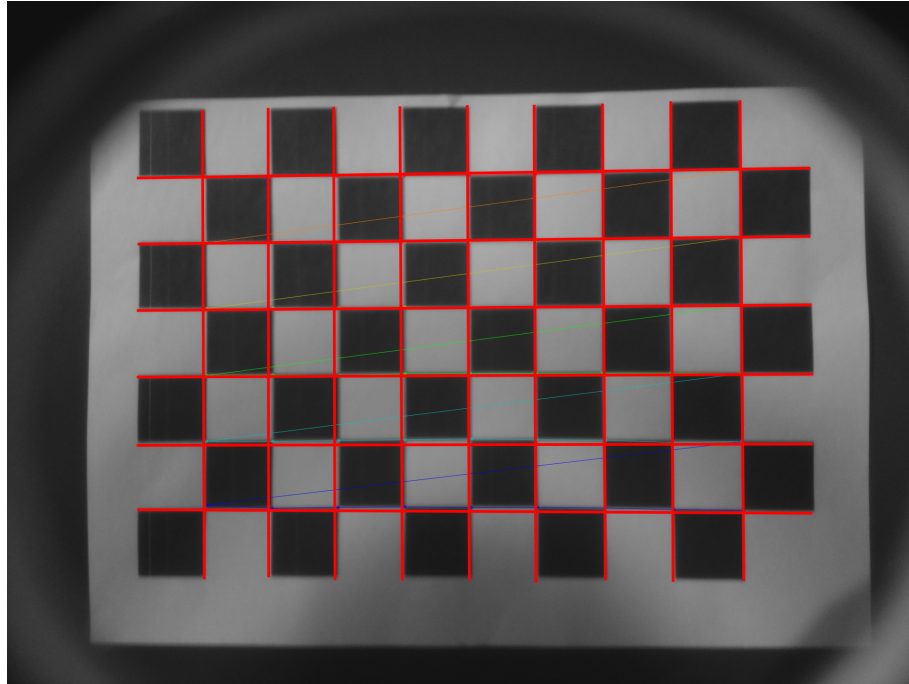


Abbildung 5.1: Originale Aufnahme des Kalibrierungs-Schachfeldes. In rot überlagert sind die Kanten des entzerrten Schachbrettes zur Verdeutlichung der Verzerrung

in diesem Fall der Heliostat AA36, mit einem Abstand zwischen Kamera und Strahlern von 36 m. Dieser Abstand ergibt sich aus der Entfernung zwischen Heliostat und Strahlern. Dieser Heliostat steht auf der y-Achse des Turm-KOS, sodass sich die Berechnung der Winkelabweichungen in diesem Fall vereinfacht.

Eine Abweichung der Kameraposition in x-Richtung des Turm-KOS stellt hierbei den größten Einfluss auf die Winkelabweichung dar. Diese berechnet sich durch

$$F_{\oplus Kamera} \propto \tan^{-1} \left( \frac{\Delta \oplus Kamera}{36 m} \right). \quad (5.3)$$

Ist die Kameraposition bis auf 1 cm genau bestimmt, so ist die resultierende Winkelabweichung 0,278 mrad.

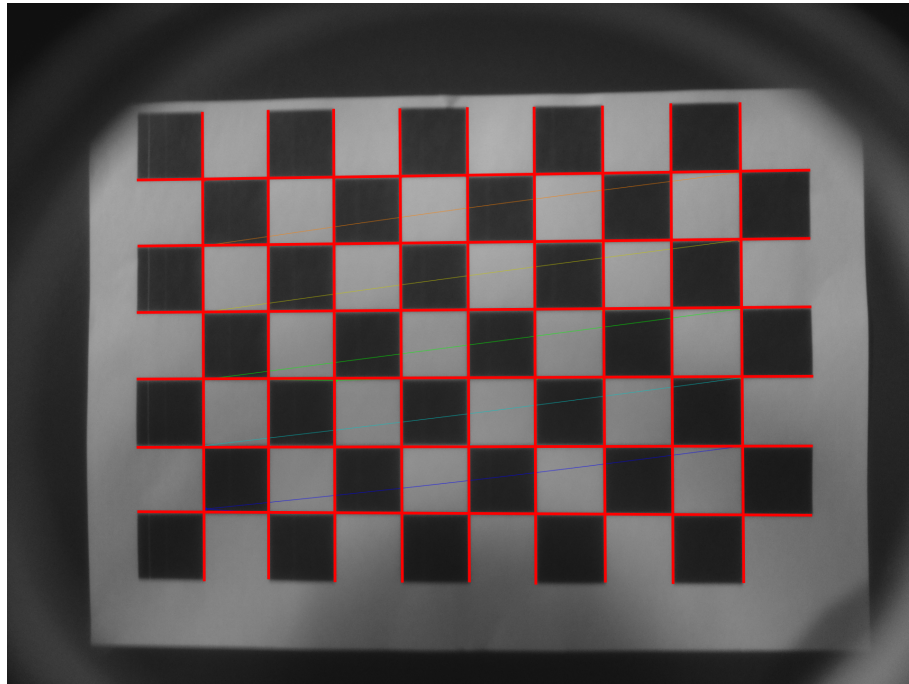


Abbildung 5.2: Entzerrte Aufnahme des Kalibrierungs-Schachfeldes. In rot überlagert sind die Kanten des entzerrten Schachbrettes

### 5.1.3 Positionierung der Halterung am Spiegel

Die Fehlerquellen der Positionierung der Halterung am Spiegel gehen auf dieselbe Weise in die gesamte Fehlerberechnung mit ein, wie die Fehler der Kamera auf der Halterung. Somit ist auch hier

$$F_{\angle Halterung} \propto \Delta \angle Halterung \quad (5.4)$$

und

$$F_{\oplus Halterung} \propto \tan^{-1} \left( \frac{\Delta \oplus Halterung}{36 m} \right). \quad (5.5)$$

Die Genauigkeit der Positionierung der Halterung am Spiegel liegt in der selben Größenordnung wie die der Positionierung der Kamera auf der Halterung. Allerdings sind auch hier durch falsche Montage große Winkelabweichungen möglich, beispielsweise durch eine nicht richtig anliegende Halterung oder einen falschen Rastenwinkel.



### 5.1.4 Position und Ausrichtung des Spiegels

Die Position des Spiegels stellt den größten Anteil der Positionsfehler dar. So ist die Position der Heliostatpylonen beim Bau des Heliostenfeldes bis auf 10 cm genau bestimmt worden. Der dadurch auftretende Fehler beträgt

$$F_{\oplus Heliostat} \propto \tan^{-1} \left( \frac{\Delta \oplus Heliostat}{36 m} \right). \quad (5.6)$$

Dadurch ergibt sich ein Fehler von bis zu 2,78 mrad.

Die Ausrichtung des Heliostaten kann allerdings für einen noch größeren Fehler in der Positionierung der Kamera im Raum sorgen, da eine Drehung im Azimut des Heliostaten um  $\Delta\alpha$  die Kameraposition um mehrere Meter verschieben kann, wie in Gleichung 5.7 zu sehen ist.

$$F_{\oplus Heliostatdrehung} \propto \tan^{-1} \left( \frac{\sin(\Delta\alpha) * b_H/2}{36 m} \right) \quad (5.7)$$

In der ersten Iteration ergibt sich damit ein Fehler von maximal 40 mrad, sollte eine Verdrehung von  $60^\circ$  vorliegen, welche die größte noch messbare Abweichung darstellt. Die Verschiebung aufgrund der Heliostatdrehung kann allerdings mit Hilfe einer iterativen Berechnung eliminiert werden. Im ersten Iterationsschritt setzt der Algorithmus die Kamera an diejenige Position, welche sie in der Referenzposition des Heliostaten einnimmt. In den folgenden Iterationen wird die Position jeweils anhand der zuvor berechneten Heliostatdrehung festgesetzt, sodass sich die Kameraposition der realen Position annähert. In Abbildung 5.3 ist zu erkennen, dass die zweite Iteration systematische Abweichungen im vorderen Feld ausgleicht, aber auch durch falsche Solverergebnisse einzelne Verschlechterungen verursachen. Eine dritte Iteration kann, wie in Abbildung 5.4 zu sehen ist, diese einzelnen Verschlechterungen wieder ausgleichen, führt aber an anderen Stellen zu den selben falschen Solverergebnissen, weshalb mehr als zwei Iterationen nicht notwendig sind.

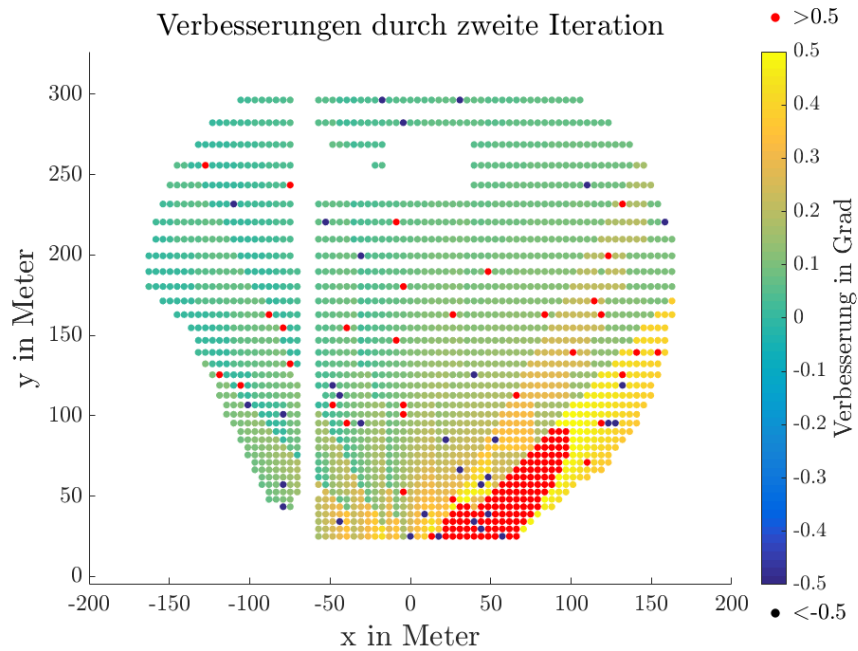


Abbildung 5.3: Verbesserungen der Ergebnisse des anhand der x-Achse bestimmten Azimutwinkels (in Grad) durch die Durchführung der zweiten Iteration gegenüber der Referenzposition. Ergebnisse anhand zuvor simulierter Eingangswerten bestimmt. Heliostattendrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

## 5.2 Softwarefehler

Fehler in der Bilderkennung, in dem Solver zur Festlegung der Bildmitte und in den Solvern zur Winkelbestimmung anhand des Kamera-KOS führen zu Ungenauigkeiten in der Winkelbestimmung und fallen in die Kategorie der Softwarefehler. Im Folgenden werden diese Einflüsse genauer betrachtet und deren Auswirkung auf die gemessenen Winkel bestimmt.

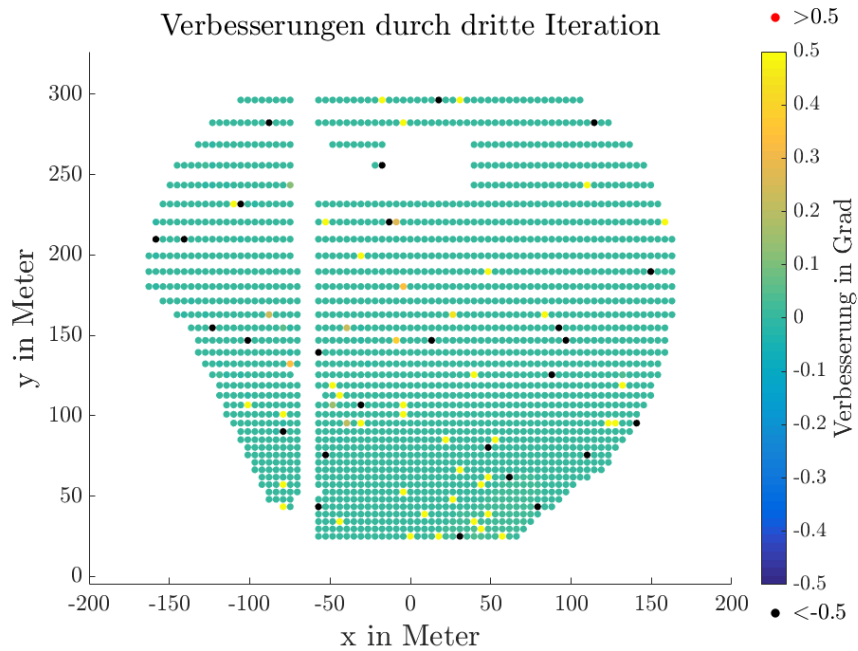


Abbildung 5.4: Verbesserungen der Ergebnisse des anhand der x-Achse bestimmten Azimutwinkels (in Grad) durch die Durchführung der dritten Iteration gegenüber der zweiten Iteration. Ergebnisse anhand zuvor simulierter Eingangswerten bestimmt. Heliostattendrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

### 5.2.1 Genauigkeit der Bilderkennung

Die Genauigkeit der Bilderkennung hängt stark von den Umgebungsbedingungen ab. Findet die Messung beispielsweise nachts statt, so sind nur die beiden Strahler im Bild vorhanden und die Auswertung kann pixel-genau geschehen. Gibt es allerdings viele helle Punkte im Bild, wie es beispielsweise an einem Tag mit Betrieb auf der Forschungsebene der Fall wäre, kann es vorkommen, dass die die Bilderkennung die Strahler nicht richtig erfasst.

Der Algorithmus sucht während der Bilderkennung nur nach hellen Flächen, welche den Abstand aufweisen, den die Strahler im idealen Fall im Bild zeigen. Dadurch schließt dieser einen Großteil der falschen Ergebnisse von vornherein aus. Kommt es allerdings dazu, dass einer der Strahler als größere Fläche erkannt wird, als er

eigentlich darstellt, beispielsweise durch auf den Strahler reflektierte Sonnenstrahlen, so kann der Mittelpunkt des Strahlers nicht korrekt festgelegt werden. Dadurch kann es auch in Einzelfällen zum Ausschluss von eigentlich sichtbaren Strahler, oder zur Verfälschung der Strahlerpositionen im Bild kommen.

Für die Berechnung der Winkelabweichung durch die Bilderkennung ist vereinfacht anzunehmen, dass eine Abweichung zur Bildmitte vorliegt. In diesem Fall errechnet sich die Fehlergröße durch:

$$F_{\angle Pixel} \propto \tan^{-1} \left( \frac{\Delta Pixel}{b_{Bild,Pixel}/2} * \tan \left( \frac{\angle FOV}{2} \right) \right). \quad (5.8)$$

Mit den Werten der Bildbreite  $b_{Bild,Pixel} = 3280$  und des Blickwinkels  $\angle FOV = 63,49^\circ$  ergibt sich eine Abweichung von  $0,368$  mrad pro Pixel. Für kleine Abweichungen kann der Zusammenhang als annähernd linear angesehen werden.

### 5.2.2 Vorgehen zur Bestimmung der Testwerte

Um den in Kapitel 4 beschriebenen Algorithmus in Simulierten Versuchen auf seine Funktionalität prüfen zu können, sind valide Werte der Strahlerpositionen aus Sicht der Kamera nötig. Anhand dieser kann die Genauigkeit des Algorithmus bestimmt werden, ohne äußere Einflüsse durch die Halterung, die Kamera oder die Bilderkennung berücksichtigen zu müssen.

Für eine vorgegebene Verdrehung der Kamera um die Winkel  $\alpha_H$  und  $\varepsilon$  und die bekannte Positionen des Heliostaten und dessen Geometrie geschieht die Berechnung der Pixelpositionen der Strahler wie folgt:

1. Berechnung der y-Ache und des  $\lambda$ -Wertes aus  $\alpha_H + \alpha_{Raste}$  und  $\varepsilon_H + \varepsilon_{Raste}$  anhand der Formeln in Abschnitt 4.5
2. Berechnung der Kameraposition anhand der Formeln in Abschnitt 4.2

3. Berechnung der Positionen der Strahler in der Bildebene, ohne Berücksichtigung von  $\lambda$ , anhand der Formeln aus Abschnitt 3.1.1. Hierbei ist aber nicht die Heliostatposition als Ursprung der Projektion zu sehen, sondern die in Punkt 2 berechnete Kameraposition. Der Mittelpunkt der Bildebene ist anhand der zuvor bestimmten Kamera y-Achse festzulegen, und anstatt des Target-Mittelpunktes zu verwenden
4. Verdrehung der Bildpositionen um den Winkel  $\lambda$  anhand der Gleichung

$$P_{Strahler,Bildebene,gedreht} = \begin{pmatrix} \cos(\lambda) & 0 & -\sin(\lambda) \\ 0 & 1 & 0 \\ \sin(\lambda) & 0 & \cos(\lambda) \end{pmatrix} * P_{Strahler,Bildebene} \quad (5.9)$$

5. Umrechnen der Bildpositionen in Pixel Werte anhand des Faktors  $mpx$  aus Gleichung 4.17 und Berechnen der Pixel-Koordinaten der Strahler in dem Koordinatensystem, wie es in Abbildung 4.12 dargestellt ist, durch die Formel

$$P_{Pixel} = \begin{pmatrix} b_{Bild}/2 \\ 0 \\ h_{Bild}/2 \end{pmatrix} - \frac{P_{Strahler,Bildebene,gedreht}}{mpx} \quad (5.10)$$

### 5.2.3 Genauigkeit des Solvers zur Bildmittenerkennung

Der Solver zur Bildmittenerkennung kann nur zwei mögliche Lösungspunkte berechnen. In diesem Fall wurde die `fsolve` Funktion der SciPy Library verwendet. Durch die nachträgliche Prüfung des Lösungspunktes auf Plausibilität ist die Genauigkeit der Bildmittenerkennung beschränkt durch falsche Berechnungen seitens des Solvers und durch Fehler in der vorangehenden Kamerapositionsbestimmung.

Bei der Prüfung auf Plausibilität berechnet der Algorithmus jeweils anhand der aus den zwei Lösungspunkten resultierenden y-Achsen die Strahlerpositionen im Bild mit Hilfe der in Abschnitt 5.2.2 erwähnten Vorgehensweise. Der Algorithmus verwirft den Lösungspunkt, welcher die größere Abweichung zwischen berechneten und gemessenen Strahlerpositionen im Bild aufzeigt (siehe Abschnitt 4.3). In diesem

Fall ergaben sich für 1,03% der Heliostaten eine Bildmitte, welche um mehr als einen Millimeter von ihrer eigentlichen Idealposition abwich, wie in Abbildung 5.5 zu sehen ist. Der Bildmitten-Solver ist so angepasst, dass er bei einem Unterschied von mehr als 50 Pixeln zwischen im Foto gemessenen und durch den Solver errechneten Strahlerpositionen erneut durchläuft. Der Solver iteriert höchstens zweifach, um die Möglichkeit einer Endlosschleife auszuschließen.

Die Winkelabweichung durch die Abweichung der Bildmitte ergibt sich aus der Definition der Achsenlänge von 1 m zu

$$F_{\oplus y\text{-Achse}} \propto \tan^{-1} \left( \frac{\Delta \oplus y\text{-Achse}}{1 \text{ m}} \right) \quad (5.11)$$

Somit entspricht eine Abweichung von 1 mm in der Bildmittenbestimmung einer Bildabweichung von 1 mrad.

### 5.2.4 Genauigkeit des Solvers zur Winkelbestimmung

Der in Kapitel 4 vorgestellte Algorithmus zur Ausrichtungsbestimmung enthält systematische Fehler, welche unvermeidbar sind. Diese Fehler lassen sich auf die Rasteneinstellung und die Heliostatenposition zurückführen. In Abbildung 5.7 ist die Verteilung der errechneten Rasteneinstellungen dargestellt, welche sich in den Auswertungen der Winkel teilweise widerspiegelt. Die Rasteneinstellung erfolgt in Abhängigkeit der  $H_{yaw}$  Winkeln der einzelnen Heliostaten, welche in Abbildung 5.6 dargestellt sind. Die Berechnungen in diesem Abschnitt basieren auf zuvor anhand der in Abschnitt 5.2.2 beschriebenen Vorgehensweise generierten Strahlerpositionen für den Heliostaten BE48, welcher um  $-15^\circ$  in Azimuts- und um  $-5^\circ$  in Elevationsrichtung aus seiner Referenzposition heraus gedreht wurde. Im Anhang finden sich ebenfalls die Ergebnisse der selben Berechnung für den Heliostaten in seiner Referenzposition, ohne zusätzliche Verdrehungen, in den Tabellen A.4, A.5 und A.6.

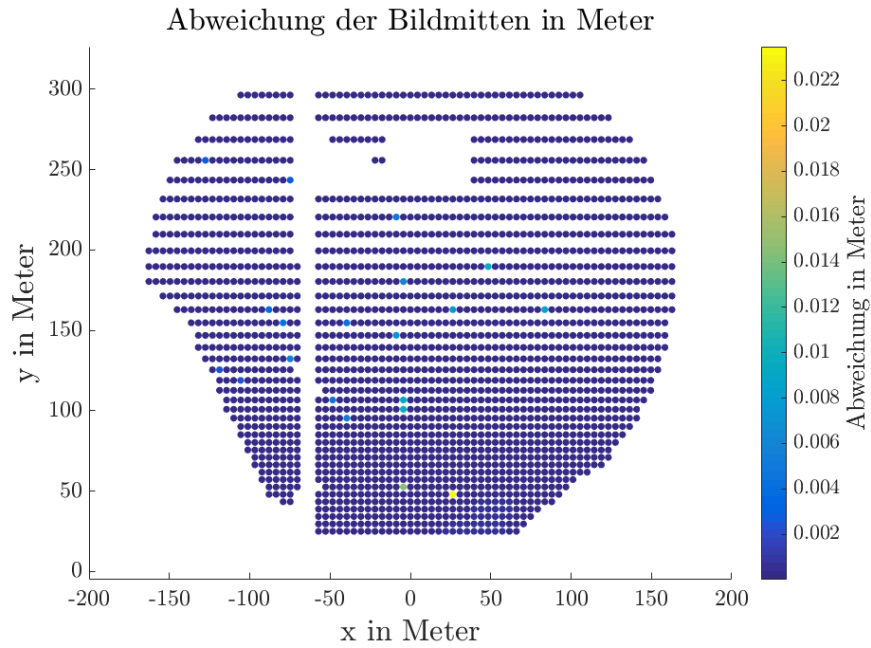


Abbildung 5.5: Quadratische Abweichungen zwischen errechneten und idealen Bildmittenposition in Metern nach zwei Iterationen. Die Heliostaten mit großen Abweichungen erklären sich durch Fehler in der Bestimmung der Bildmitte seitens des Solvers. Heliostattendrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

#### 5.2.4.1 Genauigkeit der x-Achse

In Gleichung 4.38 wird ersichtlich, dass die Bestimmung von  $\varepsilon$  bei bekanntem  $\alpha_{Raste}$  anhand der z-Komponente der x-Achse geschehen kann. Anhand des  $\varepsilon$ -Wertes und den x- und y-Komponenten der x-Achse errechnet der Solver den dazu passenden  $\alpha_H$ -Wert. Die Bestimmung der Elevation und des Azimutwinkels geschieht im endgültigen System anhand der x-Achse, da diese sowohl stabil gegenüber Verdrehungen, als auch im Falle von  $\alpha_{Raste} = 0^\circ$  vereinfacht berechenbar ist.

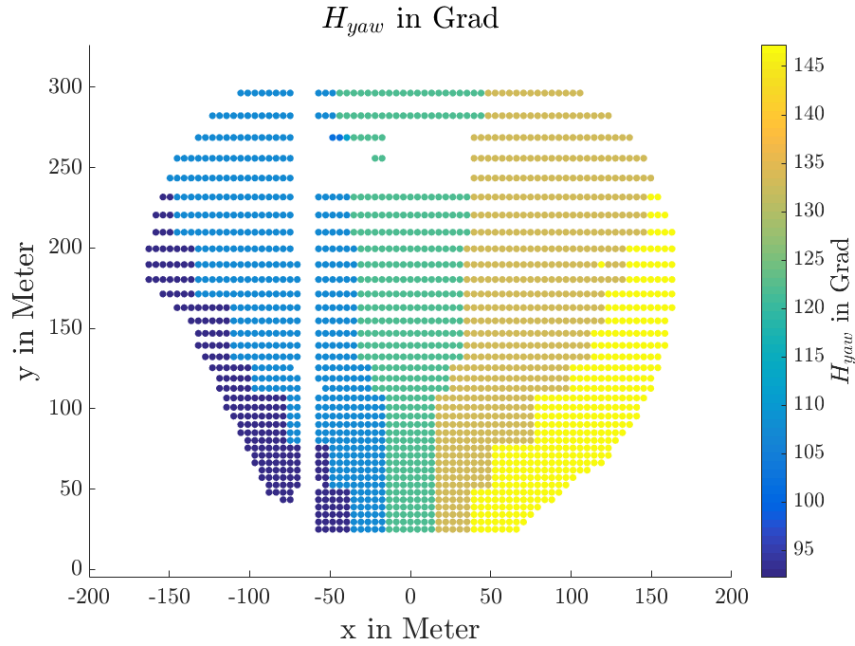


Abbildung 5.6: Ideale Referenz-Azimutswinkel  $H_{yaw}$  aller betrachteten Heliostaten

Die Fehlergröße des Solvers ist direkt proportional zu den Abweichungen in der Berechnung, da der Solver in Azimutrichtung direkt den gesuchten Winkel ausgibt und in Elevationsrichtung lediglich um  $\varepsilon_{Raste}$  verschoben ist.

$$F_{\angle x-Solver} \propto \Delta \angle x-Solver \quad (5.12)$$

Die Genauigkeit des Solvers ist nur begrenzt durch die Gleitpunktarithmetik der Programmiersprache. Da in diesem Fall Python auf einem 32-bit System verwendet wird, ergibt sich für die Berechnung der Azimutswerte eine maximale Auflösung von  $1,33 \cdot 10^{-4}$  mrad. Für  $\varepsilon$  ergibt sich eine höhere Auflösung, da dieser direkt über den Sinus berechnet wird. In diesem Fall liegt die Auflösung bei  $6,19 \cdot 10^{-14}$  mrad. In Ausnahmefällen liefert der Solver allerdings Ergebnisse weit außerhalb der geforderten Genauigkeit, wenn die richtige Lösung nicht erreicht wurde. Die Verteilung der Genauigkeiten von 20000 einzelnen Berechnungen der Winkel eines einzelnen, aus der Referenzposition verdrehten, Heliostaten sind Tabelle 5.1 zu entnehmen. Eine Darstellung der dabei ermittelten Werte erfolgt in den Abbildungen A.1, A.2, A.3, A.10



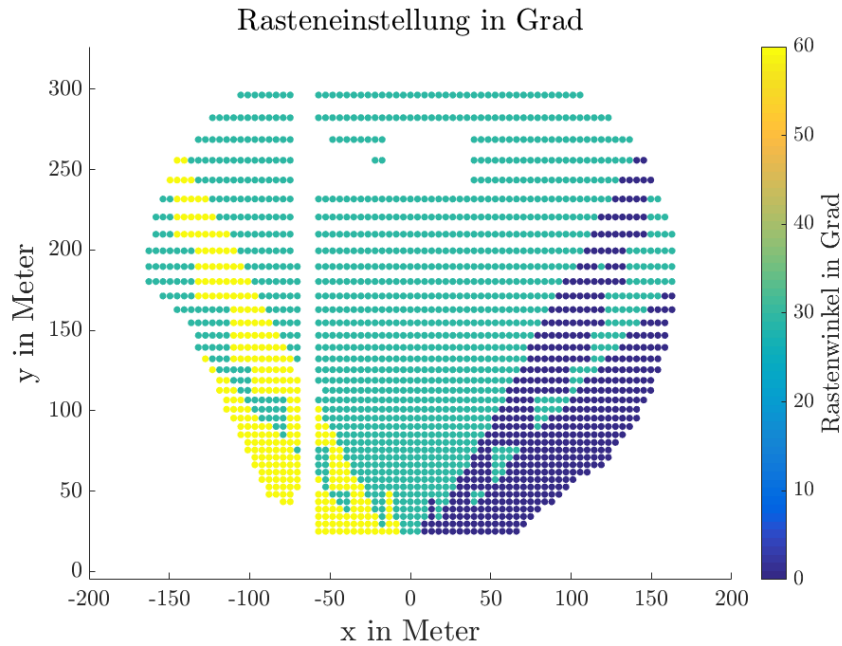


Abbildung 5.7: Einstellung der Rasten zur Kameradrehung, abhängig von Heliostatposition und  $H_{yaw}$

und A.11. Ebenso befinden sich im Anhang die Ergebnisse der selben Berechnung, jedoch ohne eine Verdrehung aus der Referenzposition heraus, in Tabelle A.4.

In den Abbildungen 5.8 und 5.9 sind die Ergebnisse jeweils einer Berechnung der Winkel für alle betrachteten Heliostaten des Solarturmes in Jülich dargestellt. Diese Ergebnisse zeigen, dass die Genauigkeit des Systems seitens der Software für  $\varepsilon$  weit unter den geforderten  $0,5^\circ$  ist. In  $\alpha_H$  gibt es seitens des Solvers noch Probleme, welche zu größeren Abweichungen in der Winkelberechnung führen. Wird ein anderer Solver oder eine vereinfachte Berechnung herangezogen, so kann die Genauigkeit der Berechnung des Azimutswinkels weiter erhöht werden. Der in diesem Fall verwendete Solver war die minimize Funktion der SciPy Library. Diese wurde gewählt, da sie Open Source und somit frei verfügbar ist, und das verwendete Gleichungssystem lösen kann. Auch bietet dieser verschiedene Lösungsmethoden an, welche allerdings nicht näher betrachtet wurden, da dies aus zeitlichen Gründen nicht mehr möglich war.

## 5. Bewertung des Algorithmus

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	-0,063	5,69	198	1,363	1,402	86,4	-141,41	11,80
2	-0,0499	8,95	228	1,391	1,614	95,9	-141,41	10,56
3	-0,0385	8,14	0,029	1,175	0,0013	91,5	11,35	11,06

Tabelle 5.1: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe aus 20000 berechneten Unterschieden zwischen den berechneten und vorgegebenen Winkeln des Heliostaten BE48. Werte anhand der x-Achse berechnet.  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

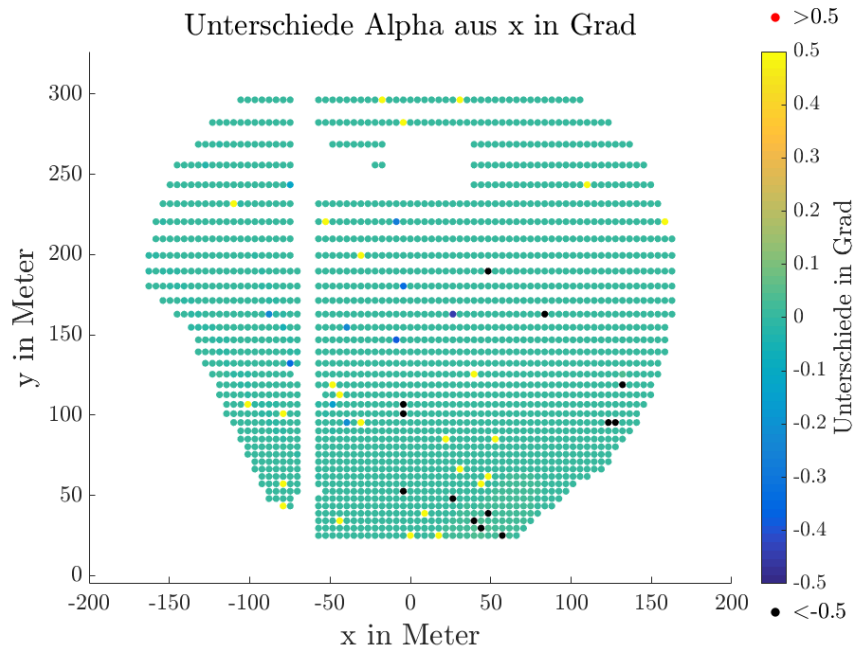


Abbildung 5.8: Unterschiede zwischen  $\alpha$  aus x berechnet und dem vorgegebenen  $\alpha$  nach zwei Iterationen. Heliostadrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

### 5.2.4.2 Genauigkeit der y-Achse

In der Elevationsausrichtung ist die Winkelbestimmung anhand der y-Achse um eine Größenordnung genauer als die anhand der x-Achse. In Azimutsrichtung kommt es

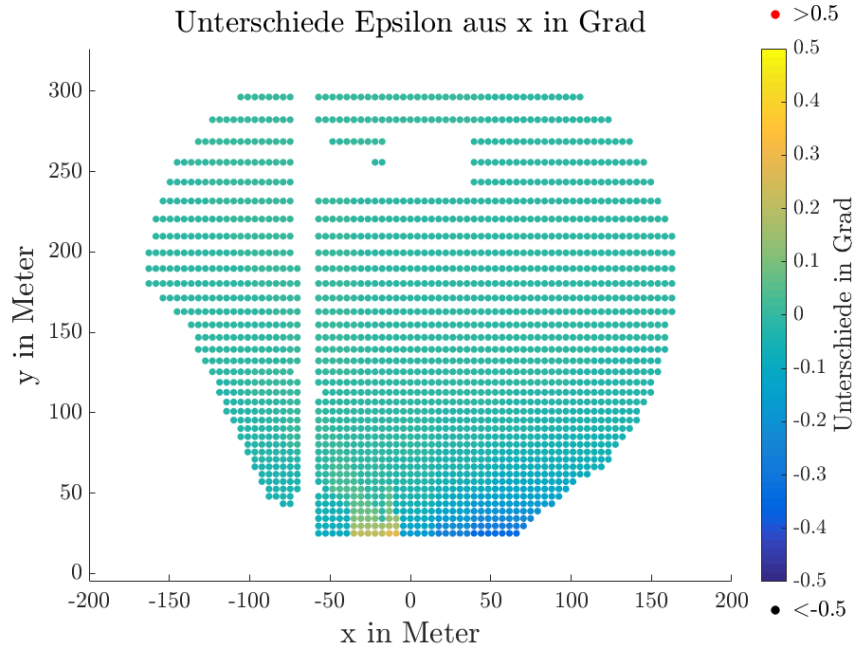


Abbildung 5.9: Unterschiede zwischen  $\varepsilon$  aus  $x$  berechnet und dem vorgegebenen  $\varepsilon$  nach zwei Iterationen. Heliostaddrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

solverbedingt zu größeren Abweichungen, wenn die richtige Lösung nicht erreicht wird. Auch ist es anhand der  $y$ -Achse für den Rastenwinkel von  $\alpha_{Raste} = 0^\circ$  nicht möglich,  $\varepsilon$  zu bestimmen. In den Abbildungen 5.10 und 5.11 ist ersichtlich, für welche Heliostate die systematischen Fehler in der  $\varepsilon$ -Berechnung auftreten und dass deutlich mehr falsche Ergebnisse in  $\alpha_H$  erzeugt werden.

Anhand der  $y$ -Achse ergibt sich ebenso wie bei der  $x$ -Achse eine maximale Auflösung von  $1,33 \cdot 10^{-4}$  mrad in Azimutsrichtung. Auch in Elevationsrichtung ist die Auflösung nahezu identisch mit derer der Bestimmung anhand der  $x$ -Achse. In diesem Fall beträgt sie  $6,2 \cdot 10^{-14}$  mrad. In Tabelle 5.2 ist die Auswertung der Ergebnisse der Winkelberechnung eines aus der Referenzposition verdrehten Heliostaten anhand der  $y$ -Achse für 20000 einzelne Berechnungen dargestellt. Diese Werte sind ebenfalls den Abbildungen A.4, A.5, A.6, A.12 und A.13 zu entnehmen. Ebenso befinden sich im Anhang die Ergebnisse der selben Berechnung, jedoch ohne eine Verdrehung aus der Referenzposition heraus, in Tabelle A.5.

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	0,3835	13,92	0,0009	1,017	0,00005	124,6	57,96	7,91
2	-0,0045	16,3	0,6753	1,024	0,0312	127,8	-12,74	7,69
3	-0,0033	15,26	0,7849	1,024	0,0346	123,8	-11,35	7,95

Tabelle 5.2: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe aus 20000 berechneten Unterschieden zwischen den berechneten und vorgegebenen Winkeln des Heliostaten BE48. Werte anhand der y-Achse berechnet.  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

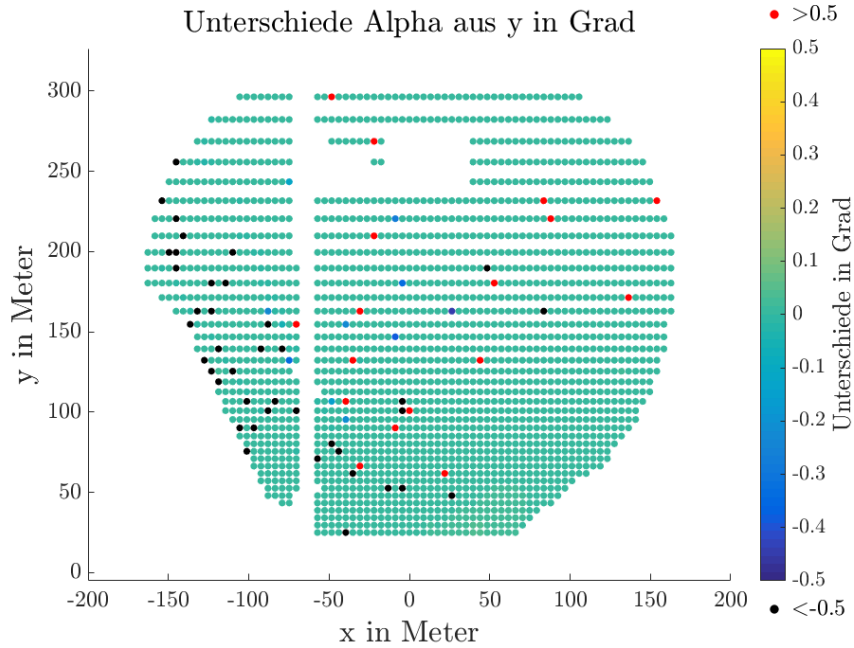


Abbildung 5.10: Unterschiede zwischen  $\alpha$  aus y berechnet und dem vorgegebenen  $\alpha$  zwei Iterationen. Heliostaddrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

### 5.2.4.3 Genauigkeit der z-Achse

Zur Winkelbestimmung anhand der z-Achse wird kein Solver benötigt, da die Berechnung einfach genug ist, um nur durch 4 unterschiedliche Fälle einfacher Sinus-

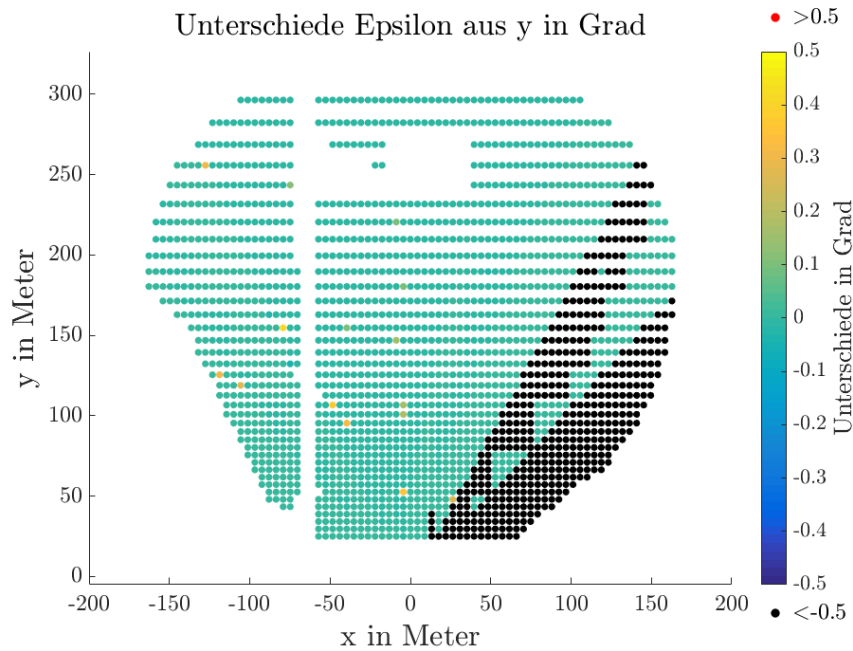


Abbildung 5.11: Unterschiede zwischen  $\varepsilon$  aus  $y$  berechnet und dem vorgegebenen  $\varepsilon$  nach zwei Iterationen. Heliostaddrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

und Cosinusberechnungen abgedeckt zu werden. Dadurch erhöht sich die Genauigkeit in Azimutsrichtung gegenüber der  $x$ - und  $y$ -Achse um 3 Größenordnungen, wie in Tabelle 5.3 und den zugehörigen Abbildungen A.7, A.8, A.9, A.14 und A.15 zu sehen ist. Deutlich erkennbar sind dort sowohl eine höhere Genauigkeit der Ergebnisse in Azimutsrichtung, als auch eine größere Varianz der Ergebnisse in Elevationsrichtung. Im Anhang finden sich die Ergebnisse der selben Berechnung, jedoch ohne eine Verdrehung aus der Referenzposition heraus, in Tabelle A.4. Anhand der  $z$ -Achse ergibt sich ebenso wie bei den  $x$ - und  $y$ -Achsen eine maximale Auflösung von  $1,33 \cdot 10^{-4}$  mrad in Azimutsrichtung. In Elevationsrichtung beträgt die maximale Auflösung in diesem Fall  $1,24 \cdot 10^{-13}$  mrad.

In den Abbildungen 5.14 und 5.13 ist zu erkennen, dass die Winkelbestimmung anhand der  $z$ -Achse ebenso wie die anhand der  $x$ -Achse, nur wenige Fehler erzeugt und geringfügige systematische Fehler im südlichen Bereich des Feldes aufzeigt. Die Ähnlichkeit der Ergebnisse ist dadurch zu begründen, dass  $x$ - und  $z$ -Achsen anhand

## 5. Bewertung des Algorithmus

der selben Werte der y-Achse und der Bildverdrehung erzeugt werden. Die Berechnung der Heliostatverdrehung, sowohl in Azimuts- als auch Elevationsrichtung, ist anhand der z-Achse durchzuführen, da diese die höchste Genauigkeit aufweist.

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	0,092	-2,08	711,30	0,0377	5,03	0,266	141,41	141,41
2	0,0096	-0,0082	792,06	0,1007	5,6	0,9	141,41	50,52
3	-0,0297	-0,011	0,175	0,0101	0,0077	0,68	-11,34	-10,56

Tabelle 5.3: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe aus 20000 berechneten Unterschieden zwischen den berechneten und vorgegebenen Winkeln des Heliostaten BE48. Werte anhand der z-Achse berechnet.  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

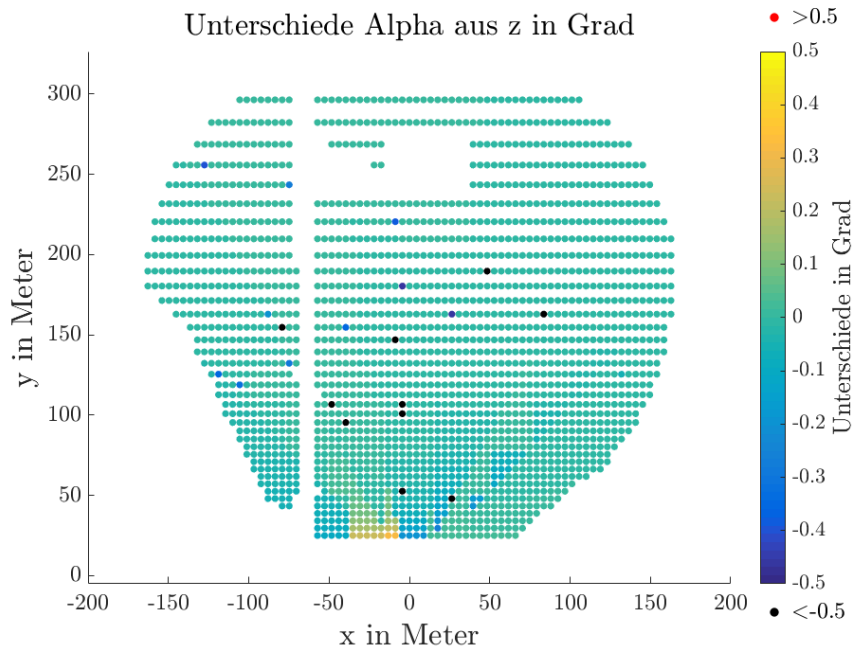


Abbildung 5.12: Unterschiede zwischen  $\alpha$  aus z berechnet und dem vorgegebenen  $\alpha$  nach zwei Iterationen. Heliostatdrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

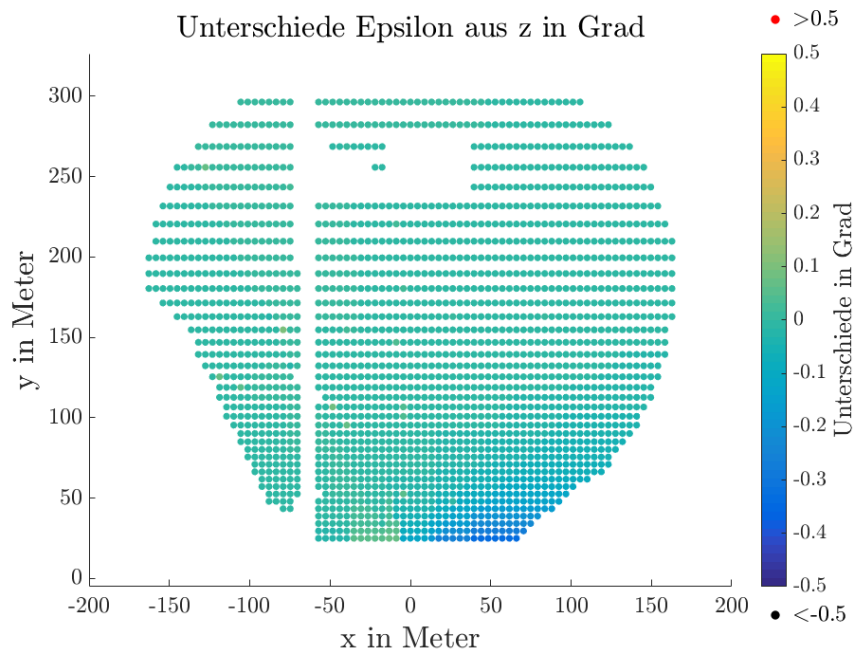


Abbildung 5.13: Unterschiede zwischen  $\varepsilon$  aus  $z$  berechnet und dem vorgegebenen  $\varepsilon$  nach zwei Iterationen. Heliostaddrehung  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

### 5.3 Gesamtfehler des Systems

Die größten Einflüsse auf die Genauigkeit des Systems entstehen durch die Winkelberechnungen anhand der Achsen. Diese haben eine Spannweite von bis zu 1,391 rad, wie in Tabelle 5.1 ersichtlich ist. Durch eine andere Herangehensweise der Berechnung könnte sich diese Spannweite allerdings um mehrere Größenordnungen verringern. Ein weiterer großer Einfluss ist die falsche Positionsbestimmung der Kamera aufgrund der Heliostaddrehung. Die dabei auftretenden maximalen 40 mrad können aber in folgenden Iterationen nahezu eliminiert werden. Der nächstgrößte Einfluss kommt durch die Verzerrung des Bildes mit bis zu 8,4 mrad. Auch diesen kann die Bildverarbeitung eliminieren.

Alle diese Fehler können sich aufaddieren oder aufheben. Darum ist an dieser Stelle zu sagen, dass die in Gleichung 5.14 erläuterte Fehlerbetrachtung nur eine Ab-

schätzung der Einflussgrößen der Fehlerquellen darstellen soll und nicht die genauen Ursachen für eine bestimmte Abweichung ergibt. Alle Positionsabweichungen des Systems addieren sich zum Positionsgesamtfehler  $F_{\oplus}$  auf. Dieser wird mit sämtlichen auftretenden Winkelfehlern zum Gesamtfehler  $F_{ges}$  aufaddiert.

$$\begin{aligned}
 F_{\oplus} &= F_{\oplus Kamera + \oplus Halterung + \oplus Heliostat + \oplus Heliostatdrehung + \oplus y-Achse} \\
 &\propto \tan^{-1} \left( \frac{\Delta \oplus Kamera + \Delta \oplus Halterung + \Delta \oplus Heliostat + \sin(\Delta\alpha) * b_h/2}{36 m} \right. \\
 &\quad \left. + \frac{\Delta \oplus y-Achse}{1 m} \right)
 \end{aligned} \tag{5.13}$$

$$\begin{aligned}
 F_{ges} &= F_{\angle Verzerrung} + F_{\angle Kamera} + F_{\angle Halterung} + F_{\angle Pixel} + F_{\angle x/y/z-Solver} + F_{\oplus} \\
 &\propto \Delta\angle Verzerrung + \Delta\angle Kamera + \Delta\angle Halterung \\
 &\quad + \tan^{-1} \left( \frac{AbweichungPixel}{b_{Bild,Pixel}/2} * \tan \left( \frac{\angle FOV}{2} \right) \right) \\
 &\quad + \Delta\angle x/y/z-Solver + \tan^{-1} \left( \frac{\Delta \oplus y-Achse}{1 m} \right) \\
 &\quad + \frac{\Delta \oplus Kamera + \Delta \oplus Halterung + \Delta \oplus Heliostat + \sin(\Delta\alpha) * b_h/2}{36 m}
 \end{aligned} \tag{5.14}$$

## 5.4 Versuchsreihe unter realen Bedingungen

Um die Genauigkeit des Systems unter realen Bedingungen testen zu können, wurden zwei verschiedene Versuchsaufbauten verwendet. Der erste Versuchsaufbau beinhaltete die Messung aus der Referenzposition des Heliostaten, wie es im Anwendungsfall vorgesehen ist. Für den zweiten Versuchsaufbau wurden die untersuchten Heliostaten so ausgerichtet, dass sie Sonnenstrahlen in das Target auf dem Turm reflektierten, während die Messung durch die Kamera durchgeführt wurde. Bei diesen Versuchen wurde der Prototyp der Halterung verwendet, dessen Geometrie nicht präzise vermessen wurde. Somit kann die Halterung die Messungen durch Fehler unbekanntem Ausmaßes in Ausrichtung und Positionierung der Kamera beeinflusst haben.



### 5.4.1 Messungen in Referenzposition

Da die Referenzposition der Heliostaten temperaturabhängig ist und die die Motoren der Heliostatenachsen ihre Schrittpositionen nicht über einen langen Zeitraum korrekt zählen können, ist es nötig, die Heliostatausrichtung in Referenzposition unmittelbar vor der Messung erneut festzulegen, um Werte zu erhalten, mit denen die des Kamera-Systems verglichen werden können.

Dafür wird der Heliostat zunächst in seine Referenzposition gefahren, um die Schrittzahl der Achsenmotoren in dieser Position festzulegen. Anschließend erfolgt eine Kalibrierung des Heliostaten nach dem Verfahren, wie es in Abschnitt 2.2.1 beschrieben wurde. Das Steuerungssystem des Heliostatfeldes berechnet anschließend anhand der zuvor bestimmten Ausrichtung und der zugehörigen Motorposition die Achsenlängen des Heliostaten. Diese gibt Rückschluss auf die Ausrichtung in der Referenzposition. Durch ein erneutes Anfahren der Referenzposition und anschließender erneuter Ausrichtungsbestimmung wird die Wahrscheinlichkeit einer fehlerhaften ersten Messung minimiert.

Ist die Ausrichtung des Heliostaten in seiner Referenzposition nun hinreichend genau bekannt, sind dessen  $H_{yaw}$  und  $F_{Pitch}$  Winkel dem Steuerungssystem direkt zu entnehmen. Diese bilden die Vergleichswerte, anhand derer die Genauigkeit des IR-Kamera-Systems in diesem Versuchsaufbau bestimmt wird.

Ist der Heliostat nach seiner Ausrichtungsbestimmung wieder in die Referenzposition gefahren, kann der Bediener die Messung anhand des IR-Kamera-Systems, so wie es in Kapitel 4 beschrieben wurde, durchführen.

Aufgrund ungünstiger Wetterbedingungen konnten nach Fertigstellung des Systems nur an zwei Tagen Messungen mit diesem Verfahren durchgeführt werden, sodass nur die in Tabelle 5.4.1 aufgeführten 5 Datenpunkte vorhanden sind. Auch konnten die Infrarotstrahler an den Versuchstagen nicht verwendet werden, da diese ausgefallen waren. Stattdessen wurden die Positionen der Strahler im Bild nachträglich von Hand als helle Punkte markiert. Eine statistische Auswertung dieser Datenpunkte

ist deshalb nicht ratsam. Aus dieser Messung lassen sich weder die Genauigkeit des Systems, noch die Ursprünge der Abweichungen ablesen. Weitere Messreihen sind deshalb für die Anpassung des Systems an die Umgebungsbedingungen und zur endgültigen Auswertung der Genauigkeit nötig. Allerdings zeigen diese Ergebnisse, dass das System in seiner praktischen Umsetzung bereits funktioniert, auch wenn es noch nicht die geforderte Genauigkeit erreicht.

	3. Nov				4. Nov		
	AY42	BE34	BE32	BG35	BE32	$\mu$	$\sigma$
$\Delta\alpha$	2,991	1,443	2,148	-1,547	-1,332	0,741	1,848
$\Delta\varepsilon$	-1,530	-2,548	-0,594	-1,574	-0,551	-1,360	0,738

Tabelle 5.4: Auswertung der Messung in Referenzposition. Dargestellt sind jeweils Unterschiede in Grad zwischen den durch das IR-Kamerasystem und den durch das Steuerungssystem ermittelten  $H_{yaw}$  und  $F_{Pitch}$  Werten

### 5.4.2 Messungen während der Kalibrierung

Da die Versuchsreihe in der Referenzposition durch das mehrfache Verfahren und die Geometriebestimmung durch das Steuerungssystem zusätzliche Messfehler unbekanntes Ausmaßes aufweisen, ist die Messung bei gleichzeitiger Heliostatkalibrierung ratsam.

Hierbei wird die Ausrichtungsbestimmung des IR-Kamera-Systems durchgeführt, während der Heliostat Sonnenstrahlen auf das Target am Turm reflektiert. Durch das Kalibrierungssystem ist eine genaue Positionsbestimmung der Reflexion am Turm möglich. Zusammen mit dem Sonnenstand zur Zeit der Messung und der Heliostatposition, kann die Heliostatausrichtung errechnet werden, die zu der aufgetretenen Reflexion führte. Diese bildet den Vergleichswert für die Messung des IR-Kamera-Systems.

Ebenso wie bei der Messung nach in Referenzposition aus Abschnitt 5.4.1, war es auch für diese Methode aufgrund der Wetterbedingungen nicht möglich, eine für sta-

tistische Auswertungen geeignete Messreihe durchzuführen. Auch konnten aufgrund des Kamerawinkels die Infrarotstrahler nicht als Positionsmarkierungen verwendet werden. Stattdessen wurde das System so angepasst, dass zwei Fenster des Solarturmes als Referenzpunkte dienten, deren Positionen den Bauzeichnungen des Turmes entnommen wurden. Alle ausgewerteten Datenpunkte sind Tabelle 5.4.2 zu entnehmen. Deutlich wurde in dieser Messreihe aber ebenfalls, dass das System grundsätzlich einsatzbereit ist, es aber noch einer genaueren Untersuchung der Ursachen der vorhandenen Abweichungen kommen muss.



Abbildung 5.14: Ausgewertetes Foto des Systems bei gleichzeitiger Kalibrierung. Die Kamera war am Heliostat AB29 angebracht und das System verwendete die Eckpunkten zweier ausgewählter Fenster als Referenzpunkte

## 5. Bewertung des Algorithmus

---

	4. Nov		8. Nov			
	AB29	AE29	AD29	AE29	$\mu$	$\sigma$
$\Delta\alpha$	1,488	3,371	0,103	-1,09	0,967	1,611
$\Delta\varepsilon$	0,208	-6,078	-2,367	-4,551	-3,197	2,367

Tabelle 5.5: Auswertung der Messung bei gleichzeitiger Kalibrierung. Dargestellt sind die Unterschiede in Grad zwischen den durch das IR-Kamerasystem und den anhand der Reflexion bestimmten momentanen Ausrichtungen

---

## 6 Zusammenfassung & Ausblick

Das geforderte System konnte prototypisch umgesetzt und in ersten Versuchsreihen auf seine Funktion getestet werden. Im Gegensatz zu dem momentan verwendeten System kann dieses die Ausrichtungsmessung eines Heliostaten ohne Sonnenschein und aus dessen Referenzposition heraus durchführen. Somit kann ein Heliostat mit geringem Bedieneraufwand nach Reparaturarbeiten in den Zustand gebracht werden, in welchem das momentan verwendete System die durch den Heliostat reflektierten Sonnenstrahlen im ersten Versuch und ohne unnötiges Verfahren erfassen kann. Die ersten Versuchsreihen zeigten allerdings, dass das neue System die geforderte Genauigkeit nicht erreichen konnte. So ergaben sich gemessene Genauigkeiten von  $0,97^\circ$  in Azimuts- ( $\sigma = 1,611^\circ$ ) und  $-3,2^\circ$  in Elevationsrichtung ( $\sigma = 2,367^\circ$ ). Diese Werte basieren allerdings auf wenigen Messreihen, weshalb in Zukunft eine genauere Untersuchung des Systems notwendig ist, um die wirkliche Genauigkeit erfassen zu können und mögliche Fehlerquellen zu beheben.

Eine Verbesserungsmöglichkeit des Systems ist beispielsweise die verwendete Halterung. Momentan weist diese einen Kippungswinkel  $\varepsilon_{Raste}$  von  $30^\circ$  auf, da es ansonsten zu einer Kollision zwischen der Halterungsplatte und der drehbaren Halterung kommt. Durch diesen Winkel kommt es dazu, dass bei vielen Heliostaten die am Turm angebrachten Strahler außerhalb des Blickfeldes der Kamera liegen, wie in Abbildung 6.1 zu sehen ist. Eine Verringerung dieses Winkels auf  $3^\circ$  würde die Zahl dieser Heliostaten auf die in Abbildung 6.2 dargestellten Fälle verringern.

Auch könnte noch eine abschließende erneute Berechnung der Pixel Positionen der Strahler im Bild anhand der zuvor bestimmten Heliostat-Verdrehung durchgeführt

werden, um Ergebnisse mit großen Abweichungen direkt ausschließen zu können und eine erneute Messung zu initiieren.

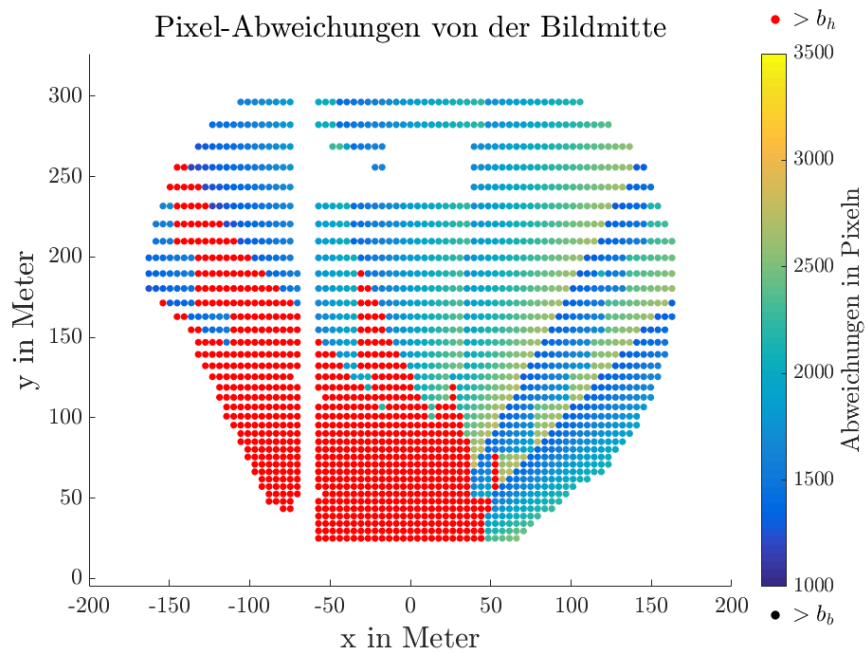


Abbildung 6.1: Verteilung der Pixel-Positionsabweichungen zum Mittelpunkt der Kamera in Referenzposition. Halterung mit  $30^\circ$  Kippungswinkel. Rote Punkte liegen außerhalb des abbildbaren Bereiches

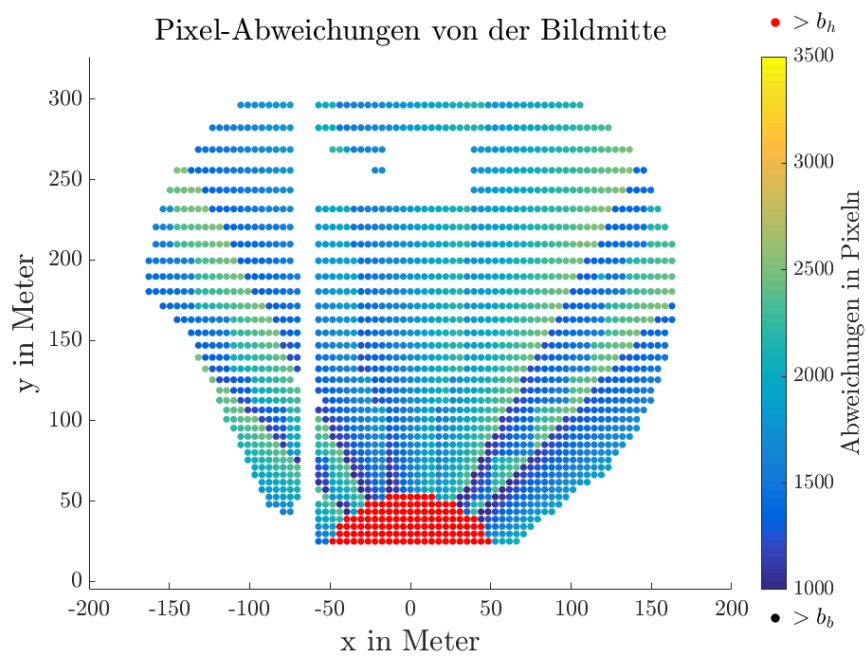


Abbildung 6.2: Verteilung der Pixel-Positionsabweichungen zum Mittelpunkt der Kamera in Referenzposition. Halterung mit  $3^\circ$  Kippungswinkel. Rote Punkte liegen außerhalb des abbildbaren Bereiches

---

# Literaturverzeichnis

Baumer AG (2015). *Baumer Ultrasonic distance measuring sensors - UNAM 70*.

Hexagon Metrology (2009). *Leica absolute tracker AT402*.

Lang, H. (2012). Genauigkeiten bei reflektorlosen Messungen elektronischer Tachymeter, Netzpublikation zur Grabungstechnik Nr. 3 herausgegeben von der Kommission Grabungstechnik des Verbandes der Landesarchäologen.

<http://www.landesarchaeologen.de/verband/kommissionen/grabungstechnik/netzpublikationen/>

Leica Geosystems AG (2015). *Leica iCON gps 60 - Intelligente Positionierung auf jeder Baustelle*.

Mordvintsev, A. and K., A. (2013a). Camera Calibration. Abgerufen 29.11.2016.

[http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_calib3d/py\\_calibration/py\\_calibration.html#calibration](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html#calibration)

Mordvintsev, A. and K., A. (2013b). Feature Detection and Description Tutorial: Introduction to SIFT. Abgerufen 29.11.2016.

[http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_feature2d/py\\_sift\\_intro/py\\_sift\\_intro.html#sift-intro](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html#sift-intro)

Mordvintsev, A. and K., A. (2013c). Feature detection and description tutorial: Orb. Abgerufen 29.11.2016.



[http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_feature2d/py\\_orb/py\\_orb.html#orb](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_orb/py_orb.html#orb)

Mordvintsev, A. and K., A. (2013d). Image Processing in OpenCV: Canny Edge Detection. Abgerufen 29.11.2016.

[http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_canny/py\\_canny.html#canny](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html#canny)

Mordvintsev, A. and K., A. (2013e). Image Processing in OpenCV: Contours in OpenCV. Abgerufen 29.11.2016.

[http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_contours/py\\_contours\\_begin/py\\_contours\\_begin.html#contours-getting-started](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html#contours-getting-started)

Mordvintsev, A. and K., A. (2013f). Image Processing in OpenCV: Template Matching. Abgerufen 29.11.2016.

[http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_template\\_matching/py\\_template\\_matching.html#template-matching](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_template_matching/py_template_matching.html#template-matching)

OpenCV.org Foundation (2016). Object detection templatematchmodes. Abgerufen 08.12.2016.

[http://docs.opencv.org/trunk/df/dfb/group\\_\\_imgproc\\_\\_object.html](http://docs.opencv.org/trunk/df/dfb/group__imgproc__object.html)

Prahl, C., Stanicki, B., Hilgert, C., Ulmer, S. and Röger, M. (2013). Airborne shape measurement of parabolic trough collector fields, *Solar Energy* .

Stone, K. (1986). Automatic heliostat track alignment method. US Patent 4,564,275.

<https://www.google.com/patents/US4564275>

Takasu, T. and Yasuda, A. (2009). Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB.

- US Department of Defense and GPS Navstar (2008). Global positioning system standard positioning service performance standard. Abgerufen 17.11.2016.  
<http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>
- van der Meulen, P. (2015). Entwicklung einer Simulationssoftware und eines Verfahrens zur Bestimmung der initialen Heliostatausrichtung mittels eines digitalen Magnetkompasses.  
<http://elib.dlr.de/98970>
- Xiao, J., Wei, X., Lu, Z., Yu, W. and Wu, H. (2012). A review of available methods for surface shape measurement of solar concentrator in solar thermal power applications, *Renewable and Sustainable Energy Reviews* .
- Yang, L., Zhang, L., Dong, H., Alelaiwi, A. and El Saddik, A. (2015). Evaluating and improving the depth accuracy of kinect for windows v2, *IEEE Sensors Journal* .

---

# A Anhang

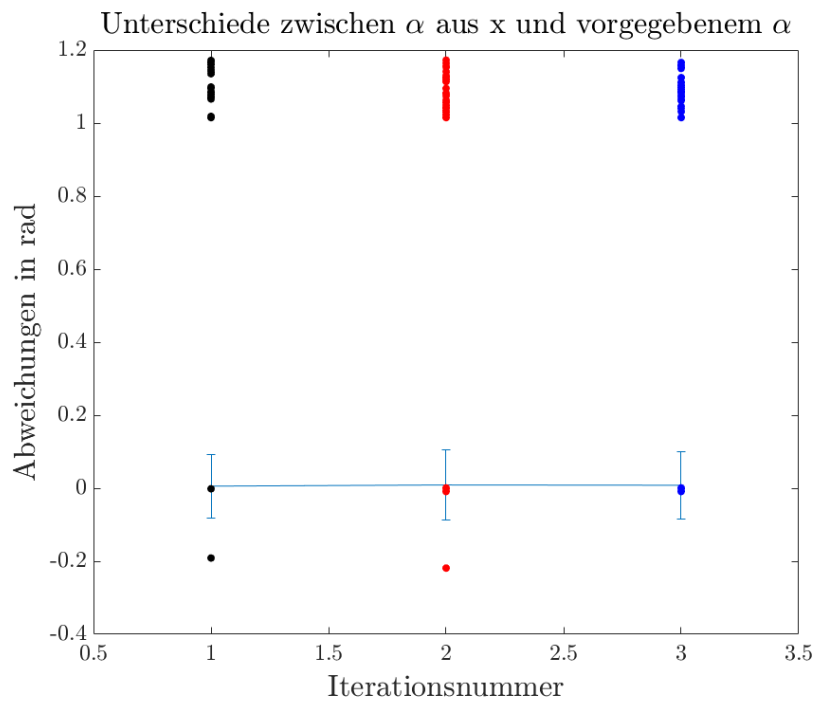


Abbildung A.1: Abweichung zwischen  $\alpha$  aus x und dem vorgegebenen  $\alpha$  berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

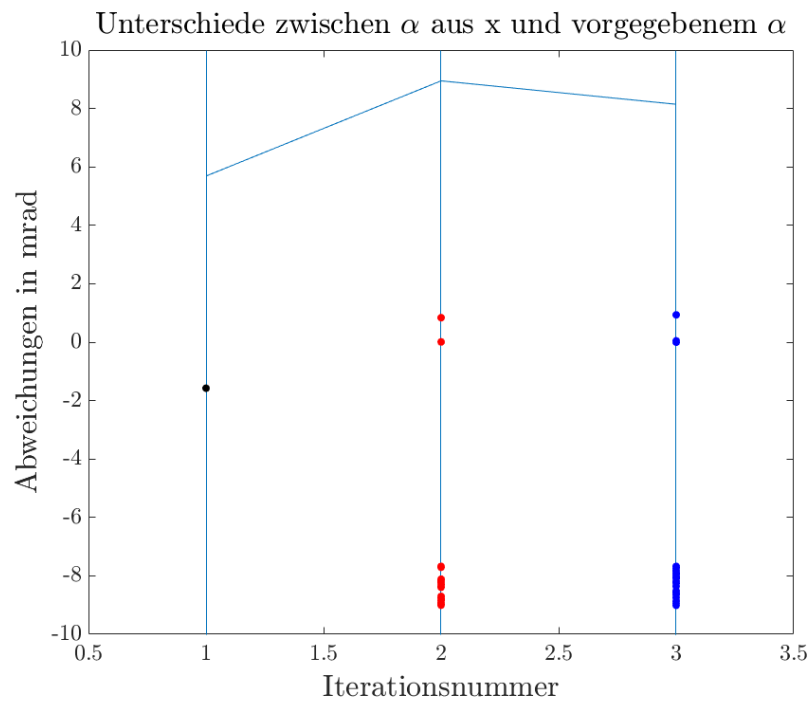


Abbildung A.2: Eingezoomte Abweichung zwischen  $\alpha$  aus x und dem vorgegebenen  $\alpha$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

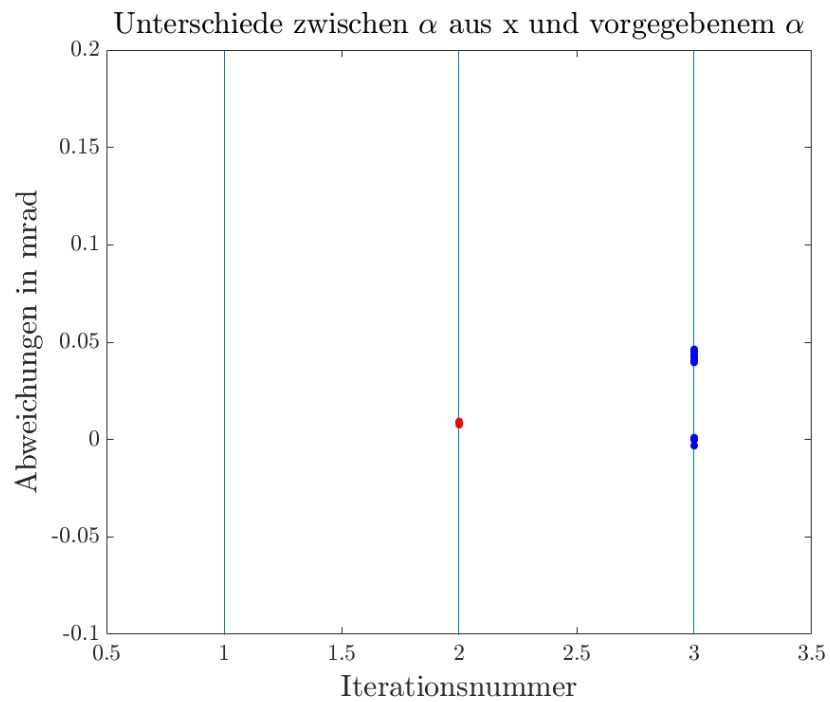


Abbildung A.3: Weiter eingezoomte Abweichung zwischen  $\alpha$  aus x und dem vorgegebenen  $\alpha$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

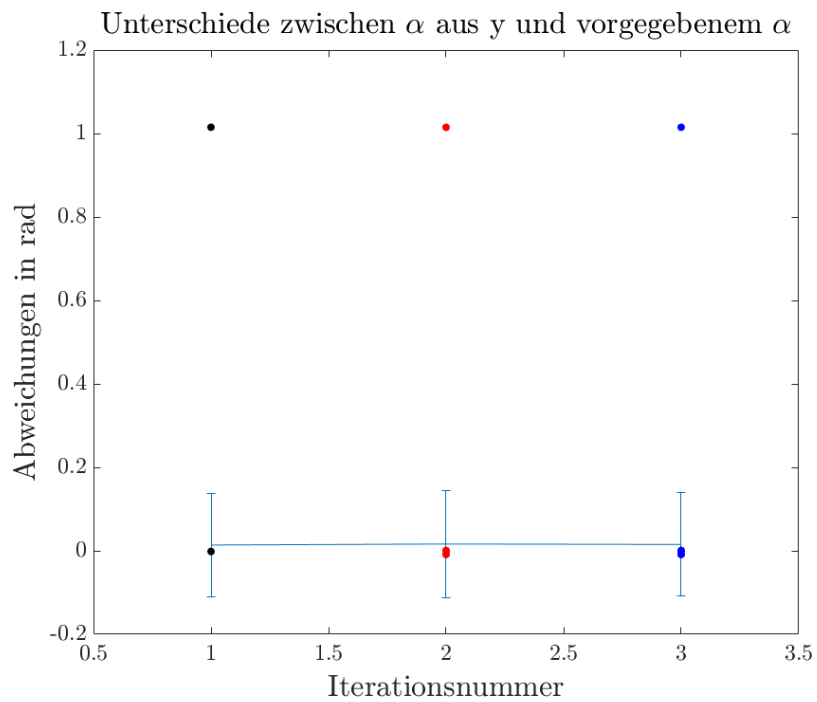


Abbildung A.4: Abweichung zwischen  $\alpha$  aus y und dem vorgegebenen  $\alpha$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

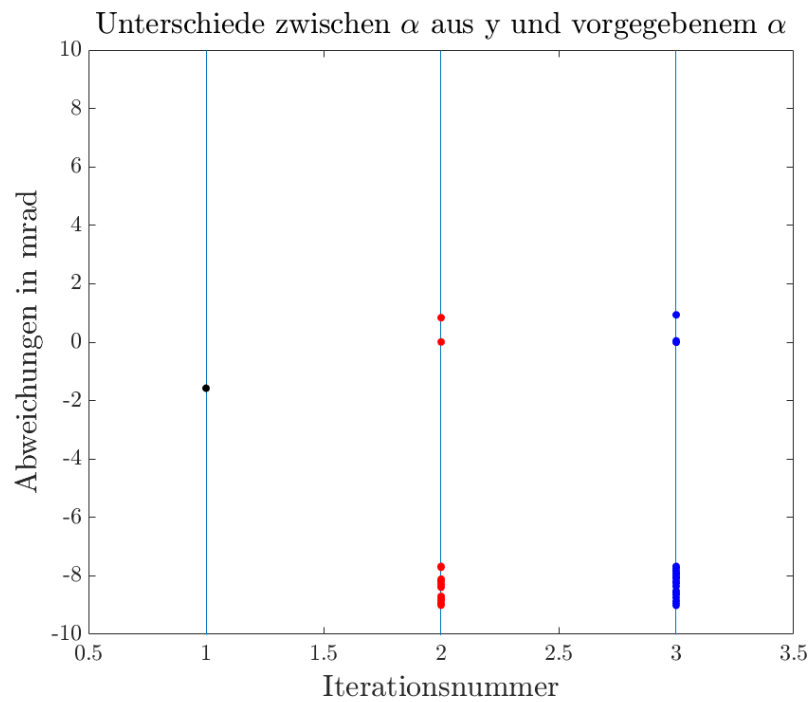


Abbildung A.5: Eingezoomte Abweichung zwischen  $\alpha$  aus y und dem vorgegebenen  $\alpha$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung





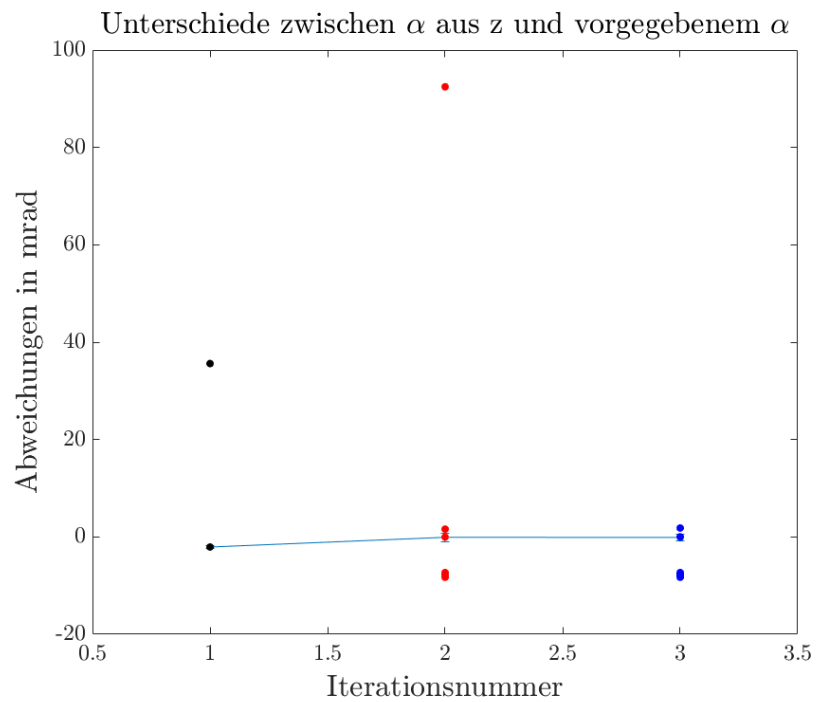


Abbildung A.7: Abweichung zwischen  $\alpha$  aus z und dem vorgegebenen  $\alpha$  berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

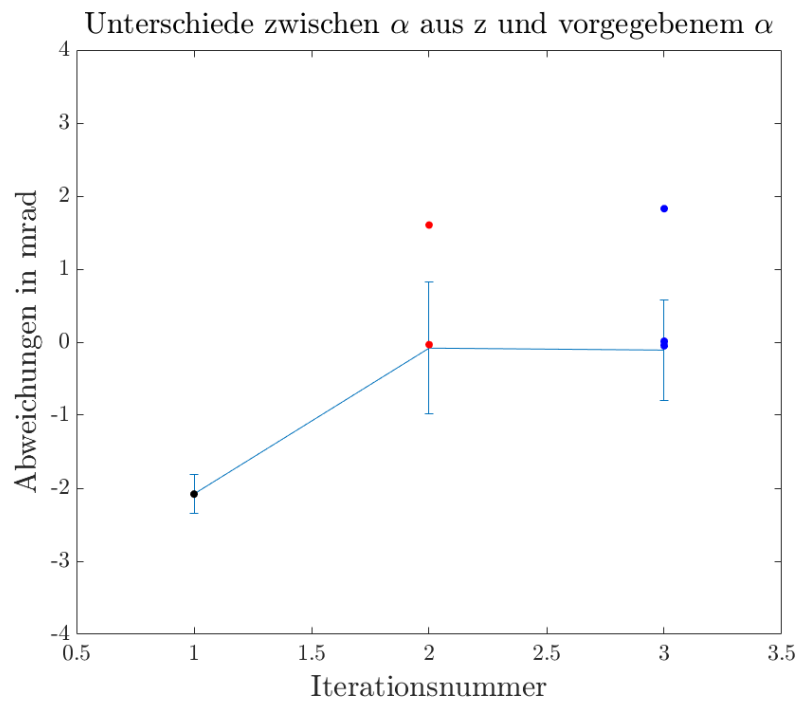


Abbildung A.8: Eingezoomte Abweichung zwischen  $\alpha$  aus z und dem vorgegebenen  $\alpha$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

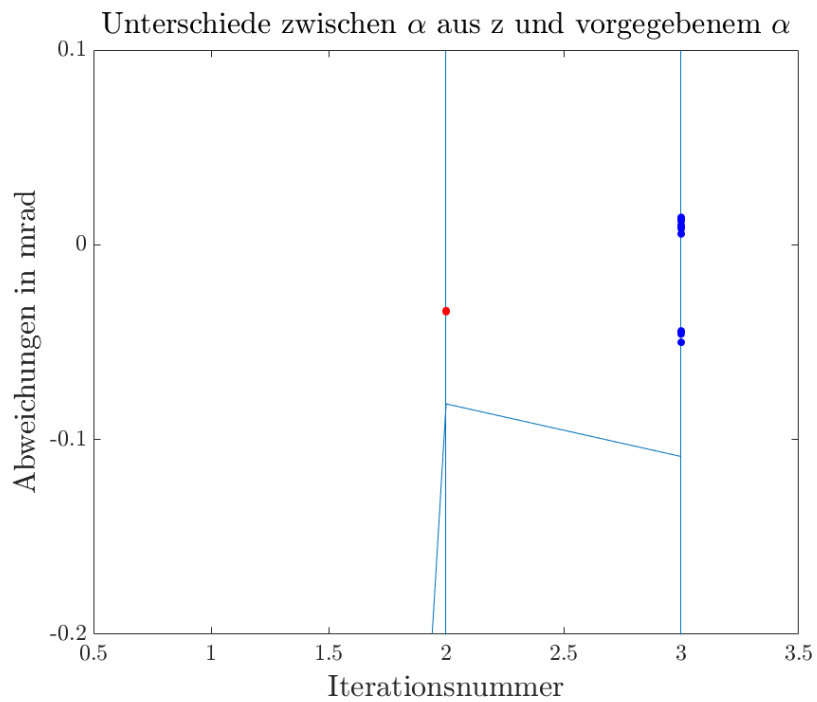


Abbildung A.9: Weiter eingezoomte Abweichung zwischen  $\alpha$  aus z und dem vorgegebenen  $\alpha$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

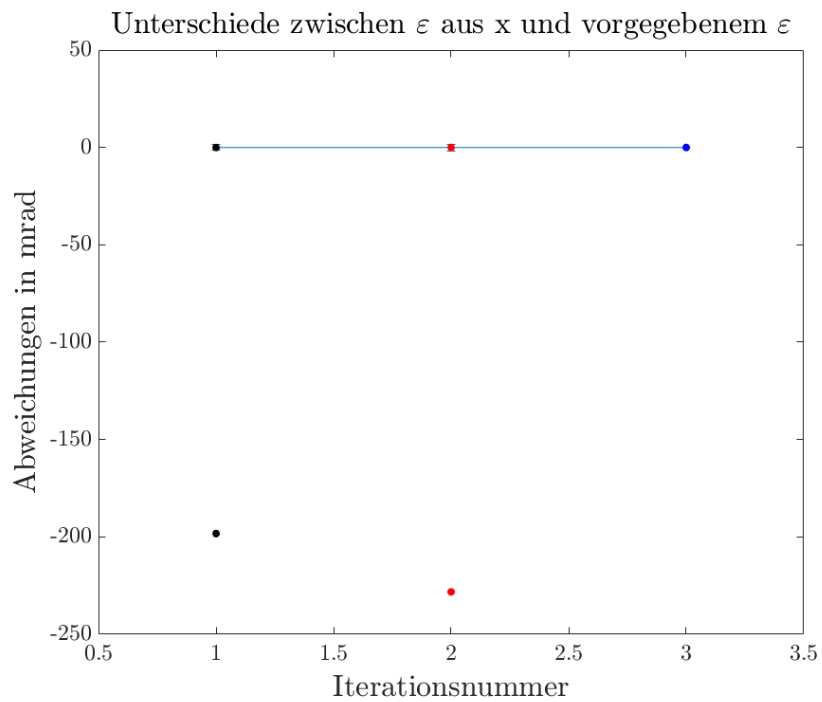


Abbildung A.10: Abweichung zwischen  $\varepsilon$  aus x und dem vorgegebenen  $\varepsilon$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

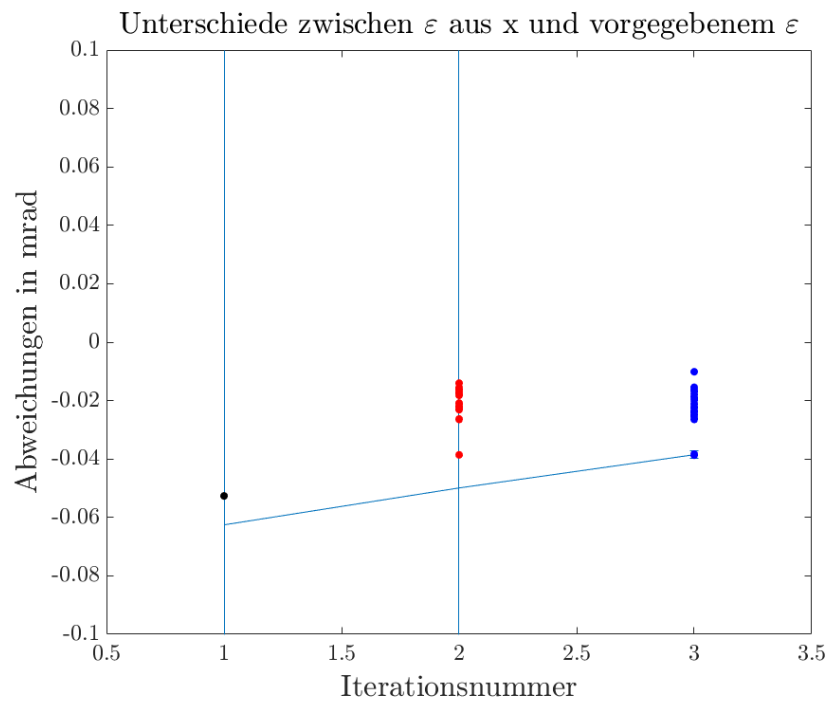


Abbildung A.11: Eingezoomte Abweichung zwischen  $\varepsilon$  aus x und dem vorgegebenen  $\varepsilon$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

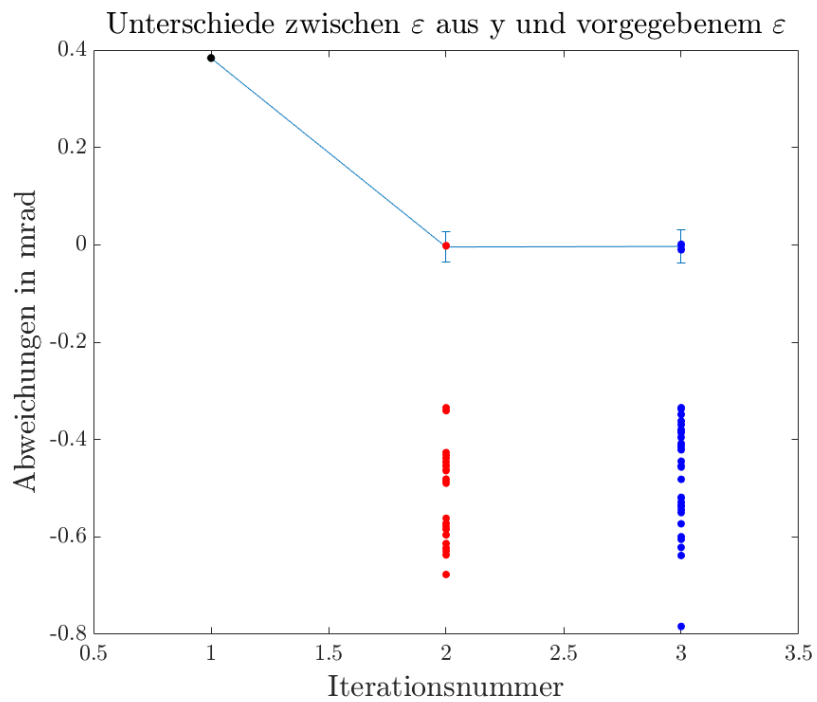


Abbildung A.12: Abweichung zwischen  $\varepsilon$  aus y und dem vorgegebenen  $\varepsilon$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

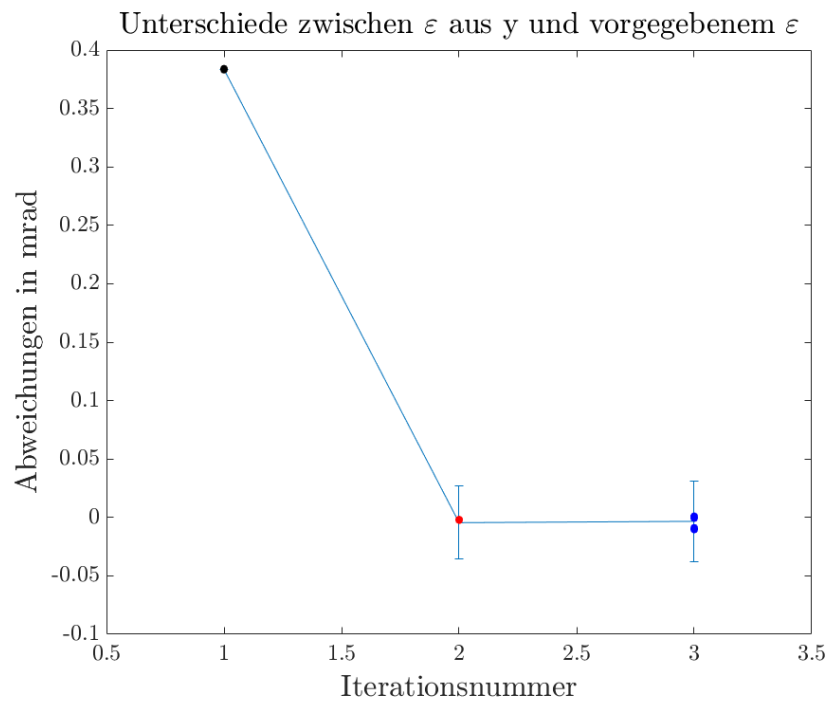


Abbildung A.13: Eingezoomte Abweichung zwischen  $\varepsilon$  aus y und dem vorgegebenen  $\varepsilon$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

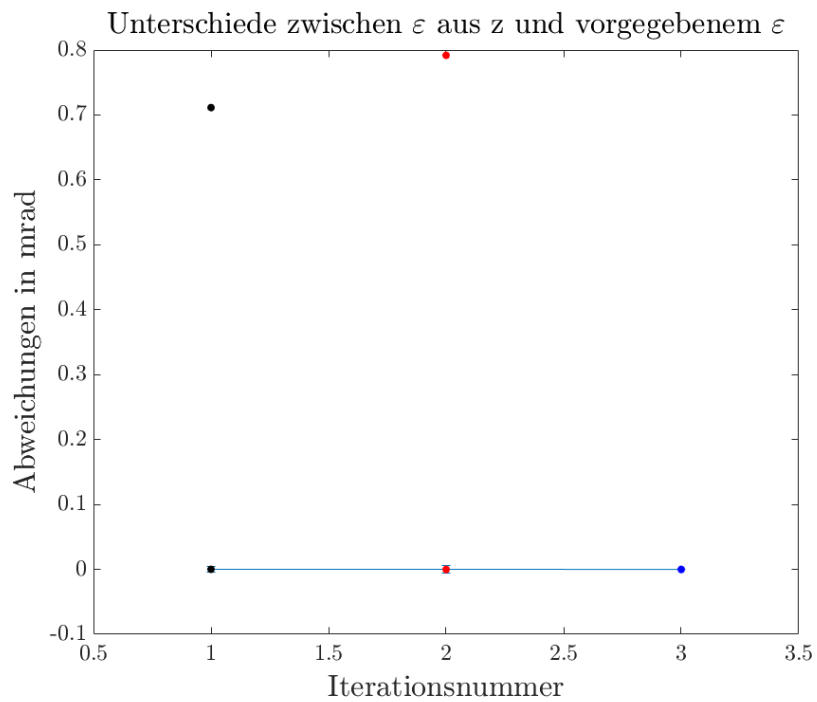


Abbildung A.14: Abweichung zwischen  $\varepsilon$  aus z und dem vorgegebenen  $\varepsilon$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung



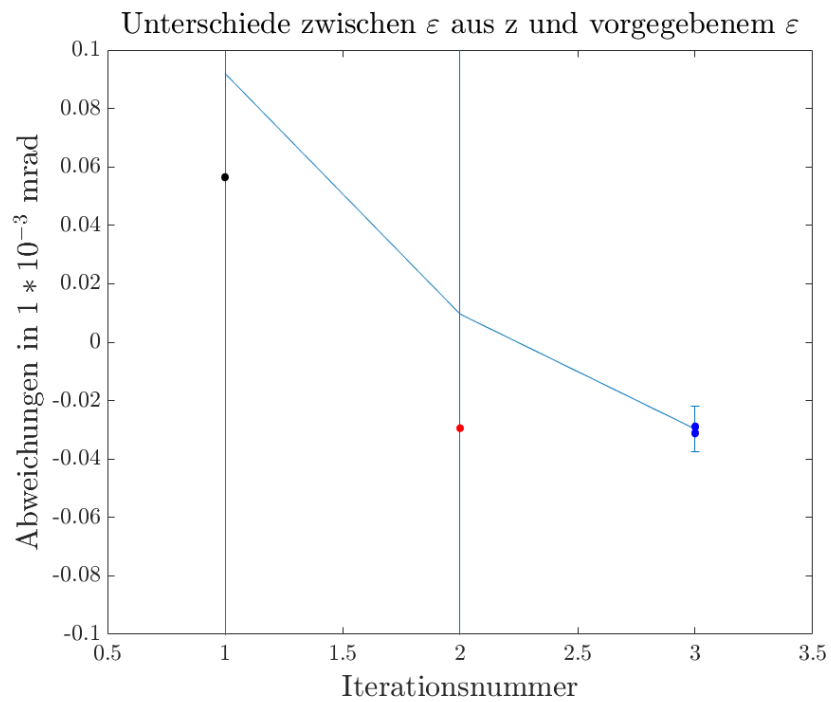


Abbildung A.15: Eingezoomte Abweichung zwischen  $\varepsilon$  aus z und dem vorgegebenen  $\varepsilon$ , berechnet mit  $\alpha = H_{yaw} - 15^\circ$  und  $\varepsilon = F_{Pitch} - 5^\circ$  für jede Iteration. Dargestellt sind Werte aus 20000 Durchführungen der Gesamtrechnung, mit eingezeichneter Standardabweichung

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	-0,063	5,69	198	1,363	1,402	86,4	-141,41	11,80
2	-0,0499	8,95	228	1,391	1,614	95,9	-141,41	10,56
3	-0,0385	8,14	0,029	1,175	0,0013	91,5	11,35	11,06

Tabelle A.1: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe der 20000 berechneten Werte des Heliostaten BE48. Werte anhand der x-Achse berechnet.  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	0,3835	13,92	0,0009	1,017	0,00005	124,6	57,96	7,91
2	-0,0045	16,3	0,6753	1,024	0,0312	127,8	-12,74	7,69
3	-0,0033	15,26	0,7849	1,024	0,0346	123,8	-11,35	7,95

Tabelle A.2: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe der 20000 berechneten Werte des Heliostaten BE48. Werte anhand der y-Achse berechnet.  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	0,092	-2,08	711,30	0,0377	5,03	0,266	141,41	141,41
2	0,0096	-0,0082	792,06	0,1007	5,6	0,9	141,41	50,52
3	-0,0297	-0,011	0,175	0,0101	0,0077	0,68	-11,34	-10,56

Tabelle A.3: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe der 20000 berechneten Werte des Heliostaten BE48. Werte anhand der z-Achse berechnet.  $\alpha = H_{yaw} - 15^\circ$ ,  $\varepsilon = F_{Pitch} - 5^\circ$

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	$-1 * 10^{-7}$	6,7	$2 * 10^{-5}$	871	$2 * 10^{-7}$	70,9	-50,9	10,57
2	$-6 * 10^{-5}$	7,3	0,013	889	$7 * 10^{-4}$	74,5	-11,12	10,02
3	-0,004	7,9	39,3	1187	0,393	77,4	-99,98	9,615

Tabelle A.4: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe der 20000 berechneten Werte des Heliostaten BE48. Werte anhand der x-Achse berechnet.  $\alpha = H_{yaw}$ ,  $\varepsilon = F_{Pitch}$

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	$-5 * 10^{-8}$	9,96	$4 * 10^{-4}$	556,3	$3 * 10^{-6}$	73,7	50,9	7,27
2	0,0012	10,05	0,2345	568,99	0,012	74,3	11,12	7,22
3	-0,153	8,87	1546,9	568,3	15,47	69,8	-99,98	7,71

Tabelle A.5: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe der 20000 berechneten Werte des Heliostaten BE48. Werte anhand der y-Achse berechnet.  $\alpha = H_{yaw}$ ,  $\varepsilon = F_{Pitch}$

Positions- Iteration	$\overline{\Delta\varepsilon}$ /mrad	$\overline{\Delta\alpha}$ /mrad	$R(\Delta\varepsilon)$ /mrad	$R(\Delta\alpha)$ /rad	$\sigma(\Delta\varepsilon)$ /mrad	$\sigma(\Delta\alpha)$ /mrad	$v(\Delta\varepsilon)$ /mrad	$v(\Delta\alpha)$ /rad
1	$-1 * 10^{-7}$	$-5 * 10^{-8}$	$8 * 10^{-5}$	$3 * 10^{-4}$	$8 * 10^{-7}$	$3 * 10^{-6}$	50,89	-75,4
2	$2 * 10^{-4}$	$-5 * 10^{-2}$	0,049	6,5	0,0026	0,532	11,12	-10,57
3	0,014	$-3 * 10^{-2}$	135,1	278,9	1,35	2,8	99,98	93,9

Tabelle A.6: Der Durchschnitt, die Spannweite, die Standardabweichung und die Schiefe der 20000 berechneten Werte des Heliostaten BE48. Werte anhand der z-Achse berechnet.  $\alpha = H_{yaw}$ ,  $\varepsilon = F_{Pitch}$

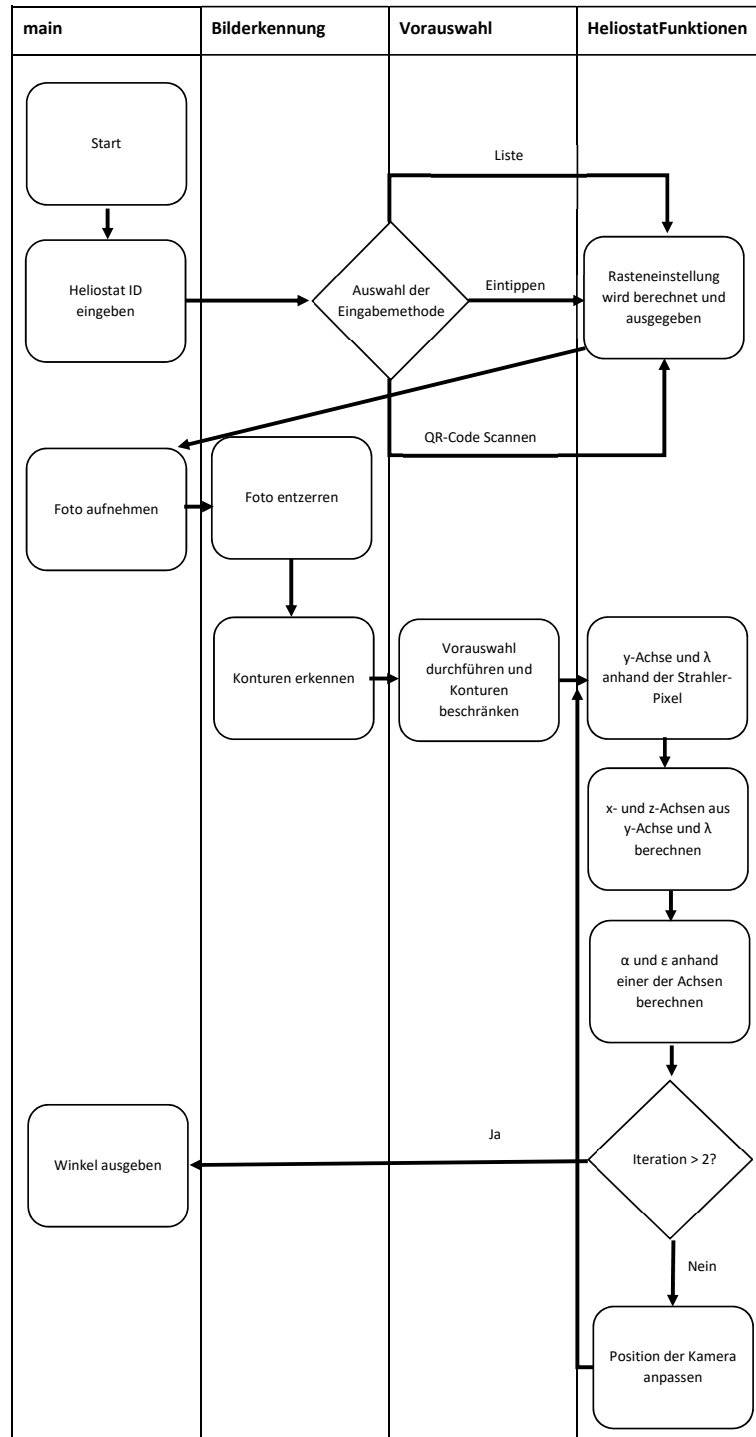


Abbildung A.16: Swimmlane Diagramm des Programmablaufs. Die Spalten stellen die verschiedenen Python Dateien dar, welche durch das Programm aufgerufen werden