

---

Masoud Roschani

# Probabilistische Planungsverfahren für die deflektometrische Oberflächeninspektion

Schriftenreihe Automatische Sichtprüfung und Bildverarbeitung | Band 16



Masoud Roschani

**Probabilistische Planungsverfahren für die  
deflektometrische Oberflächeninspektion**

Schriftenreihe Automatische Sichtprüfung und Bildverarbeitung  
**Band 16**

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Lehrstuhl für Interaktive Echtzeitsysteme  
am Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik  
und Bildauswertung IOSB



# **Probabilistische Planungsverfahren für die deflektometrische Oberflächeninspektion**

von  
Masoud Roschani

Karlsruher Institut für Technologie  
Lehrstuhl für Interaktive Echtzeitsysteme

Probabilistische Planungsverfahren für die  
deflektometrische Oberflächeninspektion

Zur Erlangung des akademischen Grades Doktor der Ingenieur-  
wissenschaften von der KIT-Fakultät für Informatik des Karlsruher  
Instituts für Technologie (KIT) genehmigte Dissertation

von Dipl.-Inform. Masoud Roschani

Tag der mündlichen Prüfung: 19. November 2018  
Referent: Prof. Dr.-Ing. habil. Jürgen Beyerer  
Korreferent: Prof. Dr.-Ing. habil. Uwe D. Hanebeck

#### Impressum



Karlsruher Institut für Technologie (KIT)  
KIT Scientific Publishing  
Straße am Forum 2  
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark  
of Karlsruhe Institute of Technology.  
Reprint using the book cover is not allowed.

[www.ksp.kit.edu](http://www.ksp.kit.edu)



*This document – excluding the cover, pictures and graphs – is licensed  
under a Creative Commons Attribution-Share Alike 4.0 International License  
(CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.en>*



*The cover page is licensed under a Creative Commons  
Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0):  
<https://creativecommons.org/licenses/by-nd/4.0/deed.en>*

Print on Demand 2019 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1866-5934  
ISBN 978-3-7315-0907-3  
DOI 10.5445/KSP/1000091802





# Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit als Wissenschaftlicher Mitarbeiter am Lehrstuhl für Interaktive Echtzeitsysteme (IES) des Karlsruher Instituts für Technologie (KIT).

Zunächst gilt mein besonderer Dank Herrn Professor Dr.-Ing. Jürgen Beyerer für die Möglichkeit diese Arbeit unter seiner Leitung durchzuführen und für die fundierte Betreuung sowie die wissenschaftliche und methodische Unterstützung.

Ich möchte mich auch bei meinen Kollegen am Lehrstuhl IES, von denen ich hier stellvertretend Christian Frese, Gaby Gross, Sebastian Höfer, Marco Huber, Mahsa Mohammadi Kaji, Alexey Pak, Miro Taphanel und Mathias Ziebarth erwähnen möchte, für die angenehme Atmosphäre und die interessanten Diskussionen bedanken.

Zu guter Letzt danke ich meiner Familie für die uneingeschränkte Unterstützung und Motivierung während meines Studiums und meiner Promotion.

Karlsruhe, im August 2019

*Masoud Roschani*



# Kurzfassung

Die deflektometrische Oberflächeninspektion beschäftigt sich mit der Prüfung spiegelnder Oberflächen nach dem Prinzip der Deflektometrie. Mittels einer Kamera wird das Spiegelbild eines Musters, das über eine Musteranzeige angezeigt wird, beobachtet. Beeinflusst durch die Form der Oberfläche, ist das beobachtete Spiegelbild verzerrt und erlaubt eine Detektion von Defekten. Wird anstatt eines einzelnen Musters eine Folge von Mustern angezeigt, die jeden Ort auf dem Schirm eindeutig kodieren, lässt sich weiter auf die Neigungen der Oberfläche schließen. Ein Punkt auf der Oberfläche wird vom Sensor beobachtet, wenn Lichtstrahlen vom Bildschirm über die Oberfläche in den Kamerasensor reflektiert werden. Der Sichtbereich einer einzelnen Messung hängt demnach im höchsten Maße von der relativen Lage des Sensors zum Prüfobjekt und von der Form des Prüfobjektes selbst ab. Diese Abhängigkeit gilt ebenfalls für weitere für die Inspektion relevante Größen wie die Ortsauflösung und die Neigungsempfindlichkeit. Für große oder komplex geformte Oberflächen müssen außerdem mehrere Messungen ausgeführt werden, um die Oberfläche vollständig zu erfassen. Die manuelle Wahl der einzelnen Sensorkonfigurationen, das heißt der Positionier- und Orientierungsparameter für den Sensor, ist aus den oben genannten Gründen nicht trivial und aufwändig.

In dieser Arbeit werden Verfahren zur automatischen Bestimmung der Sensorkonfiguration untersucht. Das Problem der Oberflächeninspektion wird in ein probabilistisches Rahmenwerk formalisiert, in dem Messunsicherheit und Unwissen über die Oberfläche einheitlich beschrieben werden können.

Während der Planung müssen die einzelnen Messungen in ein gemeinsames Oberflächenmodell fusioniert werden. Es werden zwei unterschiedliche Oberflächenmodelle untersucht. Die Oberfläche wird zum einen als ein Gauß-Prozess modelliert. Der Gauß-Prozess kann als eine Verallgemeinerung der Normalverteilung auf Funktionen aufgefasst werden. Sie ist durch eine Erwartungswertfunktion und eine Kovarianzfunktion eindeutig bestimmt. Aus der Kovarianzfunktion lässt sich ein Unsicherheitsmaß für die Oberfläche extrahieren. In einer weiteren Repräsentation wird anstatt der Höhe der Oberfläche nur die Neigung modelliert. Die Oberfläche wird als eine Menge von kleinen Spiegeln modelliert, wobei die Neigung jedes einzelnen Spiegels durch eine Wahrscheinlichkeitsverteilung gegeben ist. Durch die gemessenen Normalen wird die Verteilung der Spiegel aktualisiert. Weiterhin werden zwei Planungsverfahren untersucht. Bei der Greedy-Planung wird das Finden eines Plans mit einer Sequenz von Optimierungsproblemen approximiert, die jeweils nur für einen Zeitschritt im Voraus planen. Die Sweep-Planung nutzt ein zweistufiges Optimierungsverfahren zur Berechnung eines Plans. Im äußeren Prozess wird ein Plan mit einer festen Anzahl an Messungen gesucht. Dabei wird der innere Prozess verwendet, der bei fester Anzahl an Messungen in mehreren Durchläufen jeweils eine Sensorkonfiguration auswählt und diese optimiert. Zur Optimierung der einzelnen Sensorkonfigurationen wird das Simulated-Annealing-Verfahren verwendet. Hierzu wird die Pertubationsfunktion für die deflektometrische Anwendung angepasst. Schließlich werden die Planungsverfahren in simulationsbasierten Experimenten evaluiert und an einer realen Oberfläche angewandt.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Deflektometrie	2
1.2	Problemstellung	4
1.3	Stand der Forschung	5
1.3.1	Planung in der Deflektometrie	5
1.3.2	Sensoreinsatzplanung	6
1.4	Probabilistische Oberflächenrepräsentation	10
1.5	Beiträge zum Stand der Forschung	11
1.6	Gliederung der Arbeit	12
<b>2</b>	<b>Rahmenwerk für die probabilistische Planung in der Oberflächeninspektion</b>	<b>13</b>
2.1	Markow-Entscheidungsprozess (MDP)	14
2.2	Teilweise beobachtbarer Markow-Entscheidungsprozess	15
2.2.1	Belief-MDP	18
2.2.2	Strategie	19
2.3	Probabilistische Modellierung der Oberflächeninspektion	20
2.3.1	Trennung der Informationsbeschaffung und der Prüfaufgabe	23
2.4	Unsicherheitsbestimmung	26
2.5	Lösungsverfahren	27
2.6	Zusammenfassung	29

<b>3</b>	<b>Sensormodell</b>	<b>31</b>
3.1	Messmodell	31
3.1.1	Komponenten des Messmodells	32
3.1.2	Approximation des Messmodells	34
3.2	Kameramodell	35
3.2.1	Modellbeschreibung	36
3.2.2	Geometrische Abbildung	37
3.2.3	Schärfentiefe	39
3.3	Referenzoberflächenmodell	41
3.4	Schirmmodell	43
3.4.1	Geometrische Modellierung	43
3.4.2	Weitere Einflussgrößen	44
3.5	Sensoraufbauten und Konfiguration	45
3.6	Geometrisches Modell der deflektometrischen Messung	50
3.6.1	Zuordnung zwischen Kamerapixel und Schirmpixel	51
3.6.2	Bestimmung der Neigung	52
3.7	Deflektometrische Registrierung	54
3.7.1	Phasenschiebeverfahren	54
3.7.2	Phasenfaltung	56
3.8	Unsicherheit	56
3.8.1	Prädiktion der Phasenunsicherheit	57
3.8.2	Prädiktion der Neigungsunsicherheit	65
3.9	Simulator	66
3.10	Zusammenfassung	70
<b>4</b>	<b>Oberflächenmodelle und Fusion</b>	<b>71</b>
4.1	Überblick über Oberflächenmodelle	72
4.2	Regression	74
4.3	Gauß-Prozess Modell	75
4.3.1	Gauß-Prozess	76

4.3.2	Prior . . . . .	78
4.3.3	Auswirkung eines Operators auf den Gauß-Prozess	80
4.3.4	Notation . . . . .	81
4.3.5	Inferenz . . . . .	83
4.3.6	Approximationsverfahren . . . . .	86
4.3.7	Inferenz bei unbekanntem Referenzmodell und bekannten Regularisierungspunkten . . . . .	93
4.3.8	Gütekriterien . . . . .	95
4.3.9	Modellauswahl . . . . .	98
4.4	Diskretisiertes Modell . . . . .	100
4.4.1	Ebene Spiegel . . . . .	102
4.5	Zusammenfassung . . . . .	108
<b>5</b>	<b>Planungsverfahren . . . . .</b>	<b>109</b>
5.1	Nebenbedingungen . . . . .	111
5.2	Optimierungsalgorithmen . . . . .	112
5.2.1	Simulated-Annealing . . . . .	114
5.2.2	Nelder-Mead . . . . .	122
5.3	Greedy-Planung . . . . .	123
5.3.1	Greedy-Planung mit Teildiskretisierung . . . . .	124
5.4	Sweep-Planung . . . . .	128
5.5	Zusammenfassung . . . . .	132
<b>6</b>	<b>Evaluation der Planungsverfahren . . . . .</b>	<b>135</b>
6.1	Testaufbau der simulationsbasierten Experimente . . . . .	135
6.1.1	Durchführung der simulationsbasierten Inspektion	139
6.2	Gütemaß des Gauß-Prozess Modells . . . . .	140
6.3	Planungsverfahren anhand des Gauß-Prozess Modells . . .	142
6.4	Planungsverfahren anhand des diskreten Modells . . . . .	145
6.5	Approximationsmöglichkeit . . . . .	149
6.6	Evaluation an einer realen Oberfläche . . . . .	154

6.6.1	Aufbau . . . . .	157
6.6.2	Kalibrierung . . . . .	157
6.6.3	Nebenbedingungen . . . . .	164
6.6.4	Durchführung und Ergebnisse der Planung . . . . .	165
6.7	Zusammenfassung . . . . .	170
<b>7</b>	<b>Zusammenfassung und Ausblick . . . . .</b>	<b>177</b>
7.1	Zusammenfassung . . . . .	177
7.2	Ausblick . . . . .	179
<b>A</b>	<b>Anhang . . . . .</b>	<b>183</b>
A.1	Anhang 1 . . . . .	183
A.1.1	Marginalisierung . . . . .	183
A.1.2	Bedingte Dichte . . . . .	184
A.1.3	Verbundverteilung . . . . .	184
A.1.4	Inferenz . . . . .	185
A.2	Anhang 2 . . . . .	186

# Abkürzungs- und Symbolverzeichnis

## Abkürzung

AUV	<u>autonomous underwater vehicle</u>
GPU	<u>graphical processing unit</u>
LCD	<u>liquid crystal display</u>
MDP	<u>Markov Decision Process</u>
Pkw	<u>Personenkraftwagen</u>
POMDP	<u>Partial Observable Markov Decision Process</u>

## Generelle Notation

$a, b, c, \dots$	skalare Größe
$\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$	vektorielle Größe
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$	Matrix
$\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}, \dots$	normalisierter Vektor
$\underline{a}, \underline{b}, \underline{c}, \dots$	Zufallsgröße
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$	Koordinatensystem
${}^{\mathcal{A}}\mathbf{H}_{\mathcal{B}}$	Lage des Koordinatensystems $\mathcal{B}$ in homogenen Koordinaten relativ zum Koordinatensystem $\mathcal{A}$
${}^{\mathcal{A}}\mathbf{p}$	Koordinaten des Vektors $\mathbf{p}$ relativ zum Koordinatensystem $\mathcal{A}$
$\det \mathbf{C}$	Determinante der Matrix $\mathbf{C}$
$\text{Tr } \mathbf{C}$	Spur der Matrix $\mathbf{C}$

### Koordinatensysteme

$\mathcal{B}$	Sensorkoordinatensystem
$\tilde{\mathcal{B}}$	Projiziertes Koordinatensystems des Kamerakoordinatensystems auf die Bildebene
$\mathcal{C}$	Kamerakoordinatensystem
$\mathcal{D}$	Bildschirmkoordinatensystem
$\mathcal{F}$	Flanschkoordinatensystem
$\mathcal{O}$	Objektkoordinatensystem
$\mathcal{R}$	Roboterbasiskoordinatensystem
$\mathcal{W}$	Weltkoordinatensystem

### Alphabetisch

$\mathcal{A}$	Die Menge der Aktionen in einem MDP oder POMDP
$\mathcal{A}_I$	Aktionen zur Einstellung des Sensors ( $\mathcal{A}_I \simeq \Theta$ )
$\mathcal{A}_T$	Terminalaktionen
$a_k$	Aktion zum Zeitpunkt $k$
$\mathcal{B}$	Die Menge der Beliefstates in einem Beliefstate-MDP
$B(b_k, a_k, z_k)$	Transitionsfunktion des Beliefstates
$b$	Bildweite (Abschnitt 3.2)
$b_k$	Beliefstate
$\text{Cov} \left\{ \underline{x}, \underline{y} \right\}$	Kovarianz zwischen zwei Zufallsvariablen $\underline{x}$ und $\underline{y}$
$c(\mathcal{S}, a)$	Kosten der Prüfentscheidung
$d$	Dimension der Sensorkonfiguration
$D$	Durchmesser der kreisförmigen Eintrittspupille
$d_c$	Abstand der Kamera von der Oberfläche
$d_l$	Abstand des Schirms von der Oberfläche
$d_u$	Breite des Pixels
$d_v$	Höhe des Pixels
$E \left\{ \underline{x} \right\}$	Erwartungswert einer Zufallsvariable $\underline{x}$
$\mathcal{E}_D$	Schirmebene

$E^d$	$d$ -dimensionaler Euklidischer Raum, hier mit $d = 2$ oder $d = 3$
$\tilde{\mathcal{E}}_C$	Ideale Bildebene der Kamera
$\mathcal{E}_C$	Bildebene der Kamera
$f$	Oberfläche, repräsentiert als Funktion
$f$	Brennweite (Abschnitt 3.2)
$f_r(\mathbf{x})$	Referenzoberfläche (Darstellung als Funktion)
$g$	Gegenstandsweite (Abschnitt 3.2.3)
$g(\mathcal{H}_k, a_k)$	Schrittkostenfunktion eines POMDP
$g(x_k, a_k)$	Schrittkostenfunktion eines MDPs
$g_a(\boldsymbol{\theta}, a)$	Kostenfunktion für den Messaufwand
$\mathcal{GP}(\mu, k)$	Gauß-Prozess mit Erwartungswertfunktion $\mu$ und Kovarianzfunktion $k$
$g_{\text{sat}}$	Maximaler Grauwert (Abschnitt 3.7.1)
$g_u(b_k, a_k)$	Negativer erwarteter Informationsgewinn
$h(\mathcal{S}, \boldsymbol{\theta}, \underline{\mathcal{I}}, \underline{\mathbf{w}})$	Mengenwertige Funktion $h$ , beschreibt das deflektometrische Messmodell
$h_n(\mathcal{S}_r, \underline{\mathbf{L}}, \boldsymbol{\theta})$	Funktion zur Berechnung der Normalen
$h_n(\mathcal{S}_r, \underline{\mathbf{L}}, \boldsymbol{\theta}, u, v)$	Funktion zur Berechnung der Normalen und zugehöriger Messstelle, die dem Kamerapixel $(u, v)$ zugeordnet wird.
$h_n^{(1)}(\mathcal{S}_r, \underline{\mathbf{L}}, \boldsymbol{\theta}, u, v)$	Funktion zur Berechnung der Messstelle, die dem Kamerapixel $(u, v)$ zugeordnet wird.
$h_n^{(2)}(\underline{\mathbf{L}}, \boldsymbol{\theta}, \mathcal{S}_r, u, v)$	Funktion zur Berechnung der Normalen, die dem Kamerapixel $(u, v)$ zugeordnet wird.
$h_p(\mathcal{S}, \boldsymbol{\theta}, \underline{\mathcal{I}}, \underline{\mathbf{w}}_p)$	Physikalisches deflektometrisches Messmodell
$h_r(\underline{\mathcal{I}}, \underline{\tilde{\mathcal{L}}})$	Funktion zur Berechnung der deflektometrischen Registrierung
$h_s(\mathcal{S}, \boldsymbol{\theta}, \underline{\mathbf{w}}_s)$	Approximatives deflektometrisches Messmodell
$h_s(\mathcal{S}, \boldsymbol{\theta}, \underline{\mathbf{w}}_s, u, v)$	Funktion zur Berechnung des Schirmpixels, das dem Kamerapixel $(u, v)$ zugeordnet wird.

$\mathcal{H}_k$	Historie zum Zeitpunkt $k$
$\mathcal{I}$	Mustersequenz für die Kodierung des Bildschirms
$\tilde{\mathcal{I}}$	Aufgenommene Spiegelbilder der Mustersequenz
$J(b_0, \pi)$	Gesamtkosten der Strategie $\pi$
$J_n(b_0, \theta_{0:n-1})$	Gesamtkosten eines Plans
$K$	Verstärkungsfaktor (Abschnitt 3.8.1)
$k(\mathbf{x}_1, \mathbf{x}_2)$	kubische B-Spline Kovarianzfunktion
$\mathbf{K}_{XX}$	Kovarianzmatrix der Trainingsdaten
$k(\mathbf{x}_1, \mathbf{x}_2)$	Kovarianzfunktion eines Gauß-Prozesses
$\tilde{\mathbf{I}}(\mathbf{r})$	Deflektometrische Registrierung
$\mathbf{I}(u, v)$	Deflektometrische Registrierung in Schirmkoordinaten
$L_u$	Länge des Bildschirms
$L_v$	Höhe des Bildschirms
$M$	Länge der angezeigten Mustersequenz
$M$	Maximale Anzahl der sich wiederholenden Phasen $n$ einem Muster
$\mathcal{M}_S$	Menge der Oberfläche
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Normalverteilung mit Erwartungswert $\boldsymbol{\mu}$ und Kovarianz $\boldsymbol{\Sigma}$
$n$	Anzahl der Messungen
$\hat{\mathbf{n}}_S(\mathbf{r}_s)$	Normalenvektor am Punkt $\mathbf{r}_s$
$\mathbf{n}_D$	Normale der Schirmebene
$\mathcal{P}$	Partition des Sensorkonfigurationsraumes
$p_0(x_0)$	A-priori-Verteilung des Zustands
$p(z_{k+1}   x_k, a_k)$	Die Beobachtungswahrscheinlichkeit
$p_{\text{prop}}$	Wahrscheinlichkeitsdichtefunktion zur Perturbation im Simulated-Annealing-Verfahren
$p(x_{k+1}   x_k, a_k)$	Transitionswahrscheinlichkeit eines MDPs
$p_u$	Wahrscheinlichkeitsdichtefunktion zum Ziehen von Stellen mit hoher Unsicherheit



$q(\mathbf{r}, f_r, \mathbf{l}, \boldsymbol{\theta})$	Funktion zur Berechnung der Gradienten
$\mathcal{R}$	Menge aller Regeln aus der die Pertubationsfunktion zusammengesetzt ist
$\mathbf{R}_i(\phi)$	Homogene Rotationsmatrix um die Einheitsachse $\mathbf{e}_i$ um einem Winkel $\phi$
$\mathcal{S}$	Oberfläche
$\mathcal{S}_r$	Referenzoberfläche
$\hat{\mathbf{s}}$	Sichtstrahl von der Kamera zur Oberfläche
$s_3(x)$	Eindimensionale kubische Basisfunktion
$\text{SE}(3)$	Menge aller Rotationen gefolgt von einer Translation
$\hat{\mathbf{s}}_r$	Reflektierter Sichtstrahl
$\mathbf{T}f$	Operator angewandt auf eine Funktion $f$
$\mathbf{T}_i(t)$	Homogene Translationsmatrix entlang der Einheitsachse $\mathbf{e}_i$ um den Betrag $t$
$\mathbf{T}^{(1)}f, \mathbf{T}^{(2)}f$	Operator $\mathbf{T}$ angewendet auf das erste und zweite Argument einer Funktion $f$
$u$	Bildschirmspalte (3.8.1)
$\mathbf{U}$	Sequenz der induzierten Punkte
$U(b_k)$	Funktion zur Extraktion der Unsicherheit
$U(\mathbf{x})$	Unsicherheitskarte, gibt die Unsicherheit (Varianz) an jedem Punkt der Schätzung an.
$U_D(\mathbf{C})$	Determinante der Kovarianzmatrix $\mathbf{C}$
$U_E(\mathbf{C})$	maximaler Eigenwert der Kovarianzmatrix $\mathbf{C}$
$U_M(\mathbf{C})$	maximales Diagonalelement der Kovarianzmatrix $\mathbf{C}$
$U_{\text{suff}}$	Gewünschte maximale Unsicherheit für die Terminierung der Informationsbeschaffung.
$U_T(\mathbf{C})$	Spur der Kovarianzmatrix $\mathbf{C}$
$(u, v)$	Pixelkoordinaten auf der Kamera
$\underline{\mathbf{w}}$	Rauschvektor des deflektometrischen Messmodells
$\mathbf{W}$	Beobachtungsrauschmatrix

$W_n$	Neigungsunsicherheit
$\underline{w}_p$	Rauschvektor des physikalischen Messmodells
$W_s$	Unsicherheit der deflektometrischen Registrierung
$\underline{w}_s$	Rauschvektor des approximativen deflektometrischen Messmodells
$\mathcal{X}$	Die Menge der Zustände in einem MDP oder POMDP
$x_k$	Zustand zum Zeitpunkt $k$
$X_\star$	Sequenz der zu evaluierenden Stellen
$\underline{z}$	Messstellen und zugehörige Neigungen aus einer deflektometrischen Messung

### Griechisch

$\alpha_1$	Rotation des Schirms um seine y-Achse Eulerwinkel
$\beta$	Relative Helligkeit des gemessenen Musters
$\bar{\beta}$	Relative Helligkeit des angezeigten Musters
$\beta_1$	Rotation des Schirms um seine x-Achse Eulerwinkel
$\chi$	Sensorkonfigurationsparameter, der den Sichtpunkt der Kamera auf der Oberfläche beschreibt
$\delta(x)$	Delta-Distribution
$\delta_x$	Evaluationsfunktional an der Stelle $x$
$\epsilon$	Radius der Unschärfescheibe
$\eta(\lambda_p)$	Quanteneffizienz
$\gamma$	Relativer Kontrast des gemessenen Musters
$\bar{\gamma}$	Relativer Kontrast des angezeigten Musters
$\gamma_1$	Rotation des Schirms um seine z-Achse Eulerwinkel
$\lambda$	Gewichtungsfaktor zwischen Unsicherheitsreduktion und weiteren Kosten einer Messung
$\mu_a$	A-posteriori Erwartungswert der Funktionswerte an den Evaluierungsstellen, berechnet mittels approximierter Gauß-Prozess Inferenz

$\mu_g$	A-posteriori Erwartungswert der Funktionswerte an den Evaluierungsstellen, berechnet mittels exakter Gauß-Prozess Inferenz
$\mu_n$	A-posteriori Erwartungswert der Neigung für eine Zelle im diskreten Modell
$\mu_d$	Dunkelrauschen
$\mu_e$	Durchschnittliche Anzahl an freigesetzten Elektronen
$\mu_{e,sat}$	Sättigungskapazität
$\mu_g$	Durchschnittlicher Grauwert
$\mu_{g,dark}$	erwarteter Grauwert aufgrund des Dunkelrauschens
$\mu_p$	Durchschnittliche Anzahl von Photonen
$\mu(\mathbf{x})$	Erwartungswertfunktion eines Gauß-Prozesses
$\nabla_x$	Gradient an der Stelle $\mathbf{x}$
$\Omega$	Die Menge der Beobachtungen eines POMDP
$\phi_c$	Azimuthwinkel der Kamera
$\hat{\phi}$	Schätzung der Phase
$\phi_1$	Azimuthwinkel des Schirms
$\phi_{max}$	Schwellenwert für die Abweichung der Referenzneigung zur Neigung in der Zelle
$A_{max}$	Schwellenwert für Fläche einer Zelle
$\Phi(\mathbf{x})$	Basisfunktionen des approximierten Gauß-Prozesses
$\pi(b_k)$	Strategie eines Belief-MDP
$\Pi$	Projektor eines Objektpunktes auf die Bildebene
$\Pi_1$	Projektor eines Objektpunktes auf die idealisierte Bildebene
$\Pi_2$	Verzerrungsfunktion
$\Pi_3$	Transformationsfunktion von Bildpunkten auf der idealisierten Bildebene auf die Bildebene
$\pi_1$	Informationsbeschaffungsstrategie (Kapitel 2)

$\pi_T$	Prüfentscheidungsstrategie (Kapitel 2)
$\psi_k$	Phaseverschiebungen
$\rho_E(b_n)$	Endkosten
$\rho(b_k, a_k)$	Kostenfunktion eines $\rho$ POMDP
$\Sigma_a$	A-posteriori Kovarianzmatrix der Funktionswerte an den Evaluierungsstellen, berechnet mittels approximierter Gauß-Prozess Inferenz
$\Sigma_g$	A-posteriori Kovarianzmatrix der Funktionswerte an den Evaluierungsstellen, berechnet mittels exakter Gauß-Prozess Inferenz
$\sigma_e$	Standardabweichung der Anzahl freigesetzten Elektronen
$\sigma_g$	Standardabweichung des gemessenen Grauwertes
$\sigma_\phi$	Standardabweichung der gemessenen Phase
$\sigma_q$	Standardabweichung der Quantisierung des Grauwertes
$\Sigma_n$	A-posteriori Kovarianzmatrix der Neigung für eine Zelle im diskreten Modell
$\tau_c$	Rotationswinkel der Kamera um seine optische Achse
$\Theta$	Menge der Sensorkonfigurationen
$\theta$	Sensorkonfiguration
$\theta_{0:n-1}$	Plan
$\vartheta_c$	Elevationswinkel der Kamera
$\vartheta_1$	Elevationswinkel des Schirms
$\tilde{\theta}_k$	Sensorkonfiguration, Teil des Zustands in einer Oberflächeninspektionsaufgabe
$\zeta_i^*$	Die optimale Konfiguration des Sensorkonfigurationsraumes $\mathcal{P}_i$

# 1 Einleitung

Die automatische Oberflächeninspektion nimmt in der heutigen Industrie eine zunehmend bedeutende Rolle ein. Eine besondere Herausforderung sind spiegelnde Oberflächen, da sie nicht direkt sichtbar sind, sondern nur über die Reflexion der Umgebung beobachtet werden können. Dies führt dazu, dass herkömmliche für nichtspiegelnde Oberflächen angewandte Methoden, wie zum Beispiel projizierende Triangulationsverfahren in diesem Fall nicht herangezogen werden können. Die *Deflektometrie* ist ein Verfahren, das genau die spekulare Reflexion nutzt, um auf die Gestalt der Oberfläche zu schließen. Dazu erzeugt sie eine kontrollierte Umgebung, deren Spiegelbild über die Prüfoberfläche mittels einem Sensor beobachtet wird. Die Verformung der Spiegelbilder liefert die gewünschte Information, die anschließend für die entsprechende Inspektionsaufgabe ausgewertet werden muss. Anwendung für spiegelnde Oberflächen finden sich zum Beispiel bei solarthermischen Kraftwerken oder in der Automobilindustrie. In solarthermischen Kraftwerken bündeln Kollektoren das einfallende Sonnenlicht auf einen Absorber, um Wärme zu erzeugen. Schon kleine Abweichungen in der Form können hier zu einem niedrigen Wirkungsgrad führen. In der Automobilindustrie ist die Defektfreiheit der Karosserie ein wichtiger Faktor für die Verbraucher. Die Hersteller sind dementsprechend daran interessiert, Karosserieteile sofort nach der Lackierung auf sichtbare Defekte zu prüfen.

Spiegelnde Oberflächen werden meist noch von Menschen begutachtet. Dies ist zeitaufwendig und die repetitive Arbeit führt zur Ermüdung der Prüfer.

Durch eine Automatisierung können hier hohe Durchsatzraten und geringere Kosten erreicht werden.

## 1.1 Deflektometrie

Die Prüfung spiegelnder Oberflächen lässt sich mit den klassischen Verfahren wie dem *Shape from Shading*] [Zha99] oder *Streifenprojektionsverfahren* [Gen11] nicht bewerkstelligen, da diese Verfahren auf eine diffuse Reflexion angewiesen sind und spiegelnde Reflexionen zu Störungen führen. Ein geeignetes Verfahren hierfür ist die *Deflektometrie* [Bey15, Per97]. Sie funktioniert nach dem Prinzip, welches auch ein menschlicher Beobachter intuitiv zur Prüfung einer spiegelnden Oberfläche anwendet. Er schaut sich das Spiegelbild eines regelmäßigen Musters über die Prüfoberfläche an und versucht aus verschiedenen Blickwinkeln auf Defekte zu schließen. Die Anomalien werden im Spiegelbild des Musters durch Deformationen, wie Welligkeiten, starke Vergrößerung oder Verkleinerung sichtbar. Eine technische Umsetzung dieses Prinzips geschieht üblicherweise durch eine Kamera, die die Rolle des Beobachters einnimmt, und einen Mustergenerator, der eine kontrollierte Umgebung erzeugt. Für den Mustergenerator kann eine kostengünstige Flüssigkristallanzeige (engl. liquid crystal display, LCD) verwendet werden. Während der Bildschirm ein Muster anzeigt, beispielsweise ein Streifenmuster, beobachtet die Kamera das Spiegelbild des Musters. Abhängig von der Form der Oberfläche ist das beobachtete Muster verzerrt. Diese Information lässt bereits qualitative Aussagen über mögliche Defekte zu. Wird anstatt eines Musters eine Folge von Mustern gezeigt, mit dem Ziel jeden einzelnen Punkt auf dem Schirm eindeutig zu kodieren, so lässt sich eine Zuordnung von beobachteten Kamerapunkten und Schirmpunkten aufstellen und bei bekannter Geometrie auf die Neigung der Oberfläche am beobachteten Punkt schließen.

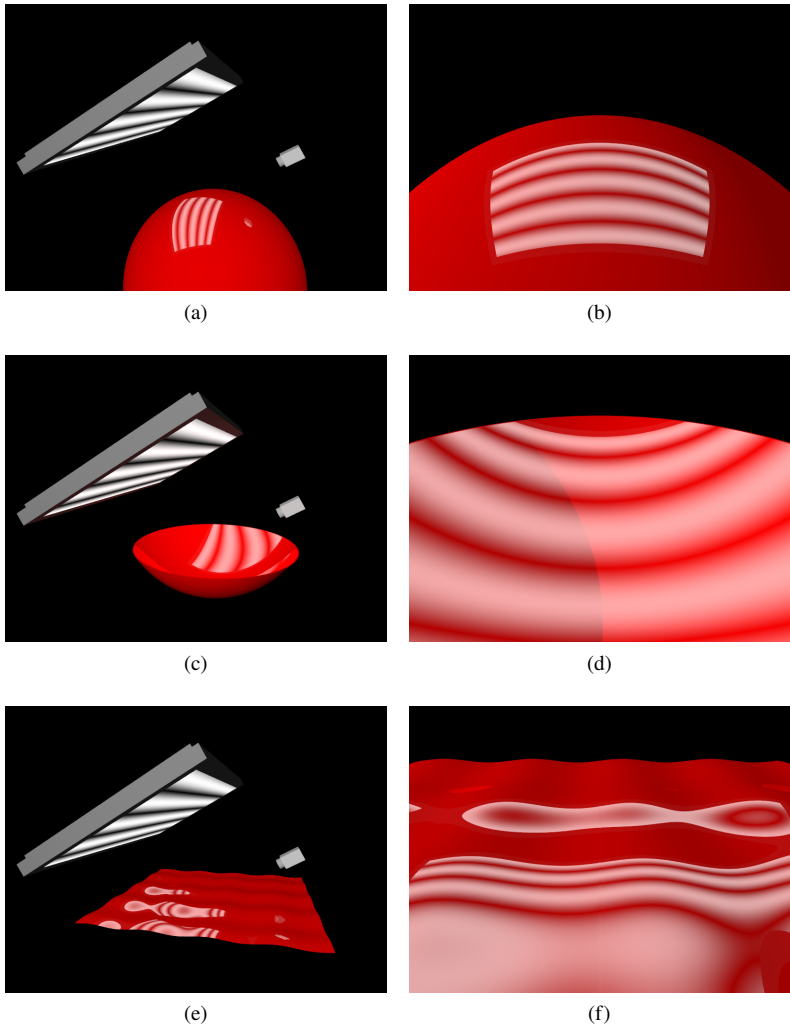


Abbildung 1.1: Abhängigkeit des Messbereichs in der Deflektometrie von der Oberfläche. (a), (c), (e) Darstellung der Szene mit unterschiedlichen Oberflächen. Kugel, Ausschnitt des Inneren einer Kugel und einer gewellten Oberfläche. (b), (d), (f) zugehöriges von der Kamera beobachtetes Bild.

Im Vergleich zu Streifenprojektionsverfahren ist die Deflektometrie besonders empfindlich gegenüber Neigungsänderungen im Gegensatz zu Höhenänderungen. Die Neigung an einem Oberflächenpunkt kann von einem deflektometrischen Sensorsystem nur gemessen werden, wenn Lichtstrahlen ungehindert vom Schirm über diesen Oberflächenpunkt zur Kamera gelangen können. Die Menge der Oberflächenpunkte, die das deflektometrische Sensorsystem messen kann, bestimmen den *Messbereich*. Der Messbereich in der Deflektometrie ist durch den Sichtbereich der Kamera und durch den Schirm beschränkt. Insbesondere aber beeinflusst die Oberfläche Größe und Form des Messbereichs. Beispielsweise werden die Sichtstrahlen von der Kamera aus betrachtet bei der Reflexion an einer konvexen Oberfläche expandiert. Dadurch kann nur ein relativ kleiner Bereich auf der Oberfläche vermessen werden. Dieser wird kleiner, je weiter der Schirm entfernt ist. Im Gegensatz dazu konvergieren die Sichtstrahlen, wenn die Oberfläche konkav ist. Bis zu einem gewissen Schirmabstand können große Bereiche des Objektes beobachtet werden. Außerdem kann der Messbereich abhängig von der Prüfoberfläche eine beliebig komplizierte Form annehmen. Er kann auch nicht zusammenhängend sein; siehe Abbildung 1.1. Die Ortsauflösung ist zum einen durch den Abstand der Kamera zum Objekt und zum anderen von der Schärfentiefe der Kamera abhängig. Wenn auf die Oberfläche fokussiert wird, kann eine hohe Ortsauflösung erreicht werden. Bei einer Fokussierung auf den Schirm kann die Unsicherheit, mit der die Normale gemessen wird, verringert werden. Wird der Abstand des Schirms vergrößert, lässt sich ebenfalls die Unsicherheit bezüglich der Normalen reduzieren.

## 1.2 Problemstellung

Häufig ist es nicht möglich, mit einer einzelnen Messung die Information über die Oberfläche zu erlangen, die für die Prüfung der Spezifikation benötigt wird. Dies gilt insbesondere für die Deflektometrie, bei der die zu



prüfende Oberfläche Teil der optischen Abbildung ist und die Messung auf komplizierte Weise beeinflusst, wie im vorherigen Abschnitt 1.1 diskutiert. Für die Prüfung eines Objekts muss die Sensorkonfiguration, wie die Lage des Sensors, die optischen Parameter der Kamera und die Anzahl der benötigten Messungen bestimmt werden. Eine manuelle Wahl, die zusätzlich zur gesamten Abdeckung des erforderlichen Bereichs der Prüfoberfläche auch die Einhaltung von Spezifikationen bezüglich Unsicherheit, Ortsauflösung und Messzeit einhält, ist nicht trivial. Deshalb werden die Aufnahmeparameter in dieser Arbeit durch ein automatisches Planungsverfahren bestimmt. Es wird zwischen einem statischen Messaufbau und einem dynamischen Aufbau, bei dem der Sensor bewegt werden kann, unterschieden. In dem zweiten Fall muss das Sensorsystem zur Positionierung und Orientierung an einem Handhabungssystem, zum Beispiel an einen Industrieroboter angebracht werden. Im Rahmen dieser Arbeit wird der Schwerpunkt auf den dynamischen Aufbau gelegt. Die eingesetzten Methoden lassen sich aber auch für die Planung eines statischen Aufbaus verwenden.

## **1.3 Stand der Forschung**

In diesem Abschnitt wird der Stand der Forschung in den relevanten Feldern zusammengefasst.

### **1.3.1 Planung in der Deflektometrie**

Es gibt nur vereinzelte Veröffentlichungen über die Planung in der Deflektometrie. Erste Ansätze finden sich in Kammel [Kam04]. Er stellt ein teilautomatisches Verfahren vor, mit dem sich die Lage und die Orientierung der Kamera bestimmen lassen. Er nimmt zunächst an, dass die Kamera unendlich weit von der Testoberfläche entfernt ist und die Strahlen parallel auf die Oberfläche treffen. Damit lässt sich für diskrete Kameraansichten ein

Histogramm der Normalen, die in Richtung der Kamera zeigen, aufstellen. Hieraus werden unterschiedliche Ansichten auf die Oberfläche berechnet. In diesen Ansichten schaut die Kamera noch parallel zu dem im Sichtbereich gemittelten Normalenvektor auf die Oberfläche. Auf Basis des Histogramms werden in einem heuristischen Verfahren die Parameter der Kamera weiter verfeinert. Für den Schirm wird eine stetige Leuchtfläche berechnet, die eine optimale Untersuchung erlaubt.

Die bisher einzig bekannte Arbeit, die Sensorkonfigurationen automatisch durch einen Algorithmus bestimmt, ist [Lob13]. Die Autoren stellen ein Verfahren zur Erkennung von Hagelschäden an Pkws vor. Ihr Aufbau ist statisch und besteht aus 13 Rückprojektionsleinwänden, die um den und über dem Pkw angebracht sind. Zwischen den Schirmen sind konstruktionsbedingt Lücken, in denen Kameras angebracht werden können. Mit demselben statischen Aufbau sollen mehrere Pkw Typen, deren CAD-Modell bekannt ist, inspiziert werden können. Hierfür wird eine Menge von Kamerapositionen gesucht, die den Positionseinschränkungen genügen und einen möglichst großen Bereich aller Pkw-Oberflächen abdecken. Die zu optimierende Zielfunktion besteht aus einer Kombination der beobachteten Flächen an den unterschiedlichen Fahrzeugtypen und wird über ein Ray-Tracing Verfahren berechnet. Zur beschleunigten Berechnung der Zielfunktion wird die beobachtete Fläche eines Fahrzeugs auf einer Grafikkarte (engl. graphical processing unit, GPU) berechnet. Die Messunsicherheit wird in der Gütefunktion nicht berücksichtigt.

### 1.3.2 Sensoreinsatzplanung

Im Forschungsfeld der *Sensoreinsatzplanung* wird eine Vielzahl von Methoden untersucht [Che08], mit dem Ziel, die optimale Konfiguration eines oder mehrerer Sensoren für eine bestimmte Aufgabe zu bestimmen.

Die Anwendungsfelder im Computersehen und der Robotik sind:

- Inspektion von Objekten,
- Rekonstruktion von Oberflächen,
- Erkennung von Objekten,
- Exploration der Umgebung,
- Sensornetzwerke.

Die Literatur in diesem Bereich ist sehr umfangreich und die folgenden Betrachtungen beschränken sich auf eine Untermenge der Verfahren, die mit der in dieser Arbeit betrachteten Problemstellung verwandt sind. Dabei wird der Schwerpunkt auf Entfernungssensoren (engl. range sensor) für die Oberflächenrekonstruktion und -inspektion gelegt. Im Vergleich zur Deflektometrie können Entfernungssensoren direkt 3D-Punkte aus der Messung extrahieren. Unter Entfernungssensoren fallen hier zum Beispiel Stereo-Kameras, Sensoren die auf Projektionsverfahren beruhen, eine einzelne Kamera, die die Tiefe aus Bildsequenzen erzeugt (Structure from Motion) oder auch Time-of-Flight-Sensoren.

Die Arbeit von Tarabanis et al. [Tar95] gibt einen Überblick über die Planung für Objektmerkmalsdetektion mittels Intensitätsbildern. Bei dieser Form von Planung ist das Ziel die Sensorkonfiguration so einzustellen, dass bei bekannter Identität und Lage des Objekts bestimmte Objektmerkmale spezielle Anforderungen im Messbild erfüllen. Diese Anforderungen können zum Beispiel die Sichtbarkeit, die Vergrößerung oder die Fokussierung des zu untersuchenden Objektmerkmals sein. Tarabanis et al. klassifizieren die Arbeiten nach der Art wie Sensorkonfigurationen erzeugt werden. Sogenannte *Generate and Test* Verfahren erzeugen zunächst eine Sensorkonfiguration und testen sie auf die Einhaltung der Anforderungen. Bei den *Synthesize* Methoden sind die Anforderungen analytisch gegeben und die Sensorkonfiguration wird aus der analytischen Beziehung gewonnen.

Schließlich wird bei den Methoden basierend auf *Expertensystemen* von einem Experten das Wissen über die Planungsdomäne in Form von Regeln definiert und das System schlägt auf Basis der vorliegenden Information geeignete Sensorkonfigurationen vor.

Eine weitere neuere Übersicht findet sich in Scott et al. [Sco03]. Sie gibt einen Überblick über Planungsalgorithmen zur 3D-Objektrekonstruktion und Inspektion für aktive triangulationsbasierte Entfernungssensoren. Sie unterscheidet zwischen modellbasierten, d.h. eine Referenzoberfläche ist gegeben, und nicht modellbasierten Methoden. Unter den modellbasierten Methoden sind Methoden, die die Oberfläche und die Sensorkonfigurationen diskretisieren und anhand einer Sichtbarkeitsmatrix (engl. visibility matrix) beschreiben, mit welcher Sensorkonfiguration welches Oberflächenstück gesehen werden kann. In einer Erweiterung beschreiben die Elemente der Matrix keinen Binärwert, sondern ein Maß für die Messqualität. Eine weitere Methode nutzt sogenannte *Aspekt-Graphen*. Der Sensorkonfigurationsraum wird in Bereiche partitioniert, die qualitativ zum selben Messergebnis führen. Diese Bereiche werden Aspekte genannt und repräsentieren Knoten im Aspekt-Graphen. Benachbarte Aspekte sind mit Kanten verbunden. Diese Beschreibung eignet sich für Objekte, die sich als Polyeder beschreiben lassen. Für Freiformoberflächen ist die Definition eines Aspekt-Graphen nicht trivial und die Konstruktion nicht praktikabel.

Schließlich gibt es Verfahren, die das Problem als eine Variation des *Problem der Museumswächter* (engl. art gallery problem) formulieren. Hierbei handelt es sich um ein klassisches Problem aus der algorithmischen Geometrie. Danner und Kavradi [Dan00] stellen einen Algorithmus für die Erweiterung in drei Dimensionen vor. Die Methode beschränkt sich auf Oberflächen, die sich durch Polyeder darstellen lassen.

Fast alle betrachteten Arbeiten in Tarabanis et al. und Scott et al. nutzen geometrische Methoden zur Planung der Sensorkonfigurationen. Eine Ausnahme bilden die Arbeiten von White und Ferrie [Wha90]. Sie nutzen ein probabilistisches Modell zum Anpassen von parametrischen Superquadriken an segmentierte Tiefendaten. Außerdem nutzen sie die Unsicherheit der Schätzung, um die nächste Ansicht zu wählen. Im Folgenden werden weitere neuere Arbeiten, die einen probabilistischen Ansatz wählen, vorgestellt.

Wenhardt [Wen07] stellt ein Planungsverfahren für die Rekonstruktion einer Oberfläche anhand einer beweglichen Kamera vor. Die Oberfläche wird als eine Menge von 3D-Punkten modelliert. Mittels eines Extended Kalman Filters wird die Schätzung der 3D-Punkte mit den beobachteten 2D-Punkten aktualisiert. Der Startwert wird über ein Triangulationsverfahren aus den ersten beiden Bildern erzeugt. Die nächste Ansicht wird so gewählt, dass die Unsicherheit der 3D-Punkte am stärksten reduziert wird. Als Unsicherheitsmaße werden die sogenannten D-, E- und T-Kriterien verwendet, die aus dem Experimental Design [EPB82] bekannt sind.

Einen ähnlichen Ansatz verwendet Dunn et al. [Dun09a], [Dun09b]. Ihr Oberflächenmodell besteht aus einer Menge von ebenen Patches. Im Unterschied zu Wenhardt verwenden sie eine Kostenfunktion basierend auf den Eigenwerten der Kovarianzmatrix der geschätzten 3D-Position der Patches und weiteren Größen, die darauf ausgelegt sind, die Bildqualität zu verbessern. Beispielsweise wird ein Maß für die Texturvariation genutzt, da diese für die Korrespondenzsuche relevant ist. Hollinger [Hol12] stellt ein Verfahren für die Planung mehrerer Ansichten für autonome Unterwasserfahrzeuge (engl. autonomous underwater vehicle, AUV) zur Unterwasserinspektionen aus sonarerzeugten Punktwolken vor. Zur Repräsentation der Oberfläche und Fusion der Messdaten verwendet er einen impliziten Gauß-Prozess. In einem iterativen Algorithmus wird die nächste Ansicht so gewählt, dass diese die durchschnittliche Varianz der Schätzung minimiert.

## 1.4 Probabilistische Oberflächenrepräsentation

Aus den unzähligen Oberflächenrepräsentationen sind für diese Arbeit nur die probabilistischen Repräsentationen interessant. Hier soll auf eine kleine Auswahl eingegangen werden. Speziell für die Anwendung in der Deflektometrie wurden Markow-Zufallsfelder [Rue05] zur Rekonstruktion der Oberfläche genutzt. Horbach [Hor07] nutzt ein Markow-Zufallsfeld für die Rekonstruktion spiegelnder Oberflächen mittels der Deflektometrie. Das Markow-Zufallsfeld wird über die Pixel auf dem Sensorchip der Kamera definiert. An jedem Pixel wird der Abstand der Oberfläche von der Testoberfläche zum Projektionszentrum der Kamera in Richtung des zugehörigen Sichtstrahls geschätzt. Diese Repräsentation ist auf die Schätzung einer einzelnen Messung beschränkt und dient nur als Zwischenrepräsentation falls sich die Oberfläche aus mehreren Teilmessungen zusammensetzt.

Pak erweitert das probabilistische Voxel-Carving [Liu11] auf die Anwendung in der Deflektometrie [Pak13]. Im Vergleich zu Horbach nutzt Pak ein Markow-Zufallsfeld, das über ein Volumen des dreidimensionalen Raums definiert ist und die gesamte zu schätzende Oberfläche in impliziter Form repräsentiert. Das Volumen ist in achsenausgerichtete Quader, sogenannte *Voxel* partitioniert, die jeweils die Information darüber enthalten, ob die Oberfläche das Voxel schneidet, und wenn ja, zusätzlich die Normale an dieser Stelle. Sowohl in der Arbeit von Horbach als auch in Pak wird die Maximum A-posteriori Lösung ohne Extraktion der Unsicherheit berechnet.

Weitere Repräsentationen der Oberfläche außerhalb der Deflektometrie nutzen Gauß-Prozesse. In [Wil07] wird die Schätzung von Oberflächen mittels impliziten Gauß-Prozessen vorgestellt. Bei impliziten Gauß-Prozessen wird die Oberfläche in impliziter Form als ein Gauß-Prozess repräsentiert. In [Dra11] beispielsweise wird diese Repräsentation verwendet, um die

Form von Objekten für das automatische Greifen mittels eines Roboters zu beschreiben. Eine weitere Anwendung findet sich in [Smi10] bei der Rekonstruktion von Oberflächen mittels 3D-Punktwolken. In [Shr09] wird für die Modellierung von Terrain ein Gauß-Prozess verwendet. Im Vergleich zu den vorherigen Arbeiten stellen die Autoren hier allerdings die Oberfläche durch eine explizite Funktion dar.

## 1.5 Beiträge zum Stand der Forschung

Der Beitrag dieser Arbeit zum Stand der Forschung ist die erste Methode zur Planung der Oberflächeninspektion in der Deflektometrie mit Berücksichtigung von Unsicherheiten. Dies beinhaltet folgende Teilbeiträge:

- Das Problem der deflektometrischen Oberflächeninspektion wird als ein sequentielles probabilistisches Planungsproblem formuliert. Die Lösung des Problems erfordert, dass während der Planung eine Wahrscheinlichkeitsverteilung über der Oberfläche genutzt wird, die die gesamte Information über die bisherigen Messungen beinhaltet. Aus ihr kann eine Unsicherheit extrahiert werden, die die Güte einer Messfolge quantifiziert. Die Planungsaufgabe kann von der Informationsbeschaffung entkoppelt werden und die Spezifikationen bezüglich der Unsicherheit gestellt werden.
- Es werden zwei probabilistische Oberflächenmodelle für die Deflektometrie vorgestellt und deren Eignung für die Planung untersucht. Das erste Modell basiert auf einem Gauß-Prozess. Dieser ermöglicht die Rekonstruktion der Oberfläche aus Neigungsdaten und zusätzlichen Punkten. Zusätzlich liefert das Modell Unsicherheiten, die für die Kostenfunktion der Planung relevant sind. Das deflektometrische Rekonstruktionsproblem wird linearisiert, indem angenommen wird, dass eine Referenzoberfläche existiert. Das zweite Modell repräsentiert

die Oberfläche als eine Menge von diskreten Spiegeln. Die Neigung der Spiegel wird anhand eines Kalman Filters geschätzt.

- Es werden Planungsverfahren zur Minimierung der Unsicherheit untersucht. Untersucht wurden der Greedy-Planungs-Algorithmus und eine weitere Methode, die für eine feste Anzahl an Messungen einen Plan erzeugt und über die Anzahl der Messungen sucht. Die vorgestellten Planungsalgorithmen reduzieren die Minimierung auf die Optimierung einzelner Sensorkonfigurationen. Hierzu wird ein Simulated-Annealing Optimierer für die Minimierung des deflektometrischen Gütemaßes angepasst.

## 1.6 Gliederung der Arbeit

Eine wichtige Rolle für die Gliederung der Arbeit spielt Kapitel 2. Hier wird das probabilistische Rahmenwerk vorgestellt, in dem das betrachtete Problem formuliert wird. Die drei Hauptkomponenten in dem Rahmenwerk sind ein Messmodell, ein probabilistisches Oberflächenmodell und ein Planungsalgorithmus. Diese werden in den nächsten drei Kapiteln beschrieben. In Kapitel 3 wird das verwendete Messmodell für den deflektometrischen Sensor vorgestellt und seine Implementierung anhand eines Simulators. Das Modell beinhaltet ebenfalls eine Beschreibung der Messunsicherheit. In Kapitel 4 werden zwei probabilistische Oberflächenmodelle vorgestellt. Hier wird ebenfalls auf die Fusion der Messdaten in das Modell und die Extraktion der Gütemaße basierend auf der Unsicherheit eingegangen. Die Planungsverfahren werden in Kapitel 5 vorgestellt. Dabei werden zunächst die Optimierungsalgorithmen behandelt. In Kapitel 6 werden die vorgestellten Planungsalgorithmen an synthetischen und realen Oberflächen untersucht. Schließlich wird in 7 eine Zusammenfassung und ein Ausblick gegeben.



## **2 Rahmenwerk für die probabilistische Planung in der Oberflächeninspektion**

In diesem Kapitel wird ein Rahmenwerk zur probabilistischen Planung in der Oberflächeninspektion vorgestellt. Diese basiert auf Huber, der in [Hub09] ein allgemeines Rahmenwerk für die Sensoreinsatzplanung beschreibt. Die Oberflächeninspektion wird als ein sequentielles Entscheidungsproblem modelliert. Ein sequentielles Entscheidungsproblem besteht aus einem Agenten und einer Umwelt. Der Agent besitzt ein Ziel und kann eine Menge möglicher Aktionen ausführen, mit denen er die Umwelt beeinflussen kann. Nachdem der Agent eine Aktion ausgeführt hat, erhält er eine Beobachtung von der Umwelt und entscheidet auf Grundlage dieser Information und der in der Vergangenheit gesammelten Information, welche Aktion als nächstes ausgeführt werden soll.

In dieser Arbeit liegt das Interesse speziell bei Agenten, die das Ziel verfolgen, mehr Information über ihre Umwelt zu sammeln. Im Anwendungsfeld der Oberflächeninspektion kann angenommen werden, dass die Umwelt stationär ist und aus dem Prüfobjekt und dem konfigurierbaren Messsystem besteht. Der betrachtete Agent steuert das Messsystem und die Aktionen entsprechen Messungen am Prüfobjekt mit unterschiedlichen Sensoreinstellungen.

In den Abschnitten 2.1 und 2.2 wird ein häufig genutztes mathematisches Modell für probabilistische sequentielle Entscheidungsprobleme vorgestellt.

In Abschnitt 2.3 wird die Oberflächeninspektion in dem vorgestellten Modell formalisiert. In diesem Modell wird die Gütefunktion über die Unsicherheit modelliert. Hier sind geeignete Unsicherheitsmaße notwendig, die in Abschnitt 2.4 vorgestellt werden. Schließlich werden in Abschnitt 2.5 einige generelle Lösungsverfahren diskutiert und anschließend das für diese Arbeit geeignetste Verfahren identifiziert.

## 2.1 Markow-Entscheidungsprozess (MDP)

Ein Markow-Entscheidungsprozess (engl. Markov Decision Process, MDP) [ey\[Ber17\]](#) modelliert probabilistische sequentielle Entscheidungsprobleme. Er findet Anwendung unter anderem in Gebieten wie dem Operations Research [[Whi85](#), [Whi88](#), [Whi93](#)] oder der Robotik. Zu diskreten Zeitpunkten  $k \in \mathbb{N}^+$  wird die für die Anwendung relevante Umwelt mit einem Zustand  $x_k \in \mathcal{X}$  beschrieben. Mit Wissen über den Zustand trifft der Agent zu jedem Zeitpunkt eine Entscheidung, wählt also eine Aktion  $a_k \in \mathcal{A}$  aus<sup>1</sup>. Dies überführt die Umwelt in einen Nachfolgezustand, der durch die Transitionswahrscheinlichkeit  $p(x_{k+1} | x_k, a_k)$  modelliert wird. Dabei hängt der Folgezustand nur von der gewählten Aktion und dem aktuellen Zustand ab, nicht aber von den vergangenen Zuständen. Hierbei handelt es sich um die Markow-Annahme, die formal besagt, dass der Folgezustand  $x_{k+1}$  bedingt unabhängig von den vergangenen Zuständen  $x_{k-1} \dots, x_0$  ist:

$$p(x_{k+1} | x_k, x_{k-1} \dots, x_0, a_k) = p(x_k | x_{k-1}, a_k). \quad (2.1)$$

Für jedes Paar von Zustand und gewählter Aktion gibt eine Schrittkostenfunktion  $g(x_k, a_k)$  die Güte der Entscheidung an. Das Ziel ist eine Minimierung

---

<sup>1</sup> Im Allgemeinen kann die Aktionsmenge  $\mathcal{A}$  sowohl vom Zeitpunkt  $k$  als auch von dem Zustand  $x_k$  abhängen.

der erwarteten Kosten. Zusammenfassend besteht das MDP aus einem 5-Tupel  $(\mathcal{X}, \mathcal{A}, p(x_{k+1} | x_k, a_k), g(x_k, a_k), p_0(x_0))$ , in dem

- $\mathcal{X}$  die Menge der Zustände,
- $\mathcal{A}$  die Menge der Aktionen,
- $p(x_{k+1} | x_k, a_k)$  die Transitionswahrscheinlichkeit,
- $g : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$  die Schrittostenfunktion und
- $p_0(x_0)$  die Anfangsverteilung der Zustände

darstellt. Bei der Oberflächeninspektion ist der Zustand nicht vollständig beobachtbar, da Informationen nur indirekt anhand Messungen bezüglich des Prüfobjektes gesammelt werden können. Für diese Art von Entscheidungsproblemen gibt es eine Erweiterung des MDPs zu einem sogenannten *teilweise beobachtbaren Markow-Entscheidungsprozessen*, die im Folgenden vorgestellt werden.

## 2.2 Teilweise beobachtbarer Markow-Entscheidungsprozess (POMDP)

Der teilweise beobachtbare Markow-Entscheidungsprozess (engl. Partial Observable Markov Decision Process, POMDP) [Kae98] ist eine Verallgemeinerung des MDP. In dieser Modellierung steht dem Agenten für eine Entscheidung der Zustand nicht zur Verfügung. Stattdessen wird eine weitere Größe  $z_k \in \Omega$  beobachtet, die in einer Relation zum Zustand steht. Zum Beispiel ist  $z_k$  nur ein Teil des Zustands oder nur ein verrauschter Zustandswert. Der Zusammenhang der Beobachtung mit dem Zustand wird über eine bedingte Dichte  $p(z_{k+1} | x_k, a_k)$  modelliert. Eine alternative und gleichwertige Beschreibung des Zusammenhangs ist über eine Funktion  $z_{k+1} = h(x_k, a_k, w_k)$  möglich, wobei  $w_k$  ein Rauschgröße ist.

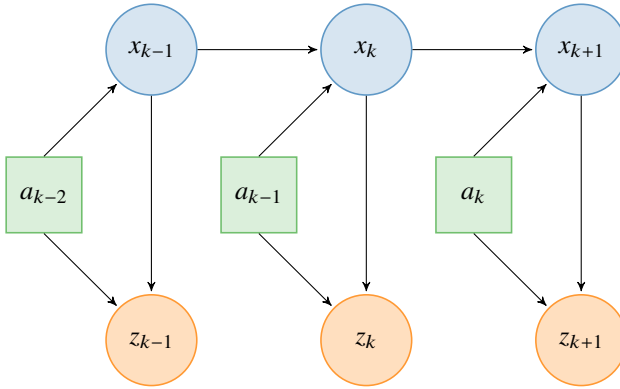


Abbildung 2.1: Schematische Darstellung der direkten Abhängigkeiten zwischen Zuständen, Beobachtungen und Aktionen in einem POMDP.

Eine schematische Darstellung der direkten Abhängigkeiten der Zufallsgrößen und Entscheidungsvariablen ist in Abbildung 2.1 gegeben. Zusammenfassend wird ein POMDP durch ein 7-Tupel

$$(\mathcal{X}, \mathcal{A}, p(x_{k+1} | x_k, a_k), \Omega, p(z_{k+1} | x_k, a_k), g, p_0(x_0)) \quad (2.2)$$

beschrieben mit

- $\mathcal{X}$  die Menge der Zustände,
- $\mathcal{A}$  die Menge der Aktionen,
- $p(x_{k+1} | x_k, a_k)$  die Transitionswahrscheinlichkeit,
- $\Omega$  die Menge der Beobachtungen,
- $p(z_{k+1} | x_k, a_k)$ : die Beobachtungswahrscheinlichkeit,
- $g : (\mathcal{A} \times \Omega)^k \times \mathcal{A} \rightarrow \mathbb{R}$  die Schrittkostenfunktion und
- $p_0(x_0)$  die Anfangsverteilung der Zustände.

In der Literatur werden die Schrittkosten des POMDP über die Schrittkosten des zugrundeliegenden MDP formuliert. Die Schrittkosten des zugrundeliegenden MDP hängen nur von dem Zustand  $x_k$  und der Aktion  $a_k$  ab. Die hier definierten Schrittkosten sind in einer allgemeineren Form definiert und hängen von allen bisherigen Beobachtungen  $z_1, \dots, z_k$  und allen bisherigen Aktionen  $a_0, \dots, a_{k-1}$  ab. Die Anordnung dieser Größen in einer chronologischen Folge

$$\mathcal{H}_k = (a_0, z_1, \dots, a_{k-1}, z_k) \in (\mathcal{A} \times \Omega)^k \quad (2.3)$$

wird auch *Historie* oder *Informationsvektor* zum Zeitpunkt  $k$  genannt. Sie fasst die gesamte Information, die bis zum Zeitpunkt  $k$  über die vergangenen und den aktuellen Zustand vorhanden ist, zusammen. Die A-posteriori-Verteilung über dem Zustandsraum ist eine zur Historie gleichwertige Größe [Sma73] und wird mit

$$b_k(x_k) = p(x_k \mid \mathcal{H}_k) \quad (2.4)$$

bezeichnet. Während die Historie nur aus einer Aneinanderreihung aller Aktionen und Beobachtungen besteht, besitzt die A-posteriori-Verteilung den Vorteil, dass sie schon die Inferenz bezüglich der interessierenden Größe, nämlich des Zustands beinhaltet. In dieser Arbeit sind folgende zwei Spezialfälle der Kostenfunktion von Interesse:

- Falls sich die Kostenfunktion als

$$g(\mathcal{H}_k, a_k) = \int \tilde{g}(x_k, a_k) b_k(x_k) dx_k \quad (2.5)$$

schreiben lässt, wobei  $\tilde{g}(x_k, a_k)$  die Kostenfunktion des zugrundeliegenden Markow-Entscheidungsprozesses ist, erhält man den in der Literatur definierten POMDP.

- Für den Fall, dass sich die Kostenfunktion als eine allgemeine Funktion  $\rho : \mathcal{B} \times \mathcal{A} \rightarrow \mathbb{R}$  der A-posteriori-Verteilung und Aktion

$$g(\mathcal{H}_k, a_k) = \rho(b_k, a_k) \quad (2.6)$$

schreiben lässt, erhält man ein  $\rho$ POMDP [Ara10]. Dabei ist  $\mathcal{B}$  die Menge der A-posteriori-Verteilungen. Dieser Fall beinhaltet auch den ersten Fall, aber zusätzlich auch Fälle, bei denen die Kostenfunktionen die Unsicherheit der A-posteriori-Verteilung  $b_k$  nutzt.

Im Folgenden wird begrifflich nicht zwischen  $\rho$ POMDP und POMDP unterschieden und beide werden mit POMDP bezeichnet.

### 2.2.1 Belief-MDP

Da in einem POMDP die A-posteriori-Verteilung über dem Zustand eine zur Historie gleichwertigen Größe ist, enthält sie die gesamte Information, auf deren Basis Entscheidungen getroffen werden können. Demnach ist es möglich, den POMDP in ein gleichwertiges MDP umzuformulieren [Son78]. Die A-posteriori-Verteilung übernimmt in dem abgeleiteten MDP die Rolle des Zustands. In diesem Zusammenhang wird die A-posteriori-Verteilung auch *Beliefstate* genannt und der MDP als *Belief-MDP* bezeichnet. Formal ist ein Belief-MDP ein 6-Tupel  $(\mathcal{B}, \mathcal{A}, \Omega, \rho, b_0)$  wobei

- $\mathcal{B}$  die Menge der Beliefstates  $b$  über dem MDP,
- $\mathcal{A}$  die Menge der Aktionen  $a$ ,
- $\Omega$  die Menge der Beobachtungen  $z$ ,
- $B : \mathcal{B} \times \mathcal{A} \times \Omega \rightarrow \mathcal{B}$  die Zustandstransitionsfunktion,
- $\rho : \mathcal{B} \times \mathcal{A} \rightarrow \mathbb{R}$  die Kostenfunktion und
- $b_0(x_0)$  die A-priori-Verteilung des Zustands

ist. Die Zustandsmenge  $\mathcal{B}$  besteht aus der Menge aller Wahrscheinlichkeitsverteilungen über dem ursprünglichen Zustandsraum  $\mathcal{X}$ . Die Menge der Aktionen wird vom POMDP übernommen. Der Zustand  $b_{k+1}$  zum nächsten Zeitpunkt wird anhand

$$\underline{b}_{k+1} = B(\underline{b}_k, a_k, \underline{z}_{k+1}) \quad (2.7)$$

berechnet mit

$$\underline{b}_{k+1}(x_{k+1}) = \frac{p(\underline{z}_{k+1} | x_k, a_k) b_{k+1}^-(x_{k+1})}{p(\underline{z}_{k+1})}, \quad (2.8)$$

wobei  $b_{k+1}^-(x_{k+1})$  im Allgemeinen die Form

$$b_{k+1}^-(x_{k+1}) = \int p(x_{k+1} | x_k, a_k) b_k(x_k) dx_k \quad (2.9)$$

hat. Für einen stationären Zustand gilt für die Transitionswahrscheinlichkeit

$$p(x_{k+1} | x_k, a_k) = \delta(x_{k+1} - x_k), \quad (2.10)$$

wobei  $\delta(x)$  die Delta-Distribution bezeichnet. In diesem Fall verändert sich der Zustand nicht und die Aktionen beeinflussen nur die Beobachtungen. Für stationäre Zustände ist  $b_{k+1}^-(x_{k+1}) = b_k(x_k)$ . Die Kostenfunktionen der Form (2.5) oder (2.6) können direkt übernommen werden, weil sie bezüglich des Beliefstates definiert sind. Der Anfangszustand  $b_0(x_0)$  ist gleich der A-priori-Verteilung  $p_0(x_0)$  des Zustands des POMDPs.

## 2.2.2 Strategie

Während in einem MDP die optimale Entscheidung zu jedem Zeitschritt nur vom Zustand desselben Zeitpunkts abhängt, muss für eine optimale Entscheidung im POMDP diese von der Beliefstate abhängen. Eine Funktion,

die abhängig von der Beliefstate und des Zeitschritts die Aktion für denselben Zeitschritt ausgibt, wird *Strategie* (engl. policy) genannt  $\pi : \mathcal{B} \times \mathbb{N}^+ \rightarrow \mathcal{A}$ . Wenn die Strategie unabhängig von dem Zeitschritt ist, spricht man von einer *stationären Strategie* und wenn sie unabhängig von der Historie ist, von einem *Plan*. Die Güte einer Strategie wird durch den Ausdruck

$$J(b_0, \pi) = \mathbb{E} \left\{ \sum_i \alpha^i g(\underline{b}_i, \pi(\underline{b}_i, i)) \right\} \quad (2.11)$$

definiert. Dabei ist  $\alpha \in (0, 1]$  eine skalare Gewichtung, durch deren geeignete Wahl sichergestellt wird, dass die Reihe konvergiert. Für den Fall  $\alpha = 1$  erhält man die erwarteten Gesamtkosten. Für den Fall, dass  $\alpha < 1$  gewählt werden muss, spricht man von *diskontierten Kosten*.

## 2.3 Probabilistische Modellierung der Oberflächeninspektion

In diesem Abschnitt wird die Oberflächeninspektion als ein POMDP modelliert. Eine schematische Darstellung der Abhängigkeiten ist in Abbildung 2.2 gegeben. Der Zustand in einer Oberflächeninspektionsaufgabe besteht aus der zu prüfenden Oberfläche und dem Zustand des Sensors

$$\mathcal{X} = \mathcal{M}_S \times \Theta, \quad (2.12)$$

wobei  $\mathcal{M}_S$  eine Menge von Oberflächen und  $\Theta$  die Menge der Sensorkonfigurationen bezeichnet. Eine Sensorkonfiguration umfasst die relevanten einstellbaren Parameter am Messinstrument, mit denen sich eine Messung beeinflussen lässt.



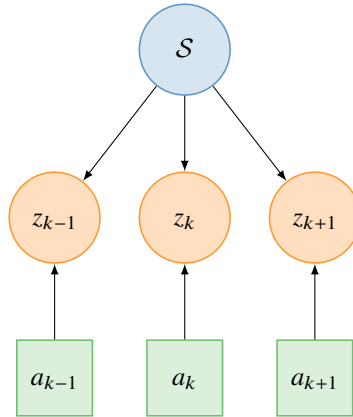


Abbildung 2.2: Direkte Abhängigkeiten von Aktionen  $a_k$ , Beobachtung  $z_{k+1}$  und der Oberfläche  $\mathcal{S}$  bei der Oberflächeninspektion.

Die zu inspizierende Oberfläche  $\mathcal{S} \in \mathcal{M}_{\mathcal{S}}$  bleibt während des Messvorgangs unverändert. Das heißt die Oberfläche selbst ist starr und befindet sich in einer stabilen Position. Außerdem wird das Prüfobjekt nicht von den Messungen beeinflusst. Eine Bewegung der Oberfläche zwischen den Messungen kann durch eine entgegengesetzte Bewegung des Sensors modelliert werden. Die Oberfläche ist teilweise beobachtbar. Über sie kann Information nur anhand von Messungen gewonnen werden. Eine Messung gibt nur eine beschränkte Sicht auf die Oberfläche. Beispielsweise erfasst sie nur spezielle Merkmale der Oberfläche in einem örtlich begrenzten Bereich und ist mit Rauschen gestört. Der Sensorkonfigurationszustand  $\tilde{\theta} \in \Theta$  wird dagegen als vollständig beobachtbar modelliert.

Die Aktionen bestehen aus einer Vereinigung zweier Aktionstypen:

$$\mathcal{A} = \mathcal{A}_{\text{I}} \cup \mathcal{A}_{\text{T}}. \quad (2.13)$$

Die Menge  $\mathcal{A}_I$  bestehen aus den Aktionen, die das Messsystem einstellen und eine nachfolgende Messung mit dieser Konfiguration ausführen. Die Menge  $\mathcal{A}_I$  ist isomorph zur Menge der Sensorkonfigurationen  $\mathcal{A}_I \simeq \Theta$ . Sei  $a \in \mathcal{A}_I$ , dann wird mit  $\theta_a$  die zugehörige Sensorkonfiguration bezeichnet. Die Menge  $\mathcal{A}_T$  besteht aus Aktionen, die die Inspektion beenden und eine Prüfentscheidung ausgeben. Sie werden *Terminalaktionen* genannt. Beispiele für Terminalaktionen bei einer Prüfaufgabe, bei der defekte Objekte aussortiert werden sollen, könnten das Aussortieren des Prüfobjekts aufgrund signifikanter Defekte, das Akzeptieren des Prüfobjekts oder die Klassifizierung des Prüfobjekts als B-Ware sein. Abhängig von der Prüfaufgabe können die Terminalaktionen auch komplexere Formen annehmen, wie zum Beispiel bei einer Prüfaufgabe, in der die Anzahl und Position der Fehler bestimmt werden sollen.

Die Zustandstransitionsfunktion für eine Aktion  $a_k \in \mathcal{A}_I$  ist durch

$$p((S_{k+1}, \tilde{\theta}_{k+1}) | (S_k, \tilde{\theta}_k), a_k) = \delta(S_{k+1} - S_k) \delta(\tilde{\theta}_{k+1} - \theta_{a_k}) \quad (2.14)$$

gegeben. Durch das Ausführen der Aktion  $a_k \in \mathcal{A}_I$  zu einem Zeitpunkt  $k$  wird der Sensor in die Sensorkonfiguration  $\theta_{a_k}$  versetzt und die Prüfoberfläche bleibt unverändert. Das Ausführen einer Terminalaktion  $a_k \in \mathcal{A}_T$  lässt sowohl die Prüfoberfläche als auch die Sensorkonfiguration unverändert.

Die Prüfung einer Oberfläche besteht aus mehreren Messungen, die zum Beispiel an lokalen örtlichen Bereichen der Oberfläche erfolgen. Die Messungen erfolgen in Zeitintervallen  $[t_k, t_{k+1})$ . Am Anfangszeitpunkt  $t_k$  jedes Intervalls wird der Sensor mit der Sensorkonfiguration  $\theta_k$  eingestellt, also eine Aktion aus der Menge  $\mathcal{A}_I$  gewählt. Danach wird die Messung ausgeführt. Ihr Ergebnis ist die Beobachtung  $z_{k+1}$  und steht am Ende der Messung zum Zeitpunkt  $t_{k+1}$  zur Verfügung. Die Beobachtung  $z_{k+1}$  kann je nach Sensor unterschiedliche Formen annehmen. Sie kann zum Beispiel bei einem triangulationsbasierten Sensor aus einer Menge von Tupeln, die aus einer

Messstelle auf der Oberfläche und der zugehörigen Höhe zusammengesetzt sind, bestehen. Bei einem deflektometrischen Sensor, wie er hier in dieser Arbeit betrachtet wird, werden stattdessen Messstellen und deren Neigungen gemessen. Der Messvorgang wird mit einer Funktion  $h$  modelliert:

$$z_{k+1} = h((S_k, \tilde{\theta}_k), a_k, \underline{w}_k) \quad (2.15)$$

wobei  $\underline{w}_k$  ein Rauschen ist.

Die Schrittkosten werden gemäß der Gleichung (2.5) gewählt. Hierbei reicht es die Kostenfunktion des zugrundeliegenden MDPs zu definieren:

$$\tilde{g}((S, \tilde{\theta}), a) = \begin{cases} c(S, a) & a \in \mathcal{A}_T \\ g_a(\tilde{\theta}, a) & a \in \mathcal{A}_I \end{cases} \quad (2.16)$$

Die Funktion  $c : \mathcal{M}_S \times \mathcal{A}_T \rightarrow \mathbb{R}$  beschreibt die Kosten für die Prüfentscheidung, zum Beispiel wird hier bestraft, wenn eine Oberfläche fälschlicherweise aussortiert wird oder Defekte übersehen werden. Die Funktion  $c$  realisiert die Spezifikation der Inspektionsaufgabe. Die Funktion  $g_a : \Theta \times \mathcal{A}_I \rightarrow \mathbb{R}$  ist eine Kostenfunktion, die die Kosten für das Einstellen des Sensors und den Messaufwand, wie Messzeit beinhaltet. Bei Kostenfunktion (2.16) treten die ausschlaggebenden Kosten am Ende der Planung durch die Terminalaktion auf.

### 2.3.1 Trennung der Informationsbeschaffung und der Prüfaufgabe

Wie die obige Modellierung andeutet, lässt sich die Oberflächeninspektion in zwei Phasen einteilen. Die erste Phase besteht aus der Informationsbeschaffung anhand von Messungen. In der zweiten Phase wird die Information über die Oberfläche ausgewertet, um eine Prüfentscheidung zu treffen. Die

obige Formulierung beinhaltet beide Phasen. Die Kostenfunktion bewertet eine Messfolge abhängig davon, ob die zugehörige Inspektionsaufgabe, wie das Aussortieren von defekten Oberflächen gut erledigt werden kann. Informationsbeschaffung ist hier nur Mittel zum Zweck. Die Formulierung hat den Vorteil, dass die hierauf basierenden Strategien optimal für die Inspektionsaufgabe sind.

Es gibt allerdings mehrere Argumente die Informationsbeschaffung und die Inspektionsaufgabe bei der Berechnung der Strategie zu separieren. Durch eine Separierung kann eine höhere Flexibilität bei der Planung erreicht werden, da eine gewisse Unabhängigkeit bezüglich der Inspektionsaufgabe erreicht werden kann, d.h. die in Gleichung (2.16) definierte Spezifikation der Aufgabe über die Kostenfunktion  $c$  kann in gewissen Grenzen nachträglich verändert werden. Dies ist vorteilhaft, zumal in der Praxis häufig die genaue Spezifikation von Defekten schwierig ist. Das ist beispielsweise bei Defekten der Fall, die nach ästhetischen Kriterien [Zie14, Zie15] bewertet werden und damit nicht leicht formalisierbar sind. Ein weiterer pragmatischer Grund für die Separierung besteht darin, dass durch die verzögerten Kosten der Terminalaktionen am Ende der Inspektion eine Berechnung der Strategie erschwert werden kann.

Eine zuverlässige Prüfentscheidung ist nicht möglich, wenn nicht genug Information bezüglich der für die Inspektion relevanten Merkmale der Oberfläche vorhanden ist. Erst durch genügend Messungen lässt sich umfassende Information über die Oberfläche sammeln und eine Prüfentscheidung treffen. Das Maß an Information über die Oberfläche lässt sich anhand der Unsicherheit bezüglich der Schätzung der Merkmale der Oberfläche beschreiben. Sie erlaubt die Trennung der beiden Phasen. Sei  $U : \mathcal{B} \rightarrow \mathbb{R}$  ein Unsicherheitsmaß, das auf der Menge der Beliefstates definiert ist. Eine Informationsbeschaffungsstrategie  $\pi_1 : \mathcal{B} \times \Theta \rightarrow \mathcal{A}_I$  führt solange Messungen aus, bis die Unsicherheit des Beliefstates eine untere Schranke  $U_{\text{suff}}$  erreicht.

In dieser Phase wird damit die Spezifikation durch die Anforderung, die hinreichende Unsicherheit zu erreichen, ersetzt. Anschließend entscheidet eine Prüfentscheidungsstrategie  $\pi_T : \mathcal{B} \rightarrow \mathcal{A}_T$  über das Inspektionsergebnis. Das ist äquivalent mit der Einschränkung der gesuchten Strategien auf

$$\pi((b_k, \tilde{\theta}), k) = \begin{cases} \pi_I((b_k, \tilde{\theta}), k) & U(b_k) > U_{\text{suff}} \\ \pi_T(b_k, k) & \text{sonst.} \end{cases} \quad (2.17)$$

Die Informationsbeschaffungsstrategie berechnet sich aus einem POMDP, wie er im vorherigen Unterabschnitt definiert wurde, mit dem Unterschied, dass anstelle der Schrittkosten (2.16) die Schrittkosten  $\rho : (\mathcal{B} \times \Theta) \times \mathcal{A}_I \rightarrow \mathbb{R}$ ,

$$\rho((b_k, \tilde{\theta}_k), a_k) = g_u(b_k, a_k) + \lambda g_a(\tilde{\theta}_k, a_k) \quad (2.18)$$

gewählt werden, wobei es sich bei der Funktion  $g_u$  um den negativen *erwarteten Informationsgewinn* handelt und dieser definiert ist als

$$g_u(b_k, a_k) = -\mathbb{E} \{U(b_k) - U(\underline{b}_{k+1})\} \quad (2.19)$$

mit  $\underline{b}_{k+1} = B(b_k, a_k, \underline{z}_{k+1})$ . Der Betrag beschreibt das Ausmaß, um das die Unsicherheit durch die Aktion  $a_k$  reduziert wird. Der Informationsgewinn selbst ist eine Zufallsvariable, da der Folgezustand  $\underline{b}_{k+1}$  von der zufallsbehafteten Beobachtung  $\underline{z}_{k+1}$  abhängt und deren Realisierung erst nach dem Ausführen der Aktion  $a_k$  bekannt wird. Die skalare Größe  $\lambda \in \mathbb{R}$  in Gleichung (2.18) ist ein Faktor, der die beiden Kostenfunktionen untereinander gewichtet. Die Schrittkosten  $\rho$  in Gleichung (2.18) haben die Form (2.6), so dass sie eine Definition der Kosten auf Basis der Unsicherheit erlauben. Falls die Kostenfunktion  $c$  bekannt ist, kann die Prüfentscheidungsstrategie aus  $\arg \min_a \int c(\mathcal{S}, a) b_n(\mathcal{S}) d\mathcal{S}$  berechnet werden, wobei  $n$  die Anzahl der Messungen ist. Ansonsten kann eine auf heuristische oder auf maschinelles Lernen basierte Prüfentscheidung getroffen werden.

## 2.4 Unsicherheitsbestimmung

Die Kostenfunktion  $\rho$  hängt von dem Beliefstate ab. Ihre Minimierung führt zur Reduktion der Unsicherheit. Mögliche Formen der Unsicherheitsquantifizierung einer Verteilung lassen sich in die zwei Klassen *Kovarianz-basierte* und *informationstheoretisch basierte* einteilen. Informationstheoretische Unsicherheitsmaße basieren auf der Shannonschen Informationstheorie [Sha49]. Bei diesen Maßen wird die bedingte differentielle Entropie des Zustandes zur Berechnung der Unsicherheit herangezogen. Sie sind für den weiteren Verlauf dieser Arbeit nicht relevant und sollen deshalb nicht näher beschrieben werden. Kovarianz-basierte Kostenfunktionen hängen von der A-posteriori-Kovarianz des Beliefstates ab:

$$\mathbf{C}(b_k) = \text{Cov} \{ \underline{x}_k \mid \mathcal{H}_k \} \quad (2.20)$$

Hierbei wird die Kovarianz bezüglich des Zustands  $\underline{x}_k$  gebildet, dessen Verteilung durch den Beliefstate  $b_k$  gegeben ist. Während die Kovarianzmatrix die Varianzen in allen Richtungen enthält, ist für die Planung nur eine skalare Unsicherheitsgröße notwendig. Es gibt mehrere Methoden eine skalare Unsicherheitsgröße aus der Kovarianz zu extrahieren:

- Die *Spur* der Matrix

$$U_T(\mathbf{C}) = \text{Tr } \mathbf{C} = \sum_i C_{ii} \quad (2.21)$$

entspricht der Summe aller Varianzen  $C_{ii}$ . Geometrisch entspricht dies der Summe aller Halbachsen des Kovarianzhyperellipsoids.

- Die *Determinante*

$$U_D(\mathbf{C}) = \det \mathbf{C} \quad (2.22)$$

ist proportional zum Volumen des Kovarianzhyperellipsoids. Für eine Normalverteilung mit Kovarianz  $\mathbf{C}$  ist die Entropie eine streng monoton wachsende Funktion der Determinanten  $\det \mathbf{C}$ . Folglich ist in diesem Fall die Minimierung der Determinanten äquivalent zur Minimierung der Entropie.

- Der *größte Eigenwert*

$$U_E(\mathbf{C}) = \max \text{eig}(\mathbf{C}) \quad (2.23)$$

ist die Länge der größten Halbachse des Kovarianzhyperellipsoids.

- Das maximale Diagonalelement

$$U_M(\mathbf{C}) = \max_i C_{ii} \quad (2.24)$$

unterscheidet sich von den anderen Größen, da es nicht invariant bezüglich einer Koordinatentransformation ist.

Bei der Oberflächeninspektion können die Varianzen  $C_{ii}$  meist physikalisch interpretiert werden. Die Interpretation ist abhängig von der gewählten Oberflächenrepräsentation. Zum Beispiel können die Varianzen in dem in Abschnitt 4.3 definierten Oberflächenmodell als Varianz der geschätzten Höhe an einer Oberflächenstelle interpretiert werden.

## 2.5 Lösungsverfahren

Die exakte Lösung des Planungsproblems besteht in der Berechnung einer optimalen Strategie. Die optimale Strategie wird in einer Planungsphase vor der eigentlichen Messung berechnet und in einer Messphase ausgeführt. Das Ausführen einer Strategie basiert auf einem geschlossenen Kreislauf. Anhand

des aktuellen Beliefstates wird mit der Strategie die aktuelle Aktion berechnet und die zugehörige Messung ausgeführt. Die Messdaten werden genutzt, um die Beliefstates zu aktualisieren. Die Ausführung mit Rückführung ist bei Entscheidungsproblemen mit hohen Unsicherheiten sinnvoll, da nicht mit Sicherheit vorausgesagt werden kann, in welchen Zustand das System wechselt oder welche Beobachtung eine Messung hervorbringt. Allerdings ist die Berechnung solch einer Strategie schon für relativ kleine Probleme aufwendig. In dieser Arbeit ist der Zustand besonders hochdimensional. Der Zustand des Belief-MDPs besteht aus einer Verteilungen über allen Oberflächen. Die bekannten exakten Lösungsmethoden [Mur00, Bri00] skalieren nur auf Probleme mit kleinen Dimensionen. Deshalb sind approximative Methoden unverzichtbar.

Die sogenannte *open loop* Planung [Ber17] ist eine approximative Lösungsmethode, bei der für jeden aktuellen Zeitpunkt ein Plan berechnet und die erste Aktion hieraus ausgeführt wird. Aufgrund von Unsicherheiten in der Zustandstransition und dem Messmodell führt eine Aktion zu einem Zustand, für den der restliche Plan nicht geeignet ist. Deshalb wird im nächsten Zeitschritt ein neuer Plan auf Basis eines aktualisierten Belief-States berechnet. Da die Planung bis zum Ende des Zeithorizontes in der Praxis noch zu aufwendig ist, wird häufig über einen kürzeren Planungshorizont geplant. Dies wird *Modellprädiktive Planung* genannt. Ein Spezialfall hiervon ist die Greedy-Planung, die nur einen Schritt im Voraus plant. Diese Algorithmen planen in der Ausführungsphase, also während der tatsächlichen Messung. Die Neuplanung in jedem Schritt ist aufgrund der probabilistischen Natur des Problems erforderlich.

Für eine Oberflächeninspektion ist eine online Planung, wie sie in der *open loop* Planung oder der Modellprädiktiven Planung erfolgt, nicht zweckmäßig. Häufig erfordern Inspektionsaufgaben hohe Taktraten, die mit diesen Methoden nicht eingehalten werden können. Die Oberflächeninspektion bietet eine



kontrollierte Umgebung, in der durch den Messaufbau die Messunsicherheit niedrig gehalten werden kann. In diesem Fall kann trotz Unsicherheit eine Approximation als ein deterministisches Planungsproblem durchgeführt werden. Hierzu muss die Zustandstransitionsfunktion der Beliefstate  $B(b_k, a_k, z_{k+1})$  unabhängig von der Realisierung der Beobachtung sein. Im Fall von geringem Rauschen kann das damit erreicht werden, indem unsichere Größen durch ihre Erwartungswerte ersetzt werden.

Eine weitere Möglichkeit die Unabhängigkeit zu erreichen ist, wenn folgende Bedingungen erfüllt sind:

1. Die Verteilung des Zustandes kann in jedem Messschritt mit einer Normalverteilung beschrieben werden.
2. Die Kosten beruhen auf einem Unsicherheitsmaß, der nur von der Kovarianzmatrix abhängt.
3. Die Beobachtungen hängen linear vom Zustand ab und das Beobachtungsrauschen ist normalverteilt.

Die Unabhängigkeit wird dadurch erreicht, dass bei Erfüllung der obigen Bedingungen die A-posteriori-Kovarianzmatrix des Zustandes nicht von den Beobachtungen abhängt. Diese Approximation wird in dieser Arbeit verfolgt. Der erste und zweite Punkt werden dadurch sichergestellt, dass die vorgestellten Oberflächenrepräsentationen auf einer Normalverteilung basieren; siehe Kapitel 4. Da der dritte Punkt im Allgemeinen nicht gegeben ist, wird für nichtlineare Beobachtungsfunktionen diese um den beobachteten Wert linearisiert; siehe Kapitel 3.

## 2.6 Zusammenfassung

In diesem Kapitel wurde ein probabilistisches Rahmenwerk zur Modellierung des Oberflächeninspektionsproblems vorgestellt und Lösungsverfahren

diskutiert. Die Oberflächeninspektion wird als ein sequentielles Entscheidungsproblem mittels eines Belief-MDPs formuliert. Zunächst wurde eine Modellierung gewählt, bei der das Ziel der Planung in der Erfüllung der Inspektionsaufgabe lag. Anschließend wurde diskutiert, dass die Trennung der Informationsbeschaffung und der Inspektion mehrere Vorteile bringt. Zum einen ist häufig die direkte Modellierung einer Gütefunktion für die Inspektion schwierig, so dass eine Trennung eine gewisse Unabhängigkeit zur dieser liefert. Zum anderen ist die Berechnung einer Strategie schwierig, da die Kosten erst am Ende, bei Erfüllen der Aufgabe, zur Verfügung stehen.

Die Trennung der Informationsbeschaffung von der Inspektion ist möglich, da durch die probabilistische Formulierung die gesamte Information über die Messungen in Form der Unsicherheit vorliegt. Diese Unsicherheit wird mittels einer Informationsbeschaffungsstrategie unter einen Schwellenwert, der sogenannten hinreichenden Unsicherheit, gesenkt. Damit wird sichergestellt, dass für eine anschließende Inspektion genügend Information bezüglich der zu inspizierenden Oberfläche vorhanden ist.

Eine exakte Lösung des so formulierten Problems erfordert eine Zustandsrückführung, weil aufgrund von hohen Unsicherheiten die Beobachtungen zufällige Realisierungen haben können. Da sowohl die Planung mit Zustandsrückführung als auch eine Approximation, bei der nach jedem Messschritt neu geplant werden muss, nicht praktikabel sind, wird ein Approximationsverfahren vorgeschlagen, das unter Einhaltung der Bedingungen aus Abschnitt 2.5 trotz Unsicherheit einen deterministischen Plan erlaubt.

Mit diesem Kapitel wurde der Grundstein für die folgenden drei Kapitel gelegt, in denen nun die einzelnen Komponenten der Planung das Messmodell, die Modellierung des Beliefstates mit der Zustandstransitionsfunktion in Form des Oberflächenmodells und das Planungsverfahren zur Lösung des Optimierungsproblems vorgestellt werden.

## 3 Sensormodell

In diesem Kapitel wird das Sensormodell vorgestellt, das es ermöglicht, während der Planung für beliebige Sensorkonfigurationen die Messdaten und Unsicherheiten zu präzisieren. Dabei wird der Sensor als ein Smart-Sensor modelliert, der eine zusätzliche Weiterverarbeitung der Messdaten zur Neigung beinhaltet. Die Implementierung des Sensormodells erfolgt als ein Simulator. In Abschnitt 3.1 wird die Modellierung der Messung diskutiert. Die Abschnitte 3.2, 3.3 und 3.4 beschreiben die Komponenten Kamera, Referenzoberfläche und Schirm einer deflektometrischen Messung, die für die Nutzung im Simulator modelliert werden müssen. Abschnitt 3.5 führt eine Parametrisierung der Sensorkonfiguration ein. In Abschnitt 3.6 wird schließlich auf die geometrische Modellierung der Messabbildung eingegangen. Dort wird der deterministische Anteil der Messabbildung beschrieben. In Abschnitt 3.7 wird zunächst die für den stochastischen Anteil der Messabbildung relevante deflektometrische Registrierung eingeführt und in Abschnitt 3.8 die Messunsicherheit behandelt.

### 3.1 Messmodell

Das Messmodell lässt sich durch eine mengenwertige Funktion

$$\underline{\mathcal{Z}} = h(\mathcal{S}, \boldsymbol{\theta}, \mathcal{I}, \underline{\mathbf{w}}) \quad (3.1)$$

beschreiben. Darin sind  $\underline{\mathcal{Z}}$  eine Menge von gemessenen Messstellen und den zugehörigen Neigungen,  $\mathcal{S}$  die zu untersuchende Oberfläche,  $\theta$  die zu einer Messaktion zugehörige Sensorkonfiguration,  $\mathcal{I}$  eine Menge von Musterbildern und  $\underline{w}$  ein Rauschvektor. Es handelt sich hier um einen aktiven Sensor, in dem Sinne, dass mittels eines in die Umgebung ausgesendeten und wieder empfangenen Signals auf das zu messende Objekt geschlossen wird. Grundsätzlich kann das Signal bestehend aus den Musterbildern  $\mathcal{I}$  in die Sensorkonfiguration mit integriert werden, da sie die Messungen beeinflussen. In der Gleichung (3.1) werden sie allerdings getrennt angegeben, da die Planung der Musterbilder in dieser Arbeit nicht betrachtet wird. Dies ist keine große Einschränkung, wenn das Material der Oberfläche homogen ist und keine zu starken konvexen Krümmungen existieren, die das beobachtete Muster unter die Abtastgrenze verkleinern. Das Codierungsverfahren kann in diesem Fall vor der Planung gewählt werden und bleibt über die Teilmessungen konstant. Um die Gleichung (3.1) und weitere folgende übersichtlich zu gestalten, wird in diesem Kapitel die Abhängigkeit der Sensorkonfiguration und der Messdaten vom Messschritt  $k$  nicht explizit angegeben. Außerdem wird der Sensorzustand  $\tilde{\theta}$  ebenfalls nicht mit angegeben, da er durch die zur Messaktion gehörenden Sensorkonfiguration  $\theta$  überschrieben wird und keinen Einfluss auf die Messung hat.

### 3.1.1 Komponenten des Messmodells

Zur Diskussion und Modellierung des Messmodells  $h$  ist es vorteilhaft eine Aufteilung dieser in die drei Komponenten

- physikalische Messung

$$\tilde{\mathcal{I}} = h_p(\mathcal{S}, \theta, \mathcal{I}, \underline{w}_p) \quad (3.2)$$

- deflektometrische Registrierung

$$\underline{l}(u, v) = h_r(\mathcal{I}, \tilde{\mathcal{I}})(u, v) \quad (3.3)$$

- und Bestimmung der Neigung

$$\underline{z} = h_n(\mathcal{S}_r, \underline{l}, \theta, \underline{w}_n) \quad (3.4)$$

vorzunehmen. Die erste Komponente (3.2) beschreibt die eigentliche physikalische Messung. Die Eingabe hierfür besteht sowohl aus der Sensorkonfiguration  $\theta$  als auch aus der Bildsequenz für die Kodierung des Bildschirms  $\mathcal{I}$ . Die Ausgabe des Sensors ist eine Sequenz von Bildern  $\tilde{\mathcal{I}}$ . Das Rauschen bezüglich der Messung ist mit  $\underline{w}_p$  bezeichnet und beinhaltet das Bildrauschen sowie die Positionierungsunsicherheit des Sensors.

Die zweite Komponente (3.3) besteht aus der Berechnung der Zuordnung von Kamerapixeln zu Bildschirmpixeln anhand der gemessenen Bildfolge. Diese Zuordnung wird auch *deflektometrische Registrierung* genannt und wird mit  $\underline{l} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  bezeichnet. Für die Berechnung sind nur die beiden Bildsignale  $\mathcal{I}$  und  $\tilde{\mathcal{I}}$  nötig.

Die dritte Komponente (3.4) berechnet bei gegebener Sensorkonfiguration aus der deflektometrischen Registrierung die Menge der Neigungen auf der Oberfläche in Form der Normalen. Man beachte, dass  $h_n$  demnach eine mengenwertige Funktion ist. Die Berechnung benötigt die Referenzoberfläche  $\mathcal{S}_r$ . Das Rauschen  $\underline{w}_n$  beschreibt die Unsicherheit bezüglich der Kalibrierung. Dieses wird in dieser Arbeit jedoch nicht berücksichtigt und deshalb im Folgenden nicht weiter in der Argumentliste von  $h_n$  mitgeführt.

### 3.1.2 Approximation des Messmodells

Die Modellierung der physikalischen Messung erfordert die Simulation der Lichtausbreitung ausgehend vom Schirm über die reflektierende Oberfläche in die Kamera. Unter anderem spielen dabei die Eigenschaften des Lichts, die Abstrahlcharakteristik des Schirms, die Oberflächeneigenschaften des Prüfobjektes, die Abbildungsoptik und der Sensor der Kamera eine Rolle. Die Physik unterscheidet drei Modelle, mit denen die Lichtausbreitung und ihre Wechselwirkung mit der Materie beschrieben werden können. Diese sind die geometrische Optik, die Wellenoptik und die Quantenoptik. Für den hier betrachteten Anwendungsfall in der Deflektometrie reicht hierbei Erstere aus, da sich hiermit bereits Wechselwirkungen wie Reflexion und Brechung im Makroskopischen beschreiben lassen. Eine Ausnahme bildet die Erzeugung des Bildes am Sensor, wofür die Quantennatur des Lichts berücksichtigt werden muss. Für die geometrische Modellierung ist es nun ausreichend, die Abbildung  $I(u, v)$  bei gegebener Oberfläche und Sensorkonfiguration zu bestimmen. Daraus folgt, dass die erste und zweite Komponente des Messmodells  $h_r$  und  $h_p$  zu einer Abbildung  $h_s$  zusammengefasst werden können, die definiert ist als

$$\underline{l} = h_s(\mathcal{S}, \underline{\theta}, \underline{\mathbf{w}}_s) \approx h_r(\mathcal{I}, h_p(\mathcal{S}, \underline{\theta}, \mathcal{I}, \underline{\mathbf{w}}_p)) \quad (3.5)$$

mit dem Rauschvektor  $\underline{\mathbf{w}}_s$ . Die Planungsverfahren sind unabhängig von der tatsächlichen Modellierung, sodass  $h_s$  bei Bedarf durch eine genauere Modellierung ersetzt werden kann. Jedoch kann eine aufwändigere Methode die Planungszeit erheblich steigern.

Die Menge  $\underline{\mathcal{Z}}$  als Zufallsvariable kann sowohl in der Anzahl der Elemente als auch in den Werten der Elemente variieren. Da sich die Neigungen bei gegebener Oberfläche für jeden Pixel unabhängig voneinander berechnen lassen und das Messrauschen für jeden Pixel als unabhängig angenommen

wird (siehe 3.8 für die Modellierung von  $\underline{w}_s$ ), lässt sich die Abbildung  $h_n$  für jeden Pixel getrennt betrachten.

Unter der Annahme, dass für einen Pixel  $(u, v)$  eine gültige Messung existiert, ist  $\underline{l}(u, v) = h_s(\mathcal{S}, \underline{\theta}, \underline{w}_s)(u, v)$  der zugehörige Messwert und hierfür wird vereinfacht  $\underline{l}(u, v) = h_s(\mathcal{S}, \underline{\theta}, \underline{w}_s, u, v)$  geschrieben. Eine gültige Messung für einen Pixel  $(u, v)$  liegt vor, wenn Licht von einem Schirmpunkt und deren Umgebung über die Oberfläche in das Pixel reflektiert wird und über die deflektometrische Registrierung die Zuordnung zum zugehörigen Schirmpixel berechnet werden kann. Ebenso wird für das Pixel  $(u, v)$  das zugehörige Paar aus Messstelle und Normale  $(\underline{x}, \underline{n}) = h_n(\mathcal{S}_r, \underline{l}, \underline{\theta}, u, v)$  geschrieben. Um die Komponenten des Paares einzeln zu benennen, wird auch die Notation  $\underline{x} = h_n^{(1)}(\mathcal{S}_r, \underline{l}, \underline{\theta}, u, v)$  und  $\underline{n} = h_n^{(2)}(\mathcal{S}_r, \underline{l}, \underline{\theta}, u, v)$  verwendet. Die Menge  $\mathcal{Z}$  setzt sich aus allen gültigen Messungen zusammen.

## 3.2 Kameramodell

Das in dieser Arbeit verwendete Kameramodell beruht auf einer Abbildung mit einer dünnen Linse. Eine Linse wird als dünn bezeichnet, wenn die Dicke der Linse viel kleiner ist als ihre Brennweite. In diesem Fall kann die Brechung der Lichtstrahlen an den Grenzflächen der Linse durch die Brechung an einer Ebene angenähert werden. Bei einer scharfen Abbildung verhält sich die geometrische Abbildung gemäß einer Zentralprojektion durch das optische Zentrum. Diese Abbildung muss noch zusätzlich erweitert werden, weil reale optische Systeme Verzeichnungen aufweisen. Sie bewirken eine systematische Abweichung der Bildpunktposition und müssen daher mit modelliert werden. Zunächst werden in Unterabschnitt 3.2.1 die nötigen Begriffe und Koordinatensysteme eingeführt. Anschließend wird in Unterabschnitt 3.2.2 das geometrische Modell und in Unterabschnitt 3.2.3 die Schärfentiefe beschrieben.

### 3.2.1 Modellbeschreibung

Das Modell der Kamera besteht aus einer dünnen Sammellinse parallel zu einer Bildebene; siehe Abbildung 3.1. Die Ebene, die die dünne Linse approximiert, heißt *Hauptebene*. Linsen werden üblicherweise als rotations-symmetrisch angenommen. Ihre Rotationsachse wird *optische Achse* genannt. Der Schnittpunkt der optischen Achse mit der Hauptebene ist das *optische Zentrum*  $O$ . Die dünne Sammellinse bewirkt, dass parallel zur optischen Achse einfallende Strahlen in einem Brennpunkt, welcher auf der optischen Achse liegt, gebündelt werden. Der Abstand vom Brennpunkt zum optischen Zentrum wird *Brennweite*  $f$  genannt. Der Abstand zwischen Bildebene und Linse wird *Bildweite*  $b$  genannt.

Die Lage der Kamera wird relativ zu einem Elternkoordinatensystem  $\mathcal{P}$  zum Beispiel einem Weltkoordinatensystem  $\mathcal{W}$  angegeben.<sup>1</sup> Das Kamerakoordinatensystem  $\mathcal{C}$  wird durch die Bildebene  $\mathcal{E}_C$  und die optische Achse aufgespannt und hat seinen Ursprung im optischen Zentrum. Der Sensor ist in der Bildebene angebracht und sein Koordinatensystem wird mit  $\mathcal{B}$  gekennzeichnet. Der Bildmittelpunkt wird mit  $\mathbf{o}$  bezeichnet. Die Sensorebene besteht aus Pixeln der Größe  $d_u \times d_v$ , die in einer Gitterstruktur angeordnet sind. Von Interesse ist zusätzlich eine ideale Bildebene, die in einem Abstand von eins zur Hauptebene liegt. Das Koordinatensystem, das durch die Projektion des Kamerakoordinatensystems  $\mathcal{C}$  auf die ideale Bildebene  $\tilde{\mathcal{E}}_C$  entsteht, wird mit  $\tilde{\mathcal{B}}$  bezeichnet.

---

<sup>1</sup> In Abschnitt 3.5 wird eine Sensorkonfigurationsparametrierung gewählt, die als Elternkoordinatensystem statt dem Weltkoordinatensystem  $\mathcal{W}$  ein spezielles Koordinatensystem auf der Prüfoberfläche wählt.



### 3.2.2 Geometrische Abbildung

Die Abbildung eines Objektpunktes auf die Bildebene kann in drei Teilabbildungen  $\Pi_1, \dots, \Pi_3$  getrennt werden. Die Gesamtabbildung ergibt sich durch die Komposition der Projektoren  $\Pi = \Pi_3 \circ \Pi_2 \circ \Pi_1$ . Die Teilabbildungen beschreiben folgende Aspekte der Abbildung:

- Die Abbildung  $\Pi_1 : \mathbb{E}^3 \rightarrow \tilde{\mathcal{E}}_C$  beschreibt die Projektion eines Objektpunktes auf die idealisierte Bildebene im Abstand eins vom Projektionszentrum, wobei  $\mathbb{E}^3$  den dreidimensionalen euklidischen Raum darstellt.
- Die Abbildung  $\Pi_2 : \tilde{\mathcal{E}}_C \rightarrow \tilde{\mathcal{E}}_C$  beschreibt die Verzerrung, die durch die nicht perfekte Linse entsteht.
- Die Abbildung  $\Pi_3 : \tilde{\mathcal{E}}_C \rightarrow \mathcal{E}_C$  schließlich transformiert die Bildpunkte in der idealisierten Bildebene auf die wahre Bildebene.

Der Umweg der Abbildung über die idealisierte Bildebene hat den Vorteil, dass die berechneten Verzeichnungparameter unabhängig von Auflösung und Brennweite der Kamera sind [For12]. Die Parameter der Abbildung  $\Pi_1$  werden als *extrinsische* Parameter bezeichnet, die Parameter der restlichen Abbildungen als *intrinsische* Parameter. Im Folgenden werden die einzelnen Abbildungen näher spezifiziert. Hierbei werden Punkte des euklidischen Raums in homogenen Koordinaten repräsentiert.

Sei ein Punkt  ${}^{\mathcal{P}}\mathbf{x} \in \mathbb{R}^4$  in homogenen Koordinaten relativ zu einem Elternkoordinatensystem  $\mathcal{P}$  gegeben. Sei

$${}^{\mathcal{P}}\mathbf{R}_e = \begin{pmatrix} {}^{\mathcal{P}}\mathbf{u}_C & {}^{\mathcal{P}}\mathbf{v}_C & {}^{\mathcal{P}}\mathbf{n}_C \end{pmatrix} \in \mathbb{R}^{3 \times 3} \quad (3.6)$$

die Rotationsmatrix, deren Spalten die Achsen des Kamerakoordinatensystems im Elternkoordinatensystem beinhalten, und  ${}^c\mathbf{t}_P \in \mathbb{R}^3$  der Ortsvektor

des Elternkoordinatensystems Ursprungs relativ zum Kamerakoordinatensystem. Für die gesamte Abbildung  $\Pi_1$  ergibt sich:

$$\Pi_1({}^{\mathcal{P}}\mathbf{x}) = \begin{pmatrix} -\mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 1} & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & 1 & 0 \end{pmatrix} \begin{pmatrix} {}^{\mathcal{C}}\mathbf{R}_{\mathcal{P}} & {}^{\mathcal{C}}\mathbf{t}_{\mathcal{P}} \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix} {}^{\mathcal{P}}\mathbf{x}, \quad (3.7)$$

mit  ${}^{\mathcal{C}}\mathbf{R}_{\mathcal{P}} = {}^{\mathcal{P}}\mathbf{R}_{\mathcal{C}}^T$  und der Einheitsmatrix  $\mathbf{I}_{2 \times 2} \in \mathbb{R}^{2 \times 2}$ . In Gleichung (3.7) wird der Punkt  ${}^{\mathcal{P}}\mathbf{x}$  in das Kamerakoordinatensystem transformiert und anschließend über die Zentralprojektion auf die ideale Bildebene abgebildet.

Die radiale Verzerrung kann durch

$$\Pi_2(\mathbf{x}) = \mathbf{x} + ar^2(\mathbf{x} - \mathbf{o}) \quad (3.8)$$

modelliert werden, wobei  $r^2 = \|\mathbf{x} - \mathbf{o}\|^2$  den quadratischen Abstand des Pixels zum Bildmittelpunkt  $\mathbf{o}$  darstellt. Radiale Verzerrungen bewirken, dass Geraden auf Kurven abgebildet werden.

Die Pixelposition auf dem Sensor wird anhand der Transformation

$$\Pi_3(\tilde{\mathcal{B}}\mathbf{x}) = \begin{pmatrix} \frac{b}{d_u} & 0 & o_u \\ 0 & -\frac{b}{d_v} & -o_v \\ 0 & 0 & 1 \end{pmatrix} \tilde{\mathcal{B}}\mathbf{x} \quad (3.9)$$

berechnet, wobei  $o_u$  und  $o_v$  die Komponenten des Bildmittelpunktes  $\mathbf{o}$  relativ zum Koordinatensystem  $\tilde{\mathcal{B}}$  sind. Die Transformation setzt sich aus einer Verschiebung des Ursprungs in die linke obere Ecke des Sensors und einer Skalierung aufgrund der Bildweite und der Pixelgröße zusammen.

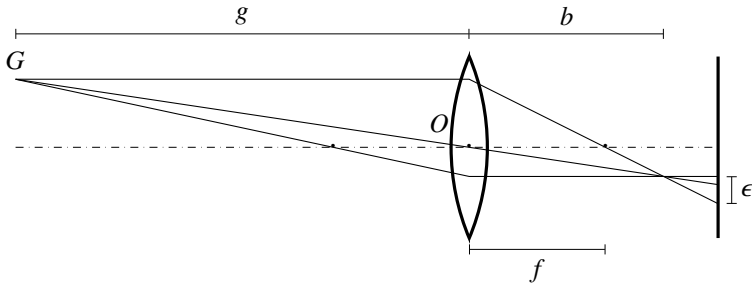


Abbildung 3.1: Unschärfescheibe auf dem Kamerasensor.

### 3.2.3 Schärfentiefe

Das Objektiv wird durch eine dünne Linse modelliert. Damit lässt sich der Schärfentiefebereich ermitteln. In diesem Modell kann die Linse nur Punkte auf einer Ebene scharf abbilden. Eine Kamera mit Bildweite  $b$  und Brennweite  $f$  bildet die Ebene im Abstand  $g$  scharf ab; siehe Abbildung 3.1. Der Zusammenhang dieser Größen ist durch

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{b} \quad (3.10)$$

gegeben. Die Punkte, die nicht auf der Ebene liegen, werden auf eine Unschärfescheibe abgebildet. In der Anwendung kann häufig eine kleine Unschärfescheibe toleriert werden, beispielsweise wenn die Unschärfescheibe kleiner als die Pixelabstand ist. Der Abstand von  $g$ , bei dem Punkte auf eine Unschärfescheibe mit dem Radius  $\epsilon$  abgebildet werden, lässt sich über die Gleichung

$$\Delta g(\epsilon) = \frac{\epsilon g}{\frac{Df}{g-f} - \epsilon} \quad (3.11)$$

bestimmen, wobei  $D$  der Durchmesser der kreisförmigen Eintrittspupille ist. Bei gegebener Toleranz  $\epsilon$  lässt sich der Bereich angeben, indem das Objekt hinreichend scharf abgebildet wird [Bey15].

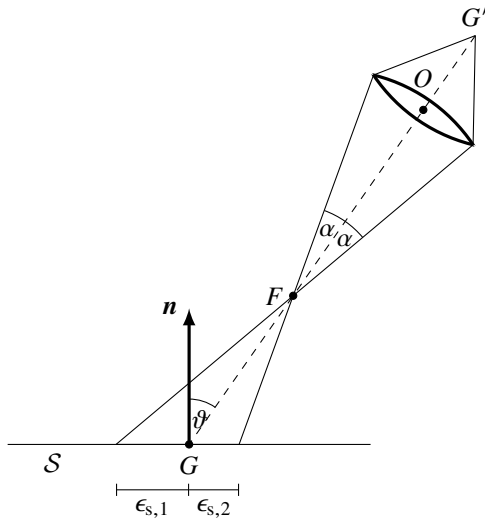


Abbildung 3.2: Unschärfescheibe auf der Oberfläche

Mit einer ähnlichen geometrischen Betrachtung kann man die Bereiche der Oberfläche, die auf einen Punkt abgebildet werden, bestimmen. Dazu betrachtet man den inversen Strahlenverlauf von der Kamera zum Schirm. Sei  $\Delta g = \overline{GF}$  der Abstand zwischen Objekt und objektseitiger Schärfeebene und  $\vartheta$  der Winkel zwischen der optischen Achse der Kamera und der Oberfläche; siehe Abbildung 3.2, dann ergibt sich der Unsicherheitsbereich  $\epsilon_s$  auf der Oberfläche aus der Summe

$$\epsilon_s = \epsilon_{s,1} + \epsilon_{s,2}, \quad (3.12)$$

wobei die Summanden sich aus

$$\epsilon_{s,1} = \frac{\sin(\alpha)}{\cos(\alpha + \vartheta)} \Delta g \quad (3.13)$$

$$\epsilon_{s,2} = \frac{\sin(\alpha)}{\cos(-\alpha + \vartheta)} \Delta g \quad (3.14)$$

berechnen lassen. Hierbei ist  $\tan \alpha = \frac{D}{2OF}$ .

### 3.3 Referenzoberflächenmodell

Bei der Oberflächeninspektion kommt ein Oberflächenmodell an mehreren Stellen zum Einsatz. Zum einen bei der Schätzung der Oberflächenabweichung aus den Messdaten, zum anderen während der Planung zur Prädiktion der Messdaten. In diesem Abschnitt geht es um Letzteres. Für die Modellierung der Referenzoberfläche lassen sich zwei Komponenten unterscheiden. Diese sind die Gestalt der Oberfläche und die Reflektanz der Oberfläche [Wer11]. Die Gestalt der Oberfläche lässt sich durch ein Oberflächenmodell beschreiben. Speziell für die Anwendung der Planung und Simulation werden folgende Anforderungen an das Oberflächenmodell gestellt:

- Es sollte eine effiziente Schnittpunktberechnung möglich sein, da diese Operation in der Simulation für jeden Pixel der Kamera mindestens einmal ausgeführt wird.
- Das Modell sollte die wahre Oberfläche ausreichend genau repräsentieren. Beispielsweise kann es auch nötig sein,  $C^2$ -Stetigkeit voraussetzen, um die Kompatibilität zu den CAD-Modellen zu gewährleisten: In der Regel sind die in der Industrie verwendeten Modelle  $C^2$ -stetig, weil ihre Formen meist optisch ansprechender sind [Kam04].

Ein Vergleich von unterschiedlichen Oberflächenmodellen ist in [Kam04] gegeben. Generell ist es sinnvoll, die Repräsentation zu verwenden, in der das CAD-Modell gegeben ist.

Die zweite Komponente für die Modellierung der Oberfläche, die Reflektanz der Oberfläche, lässt sich über die sogenannte *bidirektionale Reflektanzverteilungsfunktion* (BRDF) modellieren. Die BRDF beschreibt die reflektierte Strahldichte  $dL_r$  in eine Richtung  $(\phi_r, \vartheta_r)$  in Abhängigkeit von der einfallenden Bestrahlungsstärke  $dE_e$  aus der Richtung  $(\phi_e, \vartheta_e)$ , wobei  $\phi_r, \phi_e$  die Azimutwinkel und  $\vartheta_r, \vartheta_e$  die Elevationswinkel darstellen:

$$BRDF(\phi_r, \vartheta_r, \phi_e, \vartheta_e) = \frac{dL_r(\vartheta_r, \phi_r)}{dE_e(\vartheta_e, \phi_e)}. \quad (3.15)$$

Die hier vorgestellte BRDF ist eine vierdimensionale Funktion. Umfangreichere Modelle berücksichtigen zusätzlich weitere Abhängigkeiten wie die Wellenlänge oder den Ort. Die Vermessung solch einer BRDF für die Anwendung in der Planung erfordert aufgrund der hohen Dimension viele Abtastpunkte.

In dieser Arbeit wird eine perfekt spiegelnde Reflexion angenommen:

$$BRDF(\phi_r, \vartheta_r, \phi_e, \vartheta_e) \propto \delta(\phi_r - \phi_e - \pi)\delta(\vartheta_r - \vartheta_e). \quad (3.16)$$

Das Reflexionsverhalten teilspeiegender rauer Oberflächen kann hiermit nicht modelliert werden. Diese bewirken allerdings eine Kontrast- und Helligkeitsverminderung des beobachteten Musters, die einen wesentlichen Einfluss auf die Messunsicherheit haben. Aus diesem Grund muss diese Beschränkung bei der Schätzung der Unsicherheit anderweitig kompensiert werden; siehe Unterabschnitt 3.8.1.

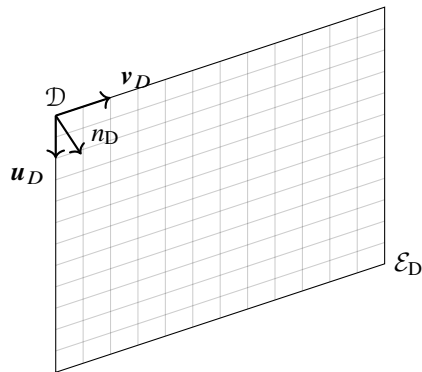


Abbildung 3.3: Bildschirmmodell

## 3.4 Schirmmodell

Für den Schirm wird gewöhnlich eine LC-Display gewählt, da es einen einfachen Weg bietet, eine planare, hochauflösende Musteranzeige zu realisieren. Relevant für die Deflektometrie sind eine hohe Helligkeit des Schirms sowie eine möglichst richtungsunabhängige Abstrahlcharakteristik. Das hier betrachtete Modell berücksichtigt allerdings nur die geometrischen Eigenschaften des LC-Displays.

### 3.4.1 Geometrische Modellierung

Die Größe des bildgebenden Bereichs des Bildschirms wird eben und rechteckig modelliert mit der Länge  $L_u$  und Höhe  $L_v$ . Die geometrischen Parameter des Bildschirms bestehen aus einer Rotation und Translation relativ zu einem Elternkoordinatensystem. Dieses System ist bei einem kompakten Sensor das Koordinatensystem der Kamera, ansonsten ist es das Weltkoordinaten- oder Objektkoordinatensystem.

Das Bildschirmkoordinatensystem  $\mathcal{D}$  liegt in der linken oberen Ecke des Schirms; siehe Abbildung 3.3. Sei

$${}^{\mathcal{P}}\mathbf{R}_{\mathcal{D}} = \begin{pmatrix} {}^{\mathcal{P}}\mathbf{u}_{\mathcal{D}} & {}^{\mathcal{P}}\mathbf{v}_{\mathcal{D}} & {}^{\mathcal{P}}\mathbf{n}_{\mathcal{D}} \end{pmatrix} \in \mathbb{R}^{3 \times 3} \quad (3.17)$$

die Rotationsmatrix, deren Spalten die Koordinatenachsen des Schirms in Elternkoordinaten beinhaltet und  ${}^{\mathcal{P}}\mathbf{t}_{\mathcal{D}} \in \mathbb{R}^3$  die Verschiebung des Schirmkoordinatenursprungs relativ zum Elternkoordinatensystem, dann ist die homogene Matrix der Koordinatentransformation

$${}^{\mathcal{P}}\mathbf{H}_{\mathcal{D}} = \begin{pmatrix} {}^{\mathcal{P}}\mathbf{R}_{\mathcal{D}} & {}^{\mathcal{P}}\mathbf{t}_{\mathcal{D}} \\ \mathbf{0} & 1 \end{pmatrix}. \quad (3.18)$$

Die Bildelebene  $\mathcal{E}_{\mathcal{D}}$  ist durch die Gleichung

$$\mathcal{E}_{\mathcal{D}} = \{ \mathbf{x} \mid \mathbf{n}_{\mathcal{D}}^T (\mathbf{x} - \mathbf{t}_{\mathcal{D}}) = 0 \} \quad (3.19)$$

gegeben, wobei  $\mathbf{n}_{\mathcal{D}}$  die Normale der Schirmebene ist.

### 3.4.2 Weitere Einflussgrößen

Im Folgenden wird eine Auswahl an Einflussgrößen diskutiert, die die Qualität der deflektometrischen Registrierung beeinflussen, die aber entweder nicht modelliert sind oder durch eine Kalibrierung oder Korrektur vor der Planung kompensiert werden können.

- Die Abstrahlcharakteristik des Schirms wird in dem Modell nicht direkt berücksichtigt. Dennoch spielt sie eine wichtige Rolle. Die Helligkeit und der Kontrast des Bildschirms nehmen mit flachem Sichtwinkel ab. In der Simulation wird diese durch einen Grenzwinkel modelliert, der den Sichtwinkel auf den Schirm eingrenzt.



- Die Annahme der Ebenheit des Schirms kann in der Realität verletzt sein und führt zu systematischen Fehlern in der Dekodierung. Ist die Abweichung bekannt, kann diese berücksichtigt werden, indem anstelle einer Schirmebene  $\mathcal{E}_D$  eine Schirmoberfläche modelliert wird. Hiermit muss gleichzeitig auch die Schnittpunktberechnung mit dem Schirm und die Bestimmung der Pixelposition aus dem Schnittpunkt angepasst werden.
- Petz [Pet06] beschreibt den systematischen Fehler, der durch die Lichtbrechung an den transparenten Schichten des LCDs eintritt. Er schätzt diesen Fehler als hoch ein. Bei bekannten Brechungsindizes und Schichtdicken kann eine Korrektur berechnet werden.
- Die nichtlineare Systemantwort des Schirms führt bei Phasenschiebverfahren zu systematischen Fehlern in der Dekodierung. Mittels einer Kalibrierung kann die Bildschirmkennlinie ermittelt und bei der Musteranzeige berücksichtigt werden; siehe [Pet06, Rap11].

### 3.5 Sensoraufbauten und Konfiguration

In der Praxis sind generell mehrere Aufbauten realisierbar. Günstig ist ein Aufbau, bei dem sowohl der Schirm als auch die Kamera unabhängig voneinander bewegt werden können. Durch diese Flexibilität ist es möglich, auch schwierige Bereiche der Oberfläche zu beobachten und größere Messbereiche zu realisieren. Der Nachteil dieses Aufbaus ist, dass zwei Handhabungswerkzeuge jeweils für die Kamera und den Schirm benötigt werden. Üblich ist ein kompakter Aufbau, bei dem die Kamera und der Schirm starr zueinander angebracht sind. Die relative Position von Kamera und Schirm ist damit vor der Messung durch den Sensor bestimmt. Eine weitere Möglichkeit ist die Lage des Schirms zu fixieren und die Kamera frei

bewegbar zu machen. Solche Aufbauten bieten sich bei größeren Schirmen an, wie sie zum Beispiel in einer Cave [Bal14] eingesetzt werden.

Die geometrischen Freiheitsgrade sind die Position und Orientierung des Schirms sowie der Kamera. Beim Aufbau mit unabhängiger Kamera und Schirm entspricht das zwölf Freiheitsgraden. Für den Aufbau mit kompakten Sensor oder fixem Schirm ergeben sich sechs Freiheitsgrade. Eine Vielzahl von Parametrierungen sind für die Lage des Sensors möglich. Im Folgenden wird eine Parametrierung der Konfiguration für den uneingeschränkten Sensor definiert. Für den kompakten Sensor kann die Komponente des Schirms weggelassen werden, da diese während der Messung konstant bleibt und relativ zur Kamera definiert ist.

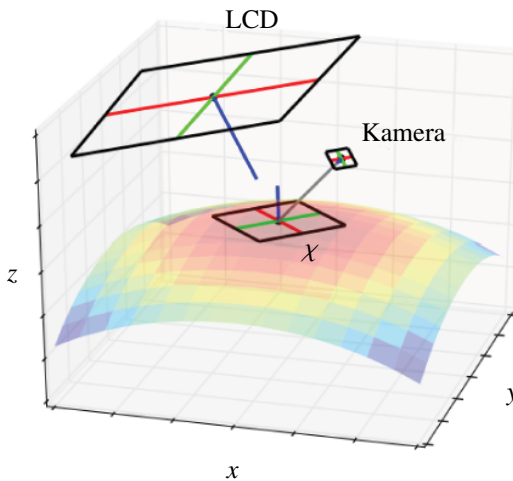


Abbildung 3.4: Schematische Darstellung der Lage eines Sensors relativ zu einer Oberfläche, dargestellt sind die Koordinatensysteme des Bildschirms, der Kamera und das lokale Koordinatensystem am Sichtpunkt  $\chi$ .

Die Parametrierung ist ein zwölfdimensionaler Vektor

$$\boldsymbol{\theta} = (\boldsymbol{\chi}^T, \phi_c, \vartheta_c, d_c, \tau_c, \phi_1, \vartheta_1, d_1, \alpha_1, \beta_1, \gamma_1)^T \quad (3.20)$$

mit

- einer Stelle  $\boldsymbol{\chi} \in \mathcal{S}_r$  auf der Referenzoberfläche,
- der Kamerakonfiguration  $\phi_c \in [0, 2\pi)$ ,  $\vartheta_c \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ ,  $d_c \in \mathbb{R}^+$  und  $\tau_c \in [0, 2\pi)$
- der Schirmposition  $\phi_1 \in [0, 2\pi)$ ,  $\vartheta_1 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  und  $d_1 \in \mathbb{R}^+$
- und der Schirmorientierung  $\alpha_1 \in [-\pi, \pi)$ ,  $\beta_1 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$  und  $\gamma_1 \in [-\pi, \pi)$ .

Im Folgenden werden die Komponenten des Parametervektors näher erläutert. Die ersten zwei Parameter bestehen aus der Stelle  $\boldsymbol{\chi}$ . Falls sich die Oberfläche als eine Funktion  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$  modellieren lässt, dann gilt  $\boldsymbol{\chi} \in \text{supp}\{f\} \subset \mathbb{R}^2$ , wobei  $(\boldsymbol{\chi}^T, f(\boldsymbol{\chi}))^T$  einen Punkt auf der Referenzoberfläche beschreibt. Er wird im folgenden *Sichtpunkt* genannt. An diesem Punkt kann ein Koordinatensystem  $\mathcal{L}_{\boldsymbol{\chi}}$  definiert werden, das die intrinsischen lokalen Eigenschaften der Oberfläche beschreibt. Diese wird durch den Tangentenvektor in Richtung des höchsten Anstiegs  $\mathbf{g}_{\boldsymbol{\chi}}$  und dem Normalenvektor  $\mathbf{n}_{\boldsymbol{\chi}}$  an dem Punkt anhand des Festsetzens einer Orientierung für das Koordinatensystem eindeutig bestimmt. Die homogene Koordinatentransformation des Koordinatensystems lautet

$$\mathbb{W}_{\mathbf{H}_{\mathcal{L}_{\boldsymbol{\chi}}}} = \begin{pmatrix} \mathbf{n}_{\boldsymbol{\chi}} & \mathbf{g}_{\boldsymbol{\chi}} \times \mathbf{n}_{\boldsymbol{\chi}} & \mathbf{g}_{\boldsymbol{\chi}} & \mathbf{0} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3.21)$$

Die  $x$ -Achse zeigt in Richtung des Vektors  $\mathbf{g}_{\boldsymbol{\chi}}$ . Falls der Gradient Null ist, dann wird  $\mathbf{g}_{\boldsymbol{\chi}}$  gleich der  $x$ -Achse gewählt. Die  $z$ -Achse zeigt in Richtung des Oberflächennormalenvektors  $\mathbf{n}_{\boldsymbol{\chi}}$  und die  $y$ -Achse wird orthogonal zu

den beiden anderen Richtungen gewählt. Hier gibt es zwei Möglichkeiten, die unterschiedlich orientierte Koordinatensysteme erzeugen. Alle weiteren Parameter werden relativ zu diesem Koordinatensystem definiert; siehe Abbildung 3.4 für eine schematische Darstellung. Das führt dazu, dass die Sensorkonfigurationen invariant bezüglich Rotationen und Translation der Oberfläche sind. Außerdem ist eine Invarianz bezüglich der lokalen Neigung gegeben. Falls an zwei Orten eine ähnliche Neigung vorhanden ist, zum Beispiel bei symmetrischen Oberflächen, können Konfigurationen durch Veränderung von  $\chi$  von einem Ort zum anderen transportiert werden.

Für die Kamera wird die Bedingung gestellt, dass ihre optische Achse die Referenzoberfläche in dem Sichtpunkt schneidet. Die Kamera wird parametrisiert durch den Elevations-  $\vartheta_c$ , dem Azimuthwinkel  $\phi_c$  und den Abstand relativ zum Sichtpunkt  $d_c$ . Der weitere freie Parameter ist die Rotation um die optische Achse der Kamera  $\tau_c$ . Für das Koordinatensystem der Kamera ergibt sich

$${}^{\mathcal{L}_\chi} \mathbf{H}_c = \mathbf{R}_2(\phi_c) \mathbf{R}_1(\vartheta_c) \mathbf{R}_3(\tau_c) \mathbf{T}_3(-d_c) \quad (3.22)$$

Dabei repräsentieren die homogenen Matrizen  $\mathbf{R}_i(\phi)$  Rotationen um die Einheitsachse  $\mathbf{e}_i$  um einem Winkel  $\phi$ , wobei  $i \in \{1, 2, 3\}$ . Sie sind durch

$$\mathbf{R}_1(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3.23)$$

$$\mathbf{R}_2(\phi) = \begin{pmatrix} \cos(\phi) & \sin(\phi) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.24)$$

und

$$\mathbf{R}_3(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.25)$$

definiert. Die homogenen Matrizen  $\mathbf{T}_i(t)$  repräsentieren eine Verschiebung entlang der Achse  $\mathbf{e}_i$  um einen Betrag  $t$ . Sie sind durch

$$\mathbf{T}_i(t) = \begin{pmatrix} \mathbf{I}_{3 \times 3} & t\mathbf{e}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix} \quad (3.26)$$

gegeben mit  $i \in \{1, 2, 3\}$ .

Die Position des Bildschirms wird durch sphärische Koordinaten  $(\phi_1, \vartheta_1, d_1)$  und seine Orientierung durch eulersche Winkel  $(\alpha_1, \beta_1, \gamma_1)$  parametrisiert. Für das Koordinatensystem des Schirms erhält man

$$\mathcal{L}_x \mathbf{H}_{\mathcal{D}} = \begin{pmatrix} 1 & 0 & 0 & d_1 \cos(\phi_1) \sin(\vartheta_1) \\ 0 & 1 & 0 & d_1 \sin(\phi_1) \sin(\vartheta_1) \\ 0 & 0 & 1 & d_1 \cos(\vartheta_1) \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{R}_2(\alpha_1) \mathbf{R}_1(\beta_1) \mathbf{R}_3(\gamma_1). \quad (3.27)$$

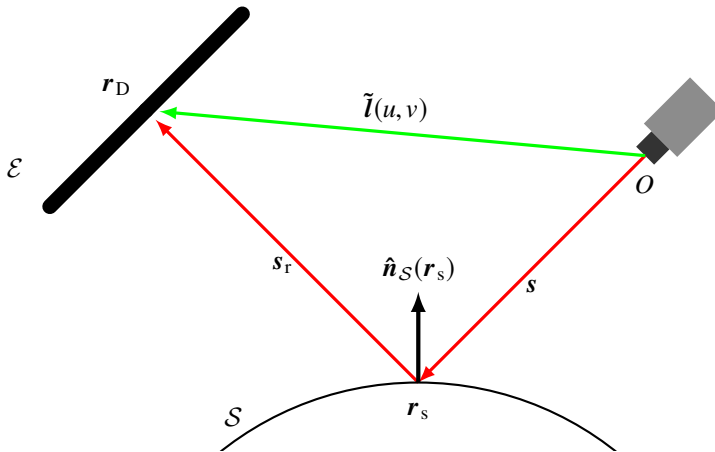


Abbildung 3.5: Schematische Darstellung zur Bestimmung des Normalenvektors aus einer deflektometrischen Messung

Im Vergleich zur Kamera ist der Bildschirm nicht an den Sichtpunkt gekoppelt. Die Parameter für die Position und Orientierung des Bildschirms sind beim kompakten Sensor konstant.

## 3.6 Geometrisches Modell der deflektometrischen Messung

Dieser Abschnitt widmet sich der Berechnung der Zuordnung von Kamera-pixeln zu Schirmpixeln  $l : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , der Neigungen und der zugehörigen Messstelle. Bei gegebener Position und Orientierung des Schirms relativ zur Kamera, lässt sich auf die Abbildung  $\tilde{l} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$  schließen, die einen Vektor im euklidischen Raum vom Projektionszentrum der Kamera  $O$  zum Schirmpixel darstellt. Die folgenden Betrachtungen erfolgen für einen Pixel mit den Koordinaten  $(u, v)$  und für den Fall, dass das Pixel eine gültige

Messung repräsentiert. Eine schematische Darstellung der geometrischen Zusammenhänge ist in Abbildung 3.5 gegeben.

### 3.6.1 Zuordnung zwischen Kamerapixel und Schirmpixel

Die Berechnung der Zuordnung zwischen Kamerapixel und Bildpunkt erfolgt durch die Verfolgung der Lichtstrahlen in umgekehrter Strahlrichtung von der Kamera zum Schirm. Der Sichtstrahl von einem Kamerapixel in Richtung der Oberfläche wird durch die Gleichung

$$\mathcal{R}_i = \{ \mathbf{t}_C + \alpha \hat{\mathbf{s}} \mid \alpha \in \mathbb{R}^+ \} \quad (3.28)$$

beschrieben. Dieser trifft auf die Oberfläche  $\mathcal{S}$  im Schnittpunkt  $\mathbf{r}_s \in \mathcal{R}_i \cap \mathcal{S}$ . Dort wird der Strahl gemäß dem Reflexionsgesetz abgelenkt. Die Richtung des reflektierten Strahls berechnet sich durch

$$\hat{\mathbf{s}}_r = \hat{\mathbf{s}} - 2\hat{\mathbf{n}}_{\mathcal{S}}(\mathbf{r}_s)\hat{\mathbf{n}}_{\mathcal{S}}(\mathbf{r}_s)^T \hat{\mathbf{s}}, \quad (3.29)$$

wobei  $\hat{\mathbf{n}}_{\mathcal{S}}(\mathbf{r}_s)$  der Normalenvektor am Schnittpunkt ist. Hiermit ergibt sich für den reflektierten Strahl

$$\mathcal{R}_o = \{ \mathbf{r}_s + \alpha \hat{\mathbf{s}}_r \mid \alpha \in \mathbb{R}^+ \}. \quad (3.30)$$

Weiter trifft der reflektierte Strahl  $\mathcal{R}_o$  auf die Schirmebene. Ihr Schnittpunkt  $\mathbf{r}_D \in \mathcal{R}_o \cap \mathcal{E}_D$  ergibt sich durch Einsetzen der Geradengleichung (3.30) in die Schirmebenengleichung (3.19). Nach Auflösen der so erhaltenen Gleichung nach dem Abstand von der Oberfläche zum Schirm ergibt sich weiter für den Schnittpunkt

$$\mathbf{r}_D = \mathbf{r}_s + s_r \quad \text{mit} \quad s_r = \frac{\hat{\mathbf{n}}_D^T (\mathbf{r}_s - \mathbf{t}_D)}{\hat{\mathbf{n}}_D^T \hat{\mathbf{s}}_r} \hat{\mathbf{s}}_r. \quad (3.31)$$

Diese Gleichung hat eine Lösung, falls die Richtung des Strahls  $\hat{\mathbf{s}}_r$  und der Normalenvektor des Schirms  $\hat{\mathbf{n}}_D$  nicht senkrecht zueinanderstehen.

Für die deflektometrische Registrierung ergibt sich schließlich

$$\mathbf{l}(u, v) = h_s(\mathcal{S}, \boldsymbol{\theta}, \mathbf{0}, u, v) = \begin{pmatrix} \mathbf{u}_D^T(\mathbf{r}_D(u, v) - \mathbf{t}_D) \\ \mathbf{v}_D^T(\mathbf{r}_D(u, v) - \mathbf{t}_D) \end{pmatrix}. \quad (3.32)$$

### 3.6.2 Bestimmung der Neigung

Ist die Kalibrierung des Sensors bekannt, lässt sich aus  $\mathbf{l}$  der Vektor  $\tilde{\mathbf{l}} \in \mathbb{R}^3$  vom Projektionszentrum der Kamera zum Schirmpixel berechnen. Aus diesem Vektor und der Richtung des Sichtstrahls  $\hat{\mathbf{s}}$  lässt sich auf die Neigung der Oberfläche schließen. Da der Abstand von der Kamera zum Schnittpunkt des Sichtstrahls mit der Oberfläche unbekannt ist, erhält man ein Normalenfeld im gesamten Sichtfeld der Kamera. Wird ein Punkt  $\mathbf{r}$  auf einen Kamerapunkt  $\mathbf{r}_C$  abgebildet, dann ergibt sich für den Sichtstrahl und den reflektierten Strahl

$$\mathbf{s}(\mathbf{r}) = \mathbf{r} - \mathbf{r}_C \quad (3.33)$$

$$\mathbf{s}_r(\mathbf{r}) = \mathbf{r} - \tilde{\mathbf{l}}(\mathbf{r}_C) \quad (3.34)$$

und damit für den Normalenvektor

$$\hat{\mathbf{n}}(\mathbf{r}) = \frac{\hat{\mathbf{s}}(\mathbf{r}) - \hat{\mathbf{s}}_r(\mathbf{r})}{\|\hat{\mathbf{s}}(\mathbf{r}) - \hat{\mathbf{s}}_r(\mathbf{r})\|}. \quad (3.35)$$

Dieses Normalenfeld hat die Eigenschaft, dass ihre Normalenvektoren an den Stellen, an denen die Oberfläche verläuft, mit den Normalenvektoren der Oberfläche übereinstimmen.



Die Bestimmung der Oberfläche anhand dieser Information ist nicht eindeutig und die Lösungsmenge ist eine eindimensionale Mannigfaltigkeit; siehe [Bal08].<sup>2</sup> Um Eindeutigkeit zu erreichen, muss Zusatzwissen genutzt werden. Dieser Vorgang wird auch *Regularisierung* genannt. Eine mögliche Form der Regularisierung ist über Linearisierung [Wer07]. Hierzu wird eine Referenzoberfläche genutzt, um den Abstand des Pixels zum Sichtpunkt auf der Oberfläche entlang des Sichtstrahls zu approximieren. Sei  $\tilde{\mathbf{r}}_s(u, v) \in \mathcal{R}_i \cap \mathcal{S}_r$  der Schnittpunkt des Strahls vom Kamerapixel  $(u, v)$  mit der Referenzoberfläche, dann erhält man mittels dieser Regularisierung für die Messstellen des Normalenvektors:

$$\mathbf{h}_n^{(1)}(\mathcal{S}_r, \underline{\mathbf{l}}, \boldsymbol{\theta}, u, v) = \tilde{\mathbf{r}}_s(u, v), \quad (3.36)$$

$$\mathbf{h}_n^{(2)}(\mathcal{S}_r, \underline{\mathbf{l}}, \boldsymbol{\theta}, u, v) = \hat{\mathbf{n}}_{\mathcal{S}_r}(\tilde{\mathbf{r}}_s(u, v)) \quad (3.37)$$

für die Messkomponente  $\mathbf{h}_n$ . In der Gleichung (3.37) wird direkt der Normalenvektor aus der Referenzoberfläche bestimmt, da der Sensor als ein Smart-Sensor modelliert wurde und damit die inverse Berechnung von der deflektometrischen Registrierung zum Normalenvektor im Messmodell enthalten ist.

Mit der Annahme, dass die Referenzoberfläche als eine Funktion  $f_r(x, y)$  darstellbar ist, kann der gemessene Gradient  $\mathbf{q}$  mit  $\hat{\mathbf{n}} = \hat{\mathbf{n}}_{\mathcal{S}_r}(\tilde{\mathbf{r}}_s(u, v))$  wie folgt berechnet werden:

$$\mathbf{q}(\mathbf{r}, \underline{\mathbf{l}}, \boldsymbol{\theta}) = - \begin{pmatrix} \hat{n}_1(\mathbf{r})/\hat{n}_3(\mathbf{r}) \\ \hat{n}_2(\mathbf{r})/\hat{n}_3(\mathbf{r}) \end{pmatrix}.$$

<sup>2</sup> In neueren Arbeiten wird zusätzlich ausgenutzt, dass das Gradientenfeld der Oberfläche rotationsfrei sein muss [Liu15, Pak14]. Mit diesem zusätzlichen Wissen ist eine eindeutige Lösung der deflektometrischen Rekonstruktion möglich. Allerdings wurde diese Methodik noch nicht ausreichend in der Praxis untersucht.

## 3.7 Deflektometrische Registrierung

Das Ziel der deflektometrischen Registrierung ist eine eindeutige Zuordnung zwischen den Kamerapixeln und den Bildschirmpixeln herzustellen. Zwar wird die deflektometrische Registrierung im vorgestellten Sensormodell nicht zur Berechnung von  $I(u, v)$  verwendet, da bei der betrachteten Approximation die Schnittpunkte der Sichtstrahlen mit dem Bildschirm direkt bestimmt werden, aber sie ist relevant für die Diskussion des Phasenausbaus. Die Zuordnung  $I(u, v)$  lässt sich durch eine zeitliche Kodierung der Bildschirmpixel mittels Bildsequenzen erreichen. Beispiele für zeitliche Kodierungsverfahren sind Gray-Code oder Phasenschiebeverfahren; siehe [Kam04] für einen Überblick über zeitliche Kodierungsverfahren für die Deflektometrie und [Sal04] für einen umfassenderen Überblick über Kodierungsverfahren generell. In der Deflektometrie wird häufig das Phasenschiebeverfahren verwendet, da es robust gegenüber unscharfer Abbildung ist und Subpixel-Genauigkeit erreichen kann.

### 3.7.1 Phasenschiebeverfahren

Beim Phasenschiebeverfahren werden die beiden Achsen der Bildebene unabhängig voneinander kodiert. Hier wird nur die Kodierung der Spalten betrachtet. Die Kodierung der Zeilen erfolgt analog dazu. Jedem Pixel in einer Zeile wird eine Phase zugeordnet:

$$\phi(u) = 2\pi \frac{M_p}{L_u} u, \quad (3.38)$$

wobei  $M_p$  die maximale Anzahl der sich wiederholenden Phasen auf dem Schirm angibt (das heißt für  $M_p > 1$  erfolgt eine uneindeutige Zuordnung) und  $L_u$  die Breite des Schirms ist. Die Anzeige einer Mustersequenz mit  $M$  Mustern, ermöglicht eine zeitliche Kodierung der Pixel. Die Grauwerte mit

dem ein Pixel kodiert wird, werden durch verschobene Kosinus-Funktionen mit zugehöriger Phase  $\phi$  beschrieben:

$$\bar{g}_k = g_{\text{sat}} \bar{\beta} (1 + \bar{\gamma} \cos(\phi + \psi_k)), \quad k = 1, \dots, M. \quad (3.39)$$

Dabei ist  $g_{\text{sat}}$  der maximale Grauwert,  $\bar{\beta} \in [0, 1]$  die relative Helligkeit und  $\bar{\gamma} \in [0, 1]$  der relative Kontrast des angezeigten Musters. Damit der Grauwert die Bedingung  $0 \leq \bar{g}_k \leq g_{\text{sat}}$  erfüllt, muss für die relative Helligkeit und den relativen Kontrast die Beziehung  $\bar{\beta}(1 + \bar{\gamma}) \leq 1$  gelten. Ein Muster mit höchstmöglichem Kontrast wird mit  $\bar{\gamma} = 1, \bar{\beta} = \frac{1}{2}$  erreicht. Bei der Menge  $\{\psi_1, \dots, \psi_M\}$  mit  $\psi_1 = 0$  handelt es sich um die Phasenverschiebungen. Man unterscheidet Phasenschiebeverfahren nach der Anzahl der Verschiebungen und ob eine gleichmäßige oder ungleichmäßige Verschiebung vorliegt. Eine große Klasse von Phasenschiebealgorithmen sind die sogenannten symmetrischen  $M$ -Schritt Algorithmen. Hierbei werden die Verschiebungsschritte äquidistant aus dem Intervall  $[0, 2\pi)$  gewählt

$$\psi_k = \frac{2\pi}{M}(k - 1), \quad k \in \{1, \dots, M\}. \quad (3.40)$$

An einer Pixelposition auf der Kamera werden  $M$  Grauwerte gemessen

$$g_k \approx g_{\text{sat}} \beta (1 + \gamma \cos(\phi + \psi_k)), \quad k = 1, \dots, M. \quad (3.41)$$

Idealerweise ändert sich an dem Signal nur der Kontrast und die Helligkeit zu  $\gamma, \beta$ . Die Phase kann mit folgendem Schätzer bestimmt werden:

$$\hat{\phi} = \arctan \left( \frac{\sum_{k=1}^M g_k \sin(\psi_k)}{\sum_{k=1}^M g_k \cos(\psi_k)} \right), \quad (3.42)$$

Aus der berechneten Phase lässt sich anhand (3.38) auf die Position auf dem Schirm schließen, allerdings ist die Position nur eindeutig in einer einzelnen

Periode. Eindeutigkeit könnte erreicht werden, indem die Wellenlänge gleich der Bildschirmbreite gewählt wird. Allerdings muss die Quantisierung des Aufnahmegeräts im Vergleich zur Bildschirmbreite relativ hoch sein. Dies ist in der Regel nicht gegeben. In der Praxis werden deshalb mehrere Kosinus-Muster mit unterschiedlichen Frequenzen gewählt. Dadurch werden jedem Pixel mehrere Phasen zugeordnet. Über eine anschließende Phasenfaltung wird die Bildschirmposition eindeutig bestimmt; siehe Unterabschnitt 3.7.2.

### 3.7.2 Phasenfaltung

Die Phasenschiebecodierung mit nur einer Wellenlänge liefert kein eindeutiges Ergebnis. Daher werden häufig mehrere Muster mit unterschiedlichen Perioden gewählt. In der Literatur gibt es hierfür mehrere Ansätze. Beispielsweise schlägt Kammel [Kam04] ein mehrstufiges Phasenschiebeverfahren vor, bei dem die Periodenlänge in jeder Stufe halbiert wird. Petz [Pet06] verwendet ein zweistufiges Phasenschiebeverfahren mit zwei teilerfremden Perioden. Diese Verfahren berücksichtigen dabei nicht die Unsicherheit der Phase. In [Pak16] verwendet Pak eine Phasenfaltung mittels Gaußmischverteilungen für die Kalibrierung von Kameras. Sie berechnet zusätzlich zur Bildschirmposition auch eine Unsicherheit. In Unterabschnitt 3.8.1 wird die detaillierte Herleitung der Phasenfaltung mittels Gaußmischverteilungen vorgestellt und ihre Anwendung diskutiert.

## 3.8 Unsicherheit

Bisher wurden die Messmodellkomponenten für den rauschfreien Fall betrachtet. Die Zufallsvariable  $\underline{w}_s$  in  $h_s$  führt dazu, dass  $\underline{l}$  eine Zufallsvariable ist sowie die daraus resultierenden Neigungen. In diesem Abschnitt soll die Unsicherheit durch das Messmodell propagiert werden, um zu einer Unsicherheit der Neigungsmessung zu gelangen.

Zunächst ist es allerdings erforderlich die Unsicherheitsquellen zu identifizieren. Hier sollen zwei dominierenden Quellen aufgezählt werden. Die Bildaufnahme ist immer mit Rauschen behaftet, welches bei der deflektometrischen Registrierung zu Phasenrauschen führt. Das Phasenrauschen hängt direkt von der Helligkeit und dem Kontrast der aufgenommenen Bilder ab und kann als Maß für die Qualität der Registrierung dienen; siehe Unterabschnitt 3.8.1. Einen weiteren Einfluss auf die Neigungsunsicherheit hat die Lageunsicherheit der Kamera und des Bildschirms. Diese wird zum einen durch die Positioniergenauigkeit des Handhabungswerkzeugs beeinflusst und zum anderen durch die Kalibrierungsunsicherheit der relativen Position von Sensorkomponenten zum Handhabungswerkzeug oder der Kamerakalibrierung. In dieser Arbeit wird angenommen, dass diese Unsicherheit in einer kontrollierten Umgebung niedrig gehalten werden kann, so dass sie nicht berücksichtigt wird. Eine zu hohe Positioniergenauigkeit kann dazu führen, dass die gemessenen Messbereiche nicht mit den geplanten Messbereichen übereinstimmen.

### 3.8.1 Prädiktion der Phasenunsicherheit

Wie weiter unten in diesem Unterabschnitt gezeigt wird, kann die Berechnung des Phasenrauschens auf die relative Helligkeit  $\beta$  und den Kontrast  $\gamma$  der deflektometrischen Registrierung und zusätzlichen standardisierten Eigenschaften der Kamera zurückgeführt werden. Unter anderem hängen der Kontrast und die relative Helligkeit von der Sensorkonfiguration, der Form und insbesondere der Rauigkeit der Oberfläche ab und sind für jeden Pixel unterschiedlich [Höf13]. Da das approximierte Modell  $h_s$  die deflektometrische Registrierung nur strahlengeometrisch modelliert, lassen sich diese Größen nicht aus dem Messmodell herleiten. Das Phasenrauschen muss vor der Planung experimentell bestimmt werden. In dieser Arbeit wird über eine Messreihe ein durchschnittlicher Wert bestimmt. Hierzu muss das

Phasenrauschen aus den beobachteten Mustersequenzen prädiert werden. Dies wird im restlichen Unterabschnitt diskutiert.

**Unsicherheit der Bildaufnahme**

Der EMVA 1288 Standard [EMV10] definiert eine einheitliche Methode zur Messung und Berechnung von Parametern zur Charakterisierung von Bildsensoren. Ihr liegt ein lineares mathematisches Modell zugrunde. Ein schematisches Model ist in Abbildung 3.6 gegeben.

Das Licht eines Oberflächenpatches aus der Szene wird durch ein Linsensystem gesammelt und auf den digitalen Bildsensor geleitet. Eine Anzahl von  $n_p$  Photonen treffen auf den lichtempfindlichen Teil der Pixel und erzeugen  $n_e$  freie Elektronen. Die Quanteneffizienz  $\eta(\lambda_p)$  beschreibt den Zusammenhang zwischen erzeugten freien Elektronen und auftreffenden Photonen:

$$\eta(\lambda_p) = \frac{\mu_e}{\mu_p} . \tag{3.43}$$

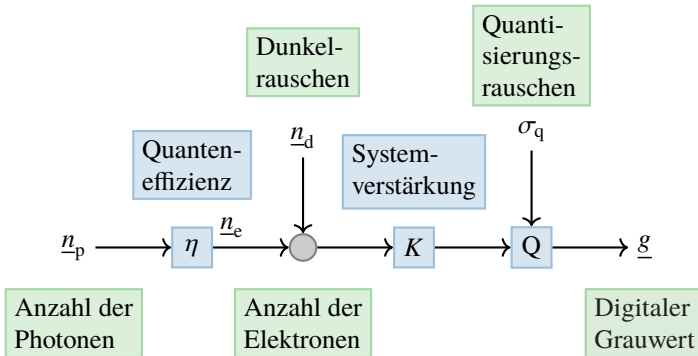


Abbildung 3.6: Mathematisches Modell der Bildentstehung durch den EMVA 1288 Standard [EMV10].

Hierbei ist  $\mu_p = E \{ \underline{n}_p \}$  die durchschnittliche Anzahl von Photonen, die während der Belichtungszeit auf die Pixelfläche treffen und  $\mu_e = E \{ \underline{n}_e \}$  die durchschnittliche Anzahl von freigesetzten Elektronen. Diese Elektronen werden in einem Potentialtopf gehalten. Anschließend werden die Ladungen in eine Spannung umgewandelt und verstärkt. Dies wird als ein linearer Prozess definiert und das erwartete Grauwertsignal ist

$$\mu_g = K(\mu_e + \mu_d), \quad (3.44)$$

wobei  $K$  der Verstärkungsfaktor ist. Die spontane Bildung von zusätzlichen  $\underline{n}_d$  Ladungsträgern, beispielsweise durch thermodynamische Prozesse, sorgt für ein zusätzliches Rauschsignal, das sogenannte *Dunkelrauschen*, dessen durchschnittliche Anzahl mit  $\mu_d = E \{ \underline{n}_d \}$  bezeichnet wird. Im letzten Schritt wird der Wert quantisiert. Aus den Gleichungen (3.43) und (3.44) ergibt sich für das durchschnittliche Signal

$$\mu_g = E \{ \underline{g} \} = K(\eta\mu_p + \mu_d). \quad (3.45)$$

Es werden drei Unsicherheitsquellen modelliert. Die Anzahl der freigesetzten Elektronen  $\underline{n}_e$  schwankt stochastisch gemäß der Poisson Verteilung, da eine zufällige Anzahl an Photonen auf die Pixelfläche treffen (*Schrotrauschen*). Die Varianz der Poisson Verteilung entspricht seinem Erwartungswert

$$\sigma_e^2 = \text{Var} \{ \underline{n}_e \} = \mu_e. \quad (3.46)$$

Der Analog-Digital-Konverter fügt eine zusätzliche Unsicherheit hinzu, die in den Quantisierungsintervallen gleichverteilt ist und eine Varianz von  $\sigma_q^2 = \frac{1}{12}$  hat. Da die drei Zufallsgrößen  $\underline{n}_p$ ,  $\underline{n}_e$  und  $\underline{n}_d$  als stochastisch

unabhängig angenommen werden können, ergibt sich für die Varianz des Grauwertes nach der Einführung der Variable  $\sigma_d^2 = \text{Var} \{ \underline{n}_d \}$  die Gleichung

$$\sigma_g^2 = \text{Var} \{ g \} = K^2(\sigma_e^2 + \sigma_d^2) + \sigma_q^2. \quad (3.47)$$

### Phasenunsicherheit

In diesem Unterabschnitt wird die Fortpflanzung der Unsicherheit durch das Phasenschiebeverfahren beschrieben. Rathjen [Rat95] untersucht die statistischen Eigenschaften der Phasenschiebealgorithmen. Seinen Überlegungen liegt die Annahme zugrunde, dass die Intensitätswerte mit einer Normalverteilung approximiert werden können. Dies ist möglich, wenn das Dunkelrauschen und Schrotrauschen die Hauptrauschquellen sind. Im Allgemeinen unterscheidet sich die Verteilung der Phase von einer Normalverteilung, da sie nichtlinear von den Intensitäten abhängt. Rathjen zeigt, dass für Algorithmen mit gewissen statistischen Eigenschaften<sup>3</sup>, worunter auch symmetrische  $M$ -Schritt Algorithmen fallen, der Phasenfehler hinreichend gut mit einer Normalverteilung approximiert werden kann, wenn für das Signal-Rausch-Verhältnis  $\text{SNR} \gg 1$  gilt. Das Signal-Rausch-Verhältnis ist definiert durch  $\text{SNR} = \frac{\beta g_{\text{sat}}}{\sigma_g}$  als das Verhältnis des durchschnittlichen Grauwertes  $\beta g_{\text{sat}}$  zur Standardabweichung  $\sigma_g$  des Grauwertes. Nach Surrel [Sur97] ist die Standardabweichung der Phase durch

$$\sigma_\phi = \sqrt{\frac{2}{M}} \frac{1}{\gamma} \frac{\sigma_g}{\beta g_{\text{sat}}} = \sqrt{\frac{2}{M}} \frac{1}{\gamma \cdot \text{SNR}} \quad (3.48)$$

---

<sup>3</sup> Für eine generelle Dekodierungsformel der Form  $\tan(\phi) = \frac{Z(g_1, \dots, g_M)}{N(g_1, \dots, g_M)}$  muss die Korrelation des Zählers  $Z$  und des Nenners  $N$  null sein und das Verhältnis ihrer Varianzen gleich eins.



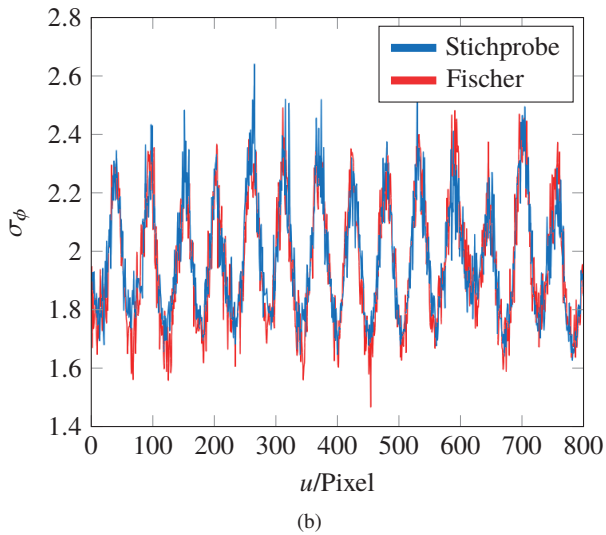
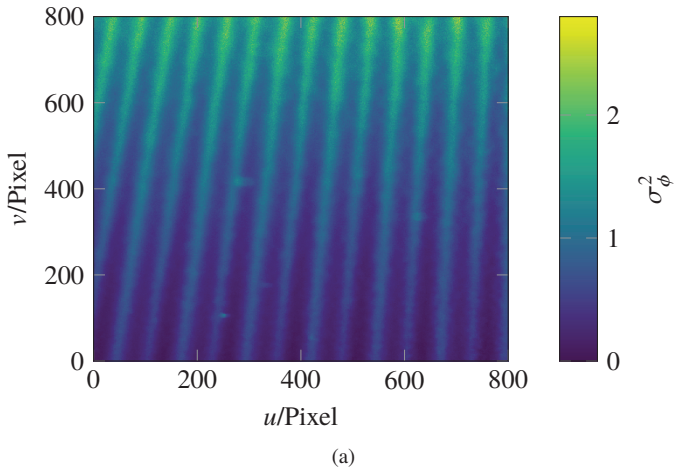


Abbildung 3.7: (a) Schätzung des Phasenrauschens. (b) Schnittlinie bei  $y = 450$  und Vergleich der Varianzen geschätzt durch die Methode von Fischer und einer Schätzung mittels einer Stichprobe.

gegeben. Basierend auf Gleichung (3.48) erweitert Fischer [Fis12] dieses Ergebnis, indem er das Kameramodell des EMVA Standards 1288 benutzt, um ein Phasenrauschen abzuleiten, dass nur von den Größen des Phasenschiebeverfahrens ( $\beta, \gamma, M$ ) und der Kameraparameter aus dem Standard ( $\mu_{e,\text{sat}}, \sigma_d, K$ ) abhängt, wobei  $\mu_{e,\text{sat}}$  die Sättigungskapazität ist. Nach dem EMVA Standard 1288 hängt die Varianz des Grauwertes von dem Grauwert ab; siehe Gleichung (3.47). Wird diese Abhängigkeit und (3.45) in (3.48) eingesetzt, erhält man für das erweiterte Signal-Rausch-Verhältnis

$$\text{SNR} = \frac{\mu_g - \mu_{g,\text{dark}}}{\sigma_g} = \frac{K\eta\mu_p}{\sqrt{K^2(\sigma_e^2 + \sigma_d^2) + \sigma_q^2}}. \quad (3.49)$$

Diese wird nun in (3.48) eingesetzt und daraus ergibt sich

$$\sigma_\phi = \sqrt{\frac{2}{M} \frac{1}{\gamma\beta\mu_{e,\text{sat}}} \sqrt{\beta\mu_{e,\text{sat}} + \sigma_d^2 + \frac{1}{12K^2}}}. \quad (3.50)$$

Der Kontrast  $\gamma$  und die Helligkeit  $\beta$  des gemessenen Pixels können berechnet werden, indem die Gleichungen (3.41) nach  $\beta$  und  $\gamma$  aufgelöst werden:

$$\beta = \frac{\frac{1}{M} \sum_{k=1}^M \tilde{g}_k}{K\mu_{e,\text{sat}}}, \quad (3.51)$$

$$\gamma = 2 \frac{\sqrt{\left(\sum_{k=1}^M \tilde{g}_k \sin \psi_k\right)^2 + \left(\sum_{k=1}^M \tilde{g}_k \cos \psi_k\right)^2}}{\sum_{k=1}^M \tilde{g}_k}. \quad (3.52)$$

Dabei ist  $\tilde{g}_k = g_k - \mu_{g,\text{dark}}$  der Grauwert abzüglich dem erwarteten Dunkelsignal. In Abbildung 3.7 wird die Prädiktion des Phasenrauschens aus einer Messung dargestellt und mit einer Schätzung aus einer Stichprobe von 100 Messungen verglichen. Es wurde ein Muster mit einer Wellenlänge wurde von 50 Pixel genutzt. Es ist zu erkennen, dass die Prädiktion gut mit der Schätzung übereinstimmt.

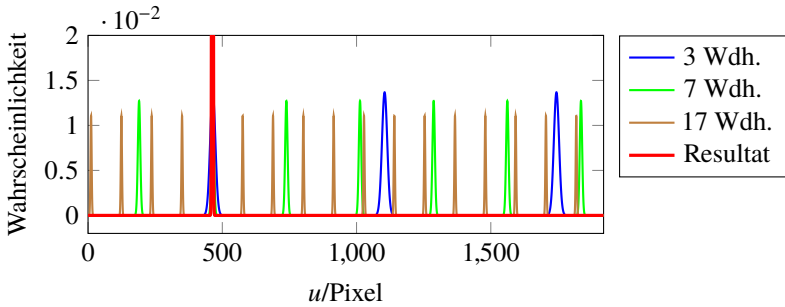


Abbildung 3.8: Ergebnis einer Schätzung mittels Gaußmischverteilung. Die Wellenlängen der Muster für 3, 7 und 17 Wiederholungen betragen jeweils 640, circa 274 und circa 112 Pixel.

### Phasenfaltung

Die probabilistische Phasenfaltung wurde bereits von Pak [Pak16] für die Kalibrierung von Kameras vorgestellt. In diesem Abschnitt wird eine detaillierte Herleitung über das Bayes-Theorem gegeben. Für die Phasenfaltung mit einer Wellenlänge  $\lambda = \frac{L_u}{M_p}$  ergibt sich eine Messunsicherheit der Phase wie in Gleichung (3.50). Sei  $u$  die zu schätzende Position und  $u' = u \bmod \lambda$  die gemessene Position. Aufgrund der Mehrdeutigkeit beim Phasenschiebepverfahren ergibt sich insgesamt für die Likelihood eine Gaußmischverteilung mit Gaußglocken, die an den mit der Messung vereinbaren Positionen  $u + k\lambda$  zentriert sind:

$$p(u' | u, \lambda) \propto \sum_k \mathcal{N}\left(u' | u + k\lambda, \sigma^2\right), \quad k \in \mathbb{N}, -\frac{u}{\lambda} \leq k \leq \frac{L_u - u}{\lambda}.$$

Die Anzahl der Gaußglocken ist durch die Länge  $L_u$  der zugehörigen Bildschirmachse beschränkt. Gesucht ist die A-posteriori-Verteilung

$$p(u | u'_{1:n}, \lambda_{1:n}) \quad (3.53)$$

nach  $n$  Messungen mit den Wellenlängen  $\lambda_{1:n}$ . Sie kann iterativ aus

$$p(u | u'_{1:n}, \lambda_{1:n}) \propto p(u'_n | u, \lambda_n) p(u | u'_{1:n-1}, \lambda_{1:n-1}) \quad (3.54)$$

$$\propto \sum_{k=-\infty}^{+\infty} \mathcal{N}(u | u'_n - k\lambda_n, \sigma^2) p(u | u'_{1:n-1}, \lambda_{1:n-1}) \quad (3.55)$$

berechnet werden. Für die obige Formulierung der A-posteriori-Verteilung wird für die Likelihood angenommen, dass  $k$  von  $-\infty$  bis  $\infty$  läuft. Da eine A-priori-Verteilung gewählt wird, die außerhalb des Bildschirms eine Wahrscheinlichkeit von Null besitzt, ist dies gleichwertig mit einer Formulierung, in der der Wertebereich für  $k$  eingeschränkt ist. Für die A-priori-Verteilung wird eine Gleichverteilung angenommen

$$p(u) = \begin{cases} \frac{1}{L_u} & u \in [0, L_u] \\ 0 & \text{sonst} \end{cases} \quad (3.56)$$

In jedem Inferenzschritt (3.54) werden zwei Gaußmischverteilungen miteinander multipliziert was zu einer Steigerung der Anzahl der Komponenten führt. Die Gewichte des Produkts zweier Gaußmischverteilungen hängen von den Gewichten, dem Erwartungswert und der Standardabweichung seiner beiden Faktoren ab. Komponenten mit kleinen Gewichten können entfernt werden, um das Produkt mit einer verringerten Anzahl an Komponenten zu approximieren. Durch geschickte Wahl der Frequenzen kann die Anzahl der Komponenten mit kleinem Gewicht vergrößert werden. Dies kann erreicht werden, indem teilerfremde Musterfrequenzen gewählt werden, sodass ihr Produkt größer ist als der Bildschirmbereich  $L_u$ . Dann überschneiden sich im Idealfall nur zwei Komponenten. Zusätzlich ist es eine gute Strategie, zuerst große Wellenlängen zu wählen, um die Gesamtanzahl der Komponenten gering zu halten. In Abbildung 3.8 ist eine Schätzung der Bildschirmposition

mittels Gaußmischverteilung dargestellt. Es wurden drei unterschiedliche Wellenlängen gewählt.

### 3.8.2 Prädiktion der Neigungsunsicherheit

In diesem Unterabschnitt wird die Unsicherheit gemessener Neigungen berechnet. Die Unsicherheit der Phasenfaltung, die im vorherigen Unterabschnitt diskutiert wurde, führt zu einer Unsicherheit der gemessenen Zuordnung  $\mathbf{l}(u, v)$ . Diese Unsicherheit wird in Form einer Gaußmischverteilung modelliert. Unter der Annahme, dass die Verteilung um einen Schätzwert konzentriert ist, was durch die Wahl geeigneter Frequenzkombinationen erreicht werden kann, kann sie weiter als eine Normalverteilung approximiert werden, in dem der Erwartungswert und die Standardabweichung berechnet werden. Sei  $\underline{\mathbf{w}}_s \in \mathbb{R}^2$  mit  $\underline{\mathbf{w}}_s(u, v) \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_s)$  die Abweichung zur gemessenen Zuordnung, dann ist

$$h_s(\mathcal{S}, \boldsymbol{\theta}, \underline{\mathbf{w}}_s, u, v) = h_s(\mathcal{S}, \boldsymbol{\theta}, \mathbf{0}, u, v) + \underline{\mathbf{w}}_s(u, v). \quad (3.57)$$

Da der Vektor  $\mathbf{l}(u, v) = h_s(\mathcal{S}, \boldsymbol{\theta}, \mathbf{0}, u, v)$  nichtlinear in die Berechnung des Normalenvektors eingeht, ist die resultierende Wahrscheinlichkeitsverteilung keine Normalverteilung. Die Unsicherheit wird mittels einer Linearisierung um den rauschfrei berechneten Vektor  $\mathbf{l}(u, v)$  abgeschätzt:

$$h_n^{(2)}(\mathcal{S}_r, \mathbf{l} + \underline{\mathbf{w}}_s, \boldsymbol{\theta}, u, v) \approx h_n^{(2)}(\mathcal{S}_r, \mathbf{l}, \boldsymbol{\theta}, u, v) + \frac{\partial h_n^{(2)}}{\partial \mathbf{l}} \underline{\mathbf{w}}_s(u, v). \quad (3.58)$$

Die partielle Ableitung bezüglich  $\mathbf{l}$  ist durch

$$\mathbf{A} = \frac{\partial h_n^{(2)}}{\partial \mathbf{l}} = \frac{\mathbf{l} - \hat{\mathbf{n}}\hat{\mathbf{n}}^T}{\|\hat{\mathbf{s}} - \hat{\mathbf{s}}_r\|} \frac{\mathbf{l} - \hat{\mathbf{s}}_r\hat{\mathbf{s}}_r^T}{\|\mathbf{s}_r\|} \quad (3.59)$$

gegeben. Insgesamt ergibt sich für die Neigungsunsicherheit

$$\mathbf{W}_n = \mathbf{A}\mathbf{W}_s\mathbf{A}^T. \quad (3.60)$$

Diese Kovarianzmatrix wird während der Simulation genutzt, um zusätzliche eine Unsicherheit für jeden Messpunkt zu berechnen.

### 3.9 Simulator

Die Implementierung des Sensormodells erfolgt über einen Simulator. Der Simulator arbeitet ähnlich zu einem Raytracer wie er in der Computergrafik angewandt wird. Für jedes Pixel auf der Kamera wird ein Lichtstrahl ausgesandt. Im Unterschied zum Raytracing werden nicht die Intensitäten, sondern direkt die Schnittpunkte und Pixelkoordinaten auf dem Bildschirm bestimmt, falls diese existieren. Für die Bestimmung der gemessenen Normalenvektoren ist es ausreichend, diese direkt an den Schnittpunkt aus der Referenzoberfläche zu extrahieren. Die alternative Berechnung der Normalenvektoren anhand  $h_s$  und  $h_n$  führt zum selben Ergebnis, da die wahre Oberfläche nicht bekannt ist und in der Simulation sowohl für die Berechnung der deflektometrischen Registrierung  $l$  als auch die Berechnung der Normalenvektoren dieselbe Referenzoberfläche gewählt wird. Die Messgleichungen  $h_s$ ,  $h_n$  werden allerdings für die Berechnung der Unsicherheit benötigt. Weitere Größen, die in diesem Zusammenhang berechnet werden, sind der Auftreffwinkel des Strahls auf dem Schirm. Da die Abstrahlcharakteristik des Schirms nicht modelliert wird, kann über diese Größe ein Grenzwert gesetzt werden, der sicherstellt, dass Strahlen, die zu flach auf den Schirm treffen, aussortiert werden. Über die Gleichung (3.14) wird die Unsicherheitscheibe auf der Oberfläche berechnet. Diese kann bei der Schätzung der Oberfläche mitberücksichtigt werden.

Weitere interessante Merkmale sind die Typen der Lichtpfade. Um diese zu beschreiben, wird eine Grammatik genutzt. Das Alphabet der Grammatik besteht aus den Stationen, die ein Strahl durchlaufen kann. Die Stationen sind der Start auf der Sensorebene der Kamera, mit  $C$  abgekürzt, der Kamerakörper  $C_B$ , die Oberfläche  $S$  und der Schirm, wobei unterschieden werden kann zwischen dem bildgebenden Schirmbereich  $D$ , außerhalb des Schirmbereiches auf der Schirmebene  $D_P$  und dem Schirmrücken  $D_B$ . Die möglichen Lichtpfade sind in der Bakus-Naur-Form gegeben durch

$$\begin{aligned}\langle L \rangle &::= D \mid D_P \mid D_B \\ \langle P \rangle &::= C [ \langle L \rangle \mid S^* [C_B \mid \langle L \rangle] ] .\end{aligned}$$

Die Strahlen, die zu einer gültigen Messung führen, haben die Form  $CSD$ . Ungültige Messungen entstehen dadurch, dass keine Oberfläche  $C$  oder kein Schirm  $CS$  beobachtet wird. Strahlen können auch durch den Schirm  $CD$  oder durch die Kamera  $CSC_B$  unterbrochen werden. Von besonderer Bedeutung für die Optimierung sind die Pfade  $C$ , d.h. die Oberfläche wird nicht beobachtet oder  $CSD_B$  die Rückseite des Schirms wird beobachtet. Verfälschte Messungen entstehen durch Mehrfachreflexionen ( $CS^*D$ ). Hier reicht es nicht, diese als ungültige Messungen zu markieren, da sie in der realen Messung nicht immer als solche identifiziert werden können. Sensorkonfigurationen mit Mehrfachreflexionen können in der Gütefunktion speziell bestraft werden, indem zum Beispiel ein hoher Bestrafungsterm addiert wird. In den Abbildungen 3.9 und 3.10 sind beispielhaft jeweils für eine konvexe Oberfläche und einer ausgewählten Konfiguration der Lichtpfad, die Standardabweichung der Normalenvektoren, der Sichtwinkel der Kamera auf die Oberfläche und die Größe der Unschärfescheibe auf der Oberfläche dargestellt.

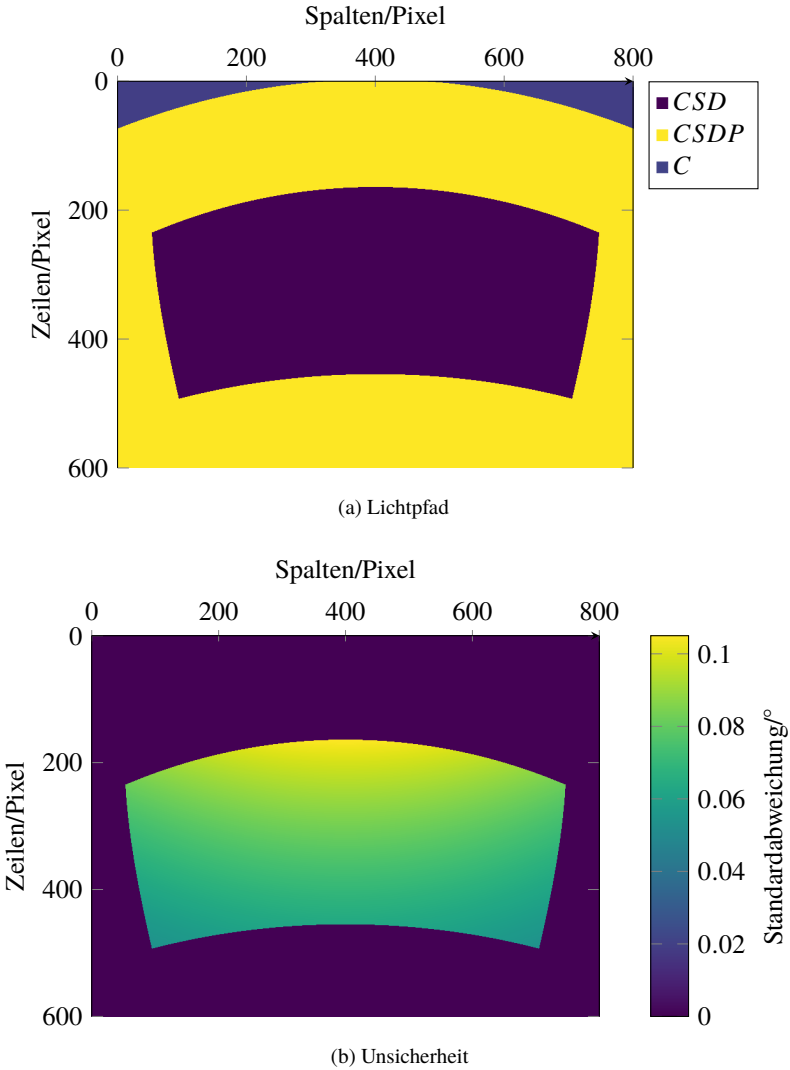


Abbildung 3.9: Dargestellt sind beispielhaft (a) der simulierte Lichtpfad und (b) die Standardabweichung der gemessenen Normalenvektoren aus der Simulation für eine konvexe Oberfläche.



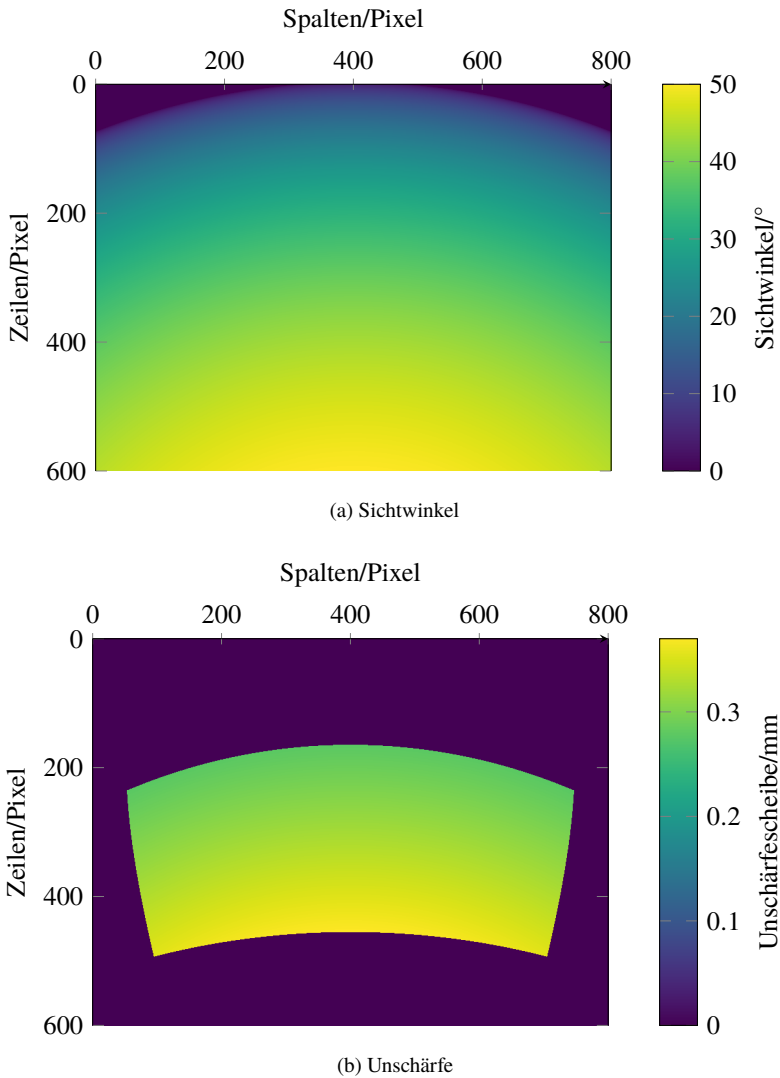


Abbildung 3.10: Dargestellt sind beispielhaft (a) der Sichtwinkel der Kamera auf die Oberfläche und (b) die Größe der Unschärfescheibe auf der Oberfläche aus der Simulation für eine konvexe Oberfläche.

## 3.10 Zusammenfassung

In diesem Kapitel wurde das Messmodell vorgestellt. Zunächst wurde auf die Modellierung der einzelnen Komponenten einer deflektometrischen Messung (die Kamera, die Oberfläche und den Bildschirm) eingegangen. Die Sensorkonfiguration beschreibt die einstellbaren Größen des Sensors. In dieser Arbeit wurden die geometrischen Größen wie die Lage des Schirms und der Kamera verwendet. Die deflektometrische Registrierung und die Berechnung der Normalenvektoren werden mittels geometrischer Betrachtungen modelliert. Hierfür muss angenommen werden, dass eine Referenzoberfläche gegeben ist.

Die Unsicherheitsquellen sind zum einen das Phasenrauschens, das aufgrund der Bildaufnahme entsteht und sich auf die deflektometrische Registrierung auswirkt und zum anderen die Positionierengenauigkeit des Handhabungsgerätes sowie die Kalibrierung, die sich auf die Neigungsunsicherheit auswirken. Letztere wurden in dieser Arbeit nicht berücksichtigt. Da mit einem geometrischen Modell die Unsicherheit der deflektometrischen Registrierung nicht hergeleitet werden kann, muss diese über ein experimentelles Verfahren geschätzt werden. Dies kann mit der Unsicherheitsprädiktion von Fischer und der Phasenfaltung über Gaußmischverteilung geschehen. Die Unsicherheit der Normalenvektoren wird über eine Linearisierung der Normalenmessgleichung bestimmt. Die Implementierung des Messmodells geschieht über einen Simulator, der zusätzlich zu den Messstellen, Normalenvektoren und Neigungsunsicherheiten noch weitere Größen wie beispielsweise die Lichtpfade, die Auftreffwinkel auf den Schirm oder Unschärfescheibe auf der Oberfläche berechnet, die während einer Planung verwendet werden können.

## 4 Oberflächenmodelle und Fusion

Während der Planung müssen die einzelnen simulierten Messungen fusioniert werden, um Aussagen über den bisherigen Plan hinsichtlich der Unsicherheit zu treffen. Dies kann dadurch erfolgen, dass die Messungen genutzt werden, um ein globales Modell der Oberfläche zu aktualisieren. Folgende Anforderungen werden an das Oberflächenmodell gestellt

- Die Oberfläche sollte probabilistisch beschrieben werden, so dass Unsicherheiten extrahiert werden können und die Unsicherheit der Messungen berücksichtigt werden können.
- Die Berechnung der Unsicherheit sollte effizient erfolgen können.
- Eine effiziente rekursive Aktualisierung des Modells sollte möglich sein. Damit lassen sich die Methoden in Kapitel 5 beschleunigen, indem während der Planung nicht in jedem Optimierungsschritt alle Messungen in das Modell fusioniert werden müssen.

In diesem Kapitel werden zwei probabilistische Oberflächenmodelle für die Anwendung in der Planung vorgestellt. Die Fusion der Messungen in diese Modelle und die Wahl des Unsicherheitsmaßes müssen gleichzeitig betrachtet werden, da das Modell die Fusionsmethodik und die Unsicherheitsbeschreibung weitgehend bestimmt.

## 4.1 Überblick über Oberflächenmodelle

Das Oberflächenmodell enthält alle Merkmale der Oberfläche, die für die Messaufgabe relevant sind. In der Deflektometrie sind vorwiegend die geometrischen Merkmale der Oberfläche, wie beispielsweise die Höhen oder Neigungen von Interesse.

Die *implizite* Darstellung einer Oberfläche  $\mathcal{S}$  beschreibt die Oberfläche durch den Kern einer Funktion  $\phi_I : \mathbb{R}^3 \rightarrow \mathbb{R}$ . Das heißt die Oberfläche ist durch

$$\mathcal{S} = \{\mathbf{x} \mid \phi_I(\mathbf{x}) = 0\} \quad (4.1)$$

gegeben. Bei einer impliziten diskreten Darstellung wird der Raum in Voxel eingeteilt, die in einer Gitterstruktur angeordnet sind. In einem Voxel wird vermerkt, ob er zur Oberfläche gehört oder nicht. Weitere geometrische Merkmale der Oberfläche wie zum Beispiel die Normale oder die Krümmungen können in einem Voxel abgespeichert werden.

Eine *explizite* Darstellung einer Oberfläche wählt eine Parametrisierung  $\phi_E : \Omega \rightarrow \mathcal{S}$  der Oberfläche, die einen zweidimensionalen Parameterraum  $\Omega \subset \mathbb{R}^2$  auf die Oberflächenmerkmale  $\mathcal{S} = \phi_E(\Omega)$  abbildet. Beispiele für häufig verwendete parametrische Modelle sind B-Splines oder NURBS. Parametrische Oberflächenrepräsentationen haben den Vorteil, dass sich dreidimensionale Probleme auf der Oberfläche in zweidimensionale Probleme im Parameterraum  $\Omega$  reduzieren lassen. So können Punkte auf der Oberfläche durch eine einfache Funktionsevaluation von  $\phi_E$  berechnet werden. Andererseits lassen sich nicht alle Oberflächen mit beliebigen Topologien stetig parametrisieren. Stetigkeit ist eine wichtige Eigenschaft, weil sie sicherstellt, dass Nachbarschaftsbeziehungen auf der Oberfläche auch im Parameterraum erhalten bleiben. Eine Möglichkeit, um parametrische Oberflächen auf beliebige Topologien zu erweitern, besteht darin, die Oberfläche in Teilstücke zu zerlegen und für jedes Teilstück eine separate Parametrisierung zu wählen.

Eine Schwierigkeit bei der so stückweise definierten Oberfläche ist, je nach Anforderung, den stetigen oder glatten Übergang von einem Teilstück zum anderen zu garantieren.

Eine spezielle Parametrisierung ist, die Oberfläche durch einen Funktionsgraphen  $f : \Omega \rightarrow \mathbb{R}$  darzustellen, wobei der Parameterraum die horizontale Ebene ist, über der die Oberfläche als Funktion aufgetragen ist:

$$S = \left\{ \left( \begin{array}{c} u \\ v \\ f(u, v) \end{array} \right) \mid (u, v) \in \Omega \right\} \quad (4.2)$$

Diese Repräsentation hat die Einschränkung, dass sie keine Überhänge oder Faltungen darstellen kann.

Die Darstellung über *Polygonnetze* beschreibt die Oberfläche durch eine Menge von Punkten, die über Kanten miteinander verbunden sind. Die durch die Ecken begrenzten Flächen werden *Facetten* genannt. Besonders häufig genutzt werden Dreiecksnetze, bei der die Facetten Dreiecke bilden.

Zur Definition einer Wahrscheinlichkeitsverteilung über den Oberflächen kann zwischen örtlich diskreten und örtlich kontinuierlichen Modellen unterschieden werden. Wahrscheinlichkeitsverteilungen für kontinuierliche Oberflächen können als eine lineare gewichtete Summe von Basisfunktionen repräsentiert werden. Eine Wahrscheinlichkeitsverteilung auf den Gewichten liefert eine Verteilung auf Oberflächen. In einer Verallgemeinerung kann die Definition der Verteilung direkt im Funktionenraum erfolgen. Das hat den Vorteil, dass die Repräsentationsstärke dieser Verteilung höher ist. Ein Beispiel für solche Verteilungen sind Gauß-Prozesse. Sie werden in Abschnitt 4.3 als Oberflächenmodell genutzt.

Bei diskreten Modellen kann ein Markow-Zufallsfeld [Rue05] verwendet werden. Hierzu werden die Merkmale an den diskreten Stellen als Zufallsvariablen modelliert. Jede Stelle hat eine Nachbarschaft und es wird angenommen, dass die zugehörigen Merkmale bezüglich ihrer Nachbarschaft bedingt unabhängig sind. Diese Annahme kann häufig in der Praxis gerechtfertigt werden und führt zu effizienten Verfahren zur Berechnung des Erwartungswertes. Die Berechnung der Varianz ist im Allgemeinen aufwendig und approximative Verfahren sind erforderlich. In Abschnitt 4.4 wird ein Modell vorgestellt, das die Oberfläche als eine Verteilung der lokalen Oberflächenform an diskreten Stellen repräsentiert. Hier wird allerdings angenommen, dass die Merkmale an den diskreten Stellen unabhängig voneinander sind.

## 4.2 Regression

Regressionsverfahren haben das Ziel einen funktionalen Zusammenhang zwischen einer unabhängigen Größe  $\mathbf{x}$  und einer abhängigen Zufallsvariable  $y$  zu finden. In der klassischen Formulierung wird angenommen, dass eine Menge von Daten  $\mathcal{Z} = \{(\mathbf{x}_i, y_i) \mid i \in 1, 2, \dots, |\mathcal{Z}|\}$  bestehend aus der unabhängigen Größen  $\mathbf{x}_i$  mit zugehörigen Realisierungen  $y_i$  der Zufallsvariable gegeben ist und die Realisierungen folgendem Modell

$$y_i = f(\mathbf{x}_i) + w_i \tag{4.3}$$

genügen. Dabei ist  $f$  die gesuchte Regressionsfunktion und der Wert  $w_i$  ist eine Realisierung der Zufallsvariablen  $\underline{w}_i$ . Die Zufallsvariable  $\underline{w}_i$  sind erwartungswertfrei  $E\{\underline{w}_i\} = 0$  und unkorreliert  $\text{Cov}\{\underline{w}_i, \underline{w}_j\} = \delta_{ij}\sigma_i^2$  mit den Varianzen  $\sigma_i^2$ .

Im Rahmen dieser Arbeit ist es nötig eine verallgemeinerte Form der Regression zu betrachten. Es wird angenommen, dass die abhängigen Größen aus der Anwendung von linearen Operatoren  $T_i : \mathcal{H} \rightarrow \mathbb{R}^d$  auf die

zu suchende Funktion entstammen, wobei  $\mathcal{H}$  ein geeigneter Funktionenraum ist. Zusätzlich wird angenommen, dass die Unsicherheit bezüglich der abhängigen Variable bekannt ist. Die Daten sind in diesem Fall durch  $\mathcal{Z} = \{(\mathbf{T}_i, \mathbf{q}_i, \mathbf{W}_i) \mid i \in 1, 2, \dots, |\mathcal{Z}|\}$  bestehend aus den Operatoren  $\mathbf{T}_i$ , der abhängigen Variable  $\mathbf{q}_i$  und der Unsicherheit  $\mathbf{W}_i$  bezüglich der abhängigen Variable gegeben. Sie erfüllen den Zusammenhang

$$\mathbf{q}_i = \mathbf{T}_i f + \mathbf{w}_i \quad (4.4)$$

mit  $E\{\underline{\mathbf{w}}_i\} = \mathbf{0}$  und  $\text{Cov}\{\underline{\mathbf{w}}_i, \underline{\mathbf{w}}_j\} = \mathbf{0} + \delta_{ij} \mathbf{W}_i$ .

Das Regressionsverfahren kann zur Schätzung der Oberfläche aus Messdaten verwendet werden. Die zu schätzende Oberfläche wird als eine Funktion  $f$  und die einzelnen Messungen durch den Operator  $\mathbf{T}$  modelliert. Wird  $\mathbf{T}_i$  gleich dem Evaluationsfunktional  $T_i f = \delta_{\mathbf{x}_i} f = f(\mathbf{x}_i)$  mit der Messstelle  $\mathbf{x}_i$  gesetzt, so wird die Höhe der Oberfläche an der Stelle gemessen. Wird nur eine unscharfe Messung in der Umgebung von  $\mathbf{x}_i$  ausgeführt kann der Operator  $T_i f = \int k(\mathbf{x}' - \mathbf{x}_i) f(\mathbf{x}_i) d\mathbf{x}'$  verwendet werden, wobei  $k$  ein Glättungskern ist. Von speziellem Interesse in der Deflektometrie ist  $\mathbf{T}_i = \nabla_{\mathbf{x}_i}$  zur Messung der Gradienten.

### 4.3 Gauß-Prozess Modell

Das Gauß-Prozess Modell [Ros12, Ros13] modelliert die Oberfläche als einen Gauß-Prozess über die Höhe. Oberflächendefekte können als Höhenabweichung zur Referenzoberfläche erkannt werden.

Über eine sogenannte *Kovarianzfunktion* wird eine A-priori-Verteilung über die Höhenabweichungen definiert. Die Kovarianzfunktion beschreibt die lineare Abhängigkeit zwischen zwei Oberflächenpunkten und verhält sich wie ein Regularisierer: Es werden implizit Glattheitsannahmen über die Oberflächen getroffen.

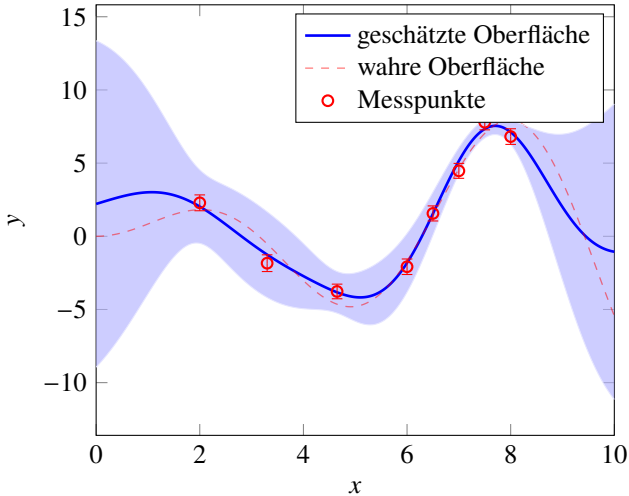


Abbildung 4.1: Schematische Darstellung der Unsicherheitsreduktion eines Gauß-Prozesses durch Beobachtungen. Die Messpunkte sind dargestellt mit einem  $1\sigma$ -Streuungsbalken. Die Unsicherheit der Schätzung ist durch den hellblauen Bereich abgebildet, der die  $1\sigma$ -Umgebung der Marginaldichten von dem Funktionswert  $f(x)$  an jeder Stelle darstellt.

Anhand von Messungen an mehreren Messstellen wird eine aktualisierte Verteilung über Oberflächen inferiert. Aufgrund der definierten Korrelation zwischen Oberflächenpunkten sinkt nicht nur die Unsicherheit an den Messstellen, sondern auch in einer Umgebung um diese; siehe Abbildung 4.1.

### 4.3.1 Gauß-Prozess

Ein stochastischer Prozess ist eine Familie von Zufallsvariablen  $f_{\underline{x}}$ ,  $x \in \mathcal{T}$ , wobei  $\mathcal{T}$  eine beliebigen Indexmenge ist. Ein Gauß-Prozess ist ein stochastischer Prozess, bei dem alle endlich-dimensionalen Randverteilungen multivariate Normalverteilungen sind.



Sei  $x_1, \dots, x_n \in \mathcal{T}$  mit beliebigem  $n \in \mathbb{N}$ , dann ist

$$\underline{f}_{x_1}, \dots, \underline{f}_{x_n} \sim \mathcal{N}(\underline{\mu}, \underline{\Sigma}) \quad (4.5)$$

normalverteilt mit zugehörigem Erwartungswert  $\underline{\mu} \in \mathbb{R}^n$  und Kovarianzmatrix  $\underline{\Sigma} \in \mathbb{R}^{n \times n}$ . Da in dieser Arbeit der Gauß-Prozess genutzt wird, um die Höhe einer Oberfläche zu repräsentieren, wird im Folgenden stets als Indexmenge  $\mathcal{T} = \mathbb{R}^2$  gewählt. Eine alternative Sichtweise interpretiert den Gauß-Prozess als eine Verallgemeinerung der Normalverteilung auf Funktionen. Demnach ist der Gauß-Prozess eine Zufallsfunktion  $\underline{f}$  und analog zu einer multivariaten Normalverteilung durch die ersten beiden Momente eindeutig bestimmt. Eine Funktion  $\mu : \mathbb{R}^2 \rightarrow \mathbb{R}$  übernimmt die Rolle des Erwartungswertes und eine Kovarianzfunktion  $k : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$  die Rolle der Kovarianzmatrix. Insgesamt schreibt man auch

$$\underline{f} \sim \mathcal{GP}(\mu, k) . \quad (4.6)$$

Der Zusammenhang zur obigen Definition lässt sich dadurch herstellen, dass für beliebige  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^2$  die Funktionswerte  $\underline{f}(\mathbf{x}_1), \dots, \underline{f}(\mathbf{x}_n)$  normalverteilte Zufallsvariablen sind mit

$$\mathbb{E} \left\{ \underline{f}(\mathbf{x}_i) \right\} = \mu(\mathbf{x}_i) \quad \text{und} \quad \text{Cov} \left\{ \underline{f}(\mathbf{x}_i), \underline{f}(\mathbf{x}_j) \right\} = k(\mathbf{x}_i, \mathbf{x}_j). \quad (4.7)$$

Gauß-Prozesse finden in zahlreichen Gebieten der Robotik oder des Maschinellen Lernens Anwendung. Dort werden sie zur Dimensionsreduktion, Regression oder Klassifikation eingesetzt. Obwohl bei einem Gauß-Prozess die Verteilung auf einen unendlich dimensional Funktionenraum definiert ist, lassen sich deren Lösungsverfahren mit Hilfe der Eigenschaft (4.7) dennoch stets in einer endlichen Dimension formulieren.

### 4.3.2 Prior

Gemäß der Bayes'schen Formulierung wird für die zu schätzende Größe eine A-priori-Verteilung definiert. Mit ihr kann Vorwissen in die Schätzung einbezogen werden. Der A-priori-Gauß-Prozess ist durch die Angabe eines A-priori-Erwartungswertes und einer A-priori-Kovarianzfunktion vollständig bestimmt.

#### Erwartungswert

Als A-priori-Erwartungswert wird die Referenzoberfläche gewählt. Mittels des Gauß-Prozesses wird dann nur noch die Abweichung zu ihr geschätzt. Da, wie im Unterabschnitt 4.3.5 zu sehen sein wird, die Unsicherheit nicht von den gemessenen Höhen oder Gradienten abhängt, beeinflusst der A-priori-Erwartungswert nicht die Kostenfunktion. Sie ist aber relevant für eine Schätzung des Erwartungswertes. Für den Fall, dass keine Referenzoberfläche gegeben ist und die Oberfläche wie in Unterabschnitt 4.3.7 geschätzt wird, ist der A-priori-Erwartungswert aber relevant. Dies liegt daran, dass die Oberfläche iterativ in mehreren Schritten geschätzt wird und in jedem Schritt anhand der Messung die Gradienten mittels des Erwartungswertes neu berechnet werden.

#### Kovarianzfunktion

Die Kovarianzfunktion beschreibt die Kovarianzen zwischen zwei Höhenwerten an unterschiedlichen Stellen auf der Oberfläche. Sie wirkt als ein Regularisierer: Zum einen schränkt sie die Menge der vom Schätzer realisierbaren Funktionen ein und zum anderen definiert sie eine Norm auf diesen Funktionenraum. Häufig zeichnen sich Funktionen mit kleineren Normen dadurch aus, dass sie glatter sind, wobei die spezielle Ausprägung der Glattheit je nach Kovarianzfunktion unterschiedlich ist.

Eine Funktion  $k : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$  ist eine Kovarianzfunktion, wenn sie

- symmetrisch  $k(\mathbf{x}_1, \mathbf{x}_2) = k(\mathbf{x}_2, \mathbf{x}_1)$  für  $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$  und
- positiv definit  $\sum_{i=1}^n \sum_{j=1}^n c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$  für beliebige  $n \in \mathbb{N}$ ,  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^2$  und  $c_1, \dots, c_n \in \mathbb{R}$

ist. Eine solche Funktion wird in der Literatur auch *Kernel* genannt und kommt zum Beispiel in Support Vector Machines (SVMs) [Bos92] oder in der Kernel-Hauptkomponentenanalyse (Kernel-PCA) [Sch99] zum Einsatz. Einen Überblick über weitere Kernel-Methoden bieten [Sch01] und [ST04].

Eine für diese Arbeit relevante Klasse von Kovarianzfunktionen sind

$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_A^2 \tilde{k} \left( \frac{\mathbf{x}_2 - \mathbf{x}_1}{l} \right) \quad (4.8)$$

wobei  $\sigma_A^2 \in \mathbb{R}_+$  und  $l \in \mathbb{R}_+$  Parameter der Kovarianzfunktion sind und  $\tilde{k} : \mathbb{R}^2 \rightarrow \mathbb{R}$  eine positiv definite Funktion mit der Eigenschaft  $\tilde{k}(0) = 1$ . Da  $k(\mathbf{x}, \mathbf{x}) = \sigma_A^2$  für alle  $\mathbf{x} \in \mathbb{R}^2$  kann der Parameter  $\sigma_A^2$  als Varianz der Höhenwerte identifiziert werden. Der Parameter  $l$  ist die Längenskala, mit der die Glattheit der Schätzung kontrolliert werden kann. Kovarianzfunktionen, die durch die Gleichung (4.8) beschrieben werden können, sind translationsinvariant.

In der Literatur wird häufig die sogenannte *squared exponential* Kovarianzfunktion benutzt. Sie ist definiert durch

$$\tilde{k}_{se}(\mathbf{x}) = e^{-\|\mathbf{x}\|} \quad (4.9)$$

Diese Kovarianzfunktion ist rotationssymmetrisch und besitzt einen unendlichen Support. In dieser Arbeit werden kubische B-Spline Funktionen als Kovarianzfunktion benutzt. Für die kubische B-Splines hat man

$$\tilde{k}_{cb}(\mathbf{x}) = s_3(x_1)s_3(x_2) \quad (4.10)$$

mit  $\mathbf{x} = (x_1, x_2)^T$  und den eindimensionalen kubischen Basisfunktionen

$$s_3(x) = \begin{cases} (2 - |x|)^3 - 4(1 - |x|)^3, & |x| < 1 \\ (2 - |x|)^3, & 1 \leq |x| \leq 2 \\ 0, & |x| > 2. \end{cases} \quad (4.11)$$

Die Kovarianzfunktion besitzt einen endlichen Support und ist separierbar. Diese beiden Eigenschaften bieten Vorteile bei der Inferenz hinsichtlich der Effizienz [Neg12]: Die Kovarianzmatrix ist dünnbesetzt und die Auswertung der Kovarianzfunktion an Punkten in einem Gitter lässt sich über das dyadische Produkt berechnen.

### 4.3.3 Auswirkung eines Operators auf den Gauß-Prozess

In diesem Unterabschnitt wird die Wahrscheinlichkeitsdichtefunktion der Messgrößen, die aus der Anwendung eines Operators auf einen Gauß-Prozess entsteht, bestimmt. Handelt es sich bei dem Operator um einen nichtlinearen Operator, dann ist die resultierende Messgröße im Allgemeinen nicht normalverteilt. In diesem Fall ist die exakte Inferenz schwierig und Approximationsverfahren müssen genutzt werden. In einem häufig auftretenden Spezialfall hiervon kann der Operator durch eine nichtlineare Funktion bezüglich einer Stelle  $\mathbf{x}$  und dem zugehörigen Funktionswert  $f(\mathbf{x})$  repräsentiert werden. Hierfür gibt es in der Literatur eine Vielzahl von approximativen Inferenzverfahren [Kus06].

Im Rahmen dieser Arbeit genügt es lineare Operatoren zu betrachten. Särkkä [Sär11] behandelt die Gauß-Prozess-Regression mit linearen Operatoren zur Berechnung von stochastischen partiellen Differentialgleichungen. Die

resultierende Größe der Anwendung des linearen Operators auf den Gauß-Prozess ist aufgrund der Linearität des Erwartungswertes und der Bilinearität der Kovarianz normalverteilt. Für den Erwartungswert erhält man

$$\mathbb{E} \left\{ \mathbf{T}_i \underline{f} \right\} = \mathbf{T}_i \mathbb{E} \left\{ \underline{f} \right\} = \mathbf{T}_i \boldsymbol{\mu}, \quad (4.12)$$

und für die Kovarianz

$$\text{Cov} \left\{ \mathbf{T}_i \underline{f}, \mathbf{T}_j \underline{f} \right\} = \mathbf{T}_i^{(1)} \mathbf{T}_j^{(2)} \text{Cov} \left\{ \underline{f}, \underline{f} \right\} \quad (4.13)$$

$$= \mathbf{T}_i^{(1)} \mathbf{T}_j^{(2)} k, \quad (4.14)$$

wobei es sich bei den Operatoren  $\mathbf{T}^{(1)}$  und  $\mathbf{T}^{(2)}$  um den Operator  $\mathbf{T}$  handelt, der jeweils auf das erste und zweite Argument der Kovarianzfunktion angewendet wird.

Abhängig vom Definitionsbereich der Operatoren muss bei der Wahl der Kovarianzfunktionen beachtet werden, dass neben der Symmetrie und der positiven Definitheit zusätzliche Bedingungen an sie gefordert werden. Beispielsweise muss für den Fall des Gradientenoperators  $\nabla_{\mathbf{x}}$  die Kovarianzfunktion zweimal differenzierbar sein.

#### 4.3.4 Notation

Bevor auf die Schätzung der Oberfläche aus den Messdaten eingegangen werden kann, muss kurz auf die Notation eingegangen werden. Seien

$$\mathbf{T} = (\mathbf{T}_1 \dots \mathbf{T}_n) \quad \text{und} \quad \tilde{\mathbf{T}} = (\tilde{\mathbf{T}}_1 \dots \tilde{\mathbf{T}}_n) \quad (4.15)$$

zwei Folgen von Operatoren mit  $T_i \in \mathcal{H} \rightarrow \mathbb{R}^{d_i}$  und  $\tilde{T}_i \in \mathcal{H} \rightarrow \mathbb{R}^{\tilde{d}_i}$ . Die durch Anwendung der Operatoren  $T_i$  auf eine Funktion  $f \in \mathbb{R}^2 \rightarrow \mathbb{R}$  entstehenden Werte, werden mittels der Notation

$$\mathbf{f}_T = \begin{pmatrix} T_1 f \\ \vdots \\ T_n f \end{pmatrix} \in \mathbb{R}^{\sum_{i=1}^n d_i} \quad (4.16)$$

zu einem Vektor zusammengefasst. Alle Werte der möglichen Anwendung der beiden Operatoren  $T_i$  und  $T_j$  auf eine Funktion

$$k : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R} \quad (4.17)$$

werden durch die Matrix

$$\mathbf{K}_{T\tilde{T}} = \begin{pmatrix} T_1^{(1)}\tilde{T}_1^{(2)}k & \dots & T_1^{(1)}\tilde{T}_m^{(2)}k \\ \vdots & & \vdots \\ T_n^{(1)}\tilde{T}_1^{(2)}k & \dots & T_n^{(1)}\tilde{T}_m^{(2)}k \end{pmatrix} \in \mathbb{R}^{(\sum_{i=1}^n d_i) \times (\sum_{i=1}^n \tilde{d}_i)} \quad (4.18)$$

zusammengefasst. Schließlich wird mit dem Spaltenvektor

$$\mathbf{k}_T \cdot (\mathbf{x}) = \begin{pmatrix} T_1 k(\cdot, \mathbf{x}) \\ \vdots \\ T_n k(\cdot, \mathbf{x}) \end{pmatrix} \in \mathbb{R}^{\sum_{i=1}^n d_i} \quad (4.19)$$

die Anwendung der Operatoren  $T_i$  auf das erste Argument der Funktion  $k$  bezeichnet. Analog besteht der Zeilenvektor  $\mathbf{k} \cdot T \in \mathbb{R}^{1 \times (\sum_{i=1}^n d_i)}$  aus Komponenten, die durch die Anwendung der Operatoren  $T_i$  auf das zweite Argument der Funktion  $k$  entstehen.

### 4.3.5 Inferenz

Die Berechnung der A-posteriori-Verteilung aus gegebenen Daten

$$\mathcal{Z} = \{(\mathbf{T}_i, \mathbf{q}_i, \mathbf{W}_i) \mid i = 1, 2, \dots, |\mathcal{Z}|\} \quad (4.20)$$

wird *Inferenz* genannt. Die Operatoren werden zu einer Folge

$$\mathbf{X} = (\mathbf{T}_1, \dots, \mathbf{T}_{|\mathcal{Z}|}) \quad (4.21)$$

zusammengefasst und die abhängigen Werte zu einem Vektor gestapelt

$$\mathbf{q} = (\mathbf{q}_1^T, \dots, \mathbf{q}_{|\mathcal{Z}|}^T)^T. \quad (4.22)$$

Die Unsicherheiten werden in einer Blockdiagonalmatrix

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_1 & & \\ & \ddots & \\ & & \mathbf{W}_{|\mathcal{Z}|} \end{pmatrix} \quad (4.23)$$

zusammengefasst. Bei der Gauß-Prozess-Regression wird zunächst eine Menge von Teststellen definiert, an denen die geschätzte Oberfläche evaluiert werden soll. In einer Formulierung mittels Operatoren, werden diese durch die Folge von Evaluationsfunktionalen

$$\mathbf{X}_\star = (\delta_{\mathbf{x}_\star 1}, \dots, \delta_{\mathbf{x}_\star |\mathcal{Z}|}) \quad (4.24)$$

repräsentiert. Die Funktionswerte der Teststellen sind gemäß Notationsvereinbarung mit  $\mathbf{f}_{\mathbf{X}_\star}$  notiert. Gesucht ist die Wahrscheinlichkeitsdichtefunktion über den Funktionswerten bei gegebenen Daten  $p(\mathbf{f}_{\mathbf{X}_\star} \mid \mathbf{q}, \mathbf{X}_\star, \mathbf{X})$ .

Die Herleitung der Schätzung erfolgt in zwei Schritten:

1. Zunächst wird die Verbundverteilung  $p(\underline{f}_{X_\star}, \underline{q} \mid X_\star, X)$  zwischen den Testwerten und den Trainingsdaten aufgestellt und
2. anschließend bezüglich der Trainingsdaten bedingt.

Zur Berechnung ersterer Wahrscheinlichkeitsdichtefunktion muss ihr Erwartungswert und ihre Kovarianz bestimmt werden. Die Kovarianz zwischen den beobachteten Werten ist

$$\text{Cov} \left\{ \underline{q}_i, \underline{q}_j \right\} = \text{Cov} \left\{ \underline{T}_i \underline{f} + \underline{w}_i, \underline{T}_j \underline{f} + \underline{w}_j \right\} \quad (4.25)$$

$$= \underline{T}_i^{(1)} \underline{T}_j^{(2)} k + \underline{W}_i \delta_{ij}, \quad (4.26)$$

beziehungsweise für den gesamten Datensatz

$$\text{Cov} \left\{ \underline{q}, \underline{q} \right\} = \underline{K}_{XX} + \underline{W}. \quad (4.27)$$

Die Kovarianz zwischen einem beobachteten Wert und einem Funktionswert ist gegeben durch

$$\text{Cov} \left\{ \underline{q}_i, \underline{f}(\underline{x}_{\star j}) \right\} = \text{Cov} \left\{ \underline{T}_i \underline{f} + \underline{w}_i, \underline{f}(\underline{x}_{\star j}) \right\} \quad (4.28)$$

$$= \underline{T}_i^{(1)} k(\cdot, \underline{x}_j) \quad (4.29)$$

$$= \underline{T}_i^{(1)} \delta_{\underline{x}_{\star j}}^{(2)} k, \quad (4.30)$$

beziehungsweise für den gesamten Datensatz durch

$$\text{Cov} \left\{ \underline{q}, \underline{f}_{X_\star} \right\} = \underline{K}_{XX_\star}. \quad (4.31)$$

Analog enthält man aus Symmetriegründen  $\text{Cov} \left\{ \underline{f}_{X_\star}, \underline{q} \right\} = \underline{K}_{X_\star X}$ . Unter Zuhilfenahme der obigen Gleichungen (4.27) und (4.31) für die Kovarianzen erhält man insgesamt für die Verbundverteilung:



$$p(\underline{f}_{X_\star}, \underline{q} \mid X_\star, X) = \mathcal{N} \left( \begin{pmatrix} \underline{f}_{X_\star} \\ \underline{q} \end{pmatrix}; \begin{pmatrix} \underline{\mu}_{X_\star} \\ \underline{\mu}_X \end{pmatrix}, \begin{pmatrix} \mathbf{K}_{X_\star X_\star} & \mathbf{K}_{X_\star X} \\ \mathbf{K}_{X X_\star} & \mathbf{K}_{X X} + \mathbf{W} \end{pmatrix} \right). \quad (4.32)$$

Hieraus folgt für die gesuchte A-posteriori-Wahrscheinlichkeitsdichtefunktion mittels der Identität über die Umformung von Normalverteilungen; siehe Anhang A.1

$$p(\underline{f}_{X_\star} \mid \underline{q}, X_\star, X) = \mathcal{N} \left( \underline{f}_{X_\star}; \underline{\mu}_g, \underline{\Sigma}_g \right) \quad (4.33)$$

mit

$$\underline{\mu}_g = \underline{\mu}_{X_\star} + \mathbf{K}_{X_\star X} \mathbf{K}^{-1} (\underline{q} - \underline{\mu}_X) \quad (4.34)$$

$$\underline{\Sigma}_g = \mathbf{K}_{X_\star X_\star} - \mathbf{K}_{X_\star X} \mathbf{K}^{-1} \mathbf{K}_{X X_\star} \quad (4.35)$$

$$\mathbf{K} = \mathbf{K}_{X X} + \mathbf{W}.$$

Eine anschauliche Interpretation des Erwartungswertes der A-posteriori-Wahrscheinlichkeitsdichtefunktion erhält man, wenn man die Schätzung für eine Stelle  $\mathbf{x}_\star$  betrachtet. Hierfür können für feste  $i$  die transformierten Kovarianzfunktionen

$$\phi_i(\mathbf{x}_\star) = \mathbf{T}_i^{(2)} k(\mathbf{x}_\star, \cdot) \quad (4.36)$$

als Funktion von  $\mathbf{x}_\star$  als Basisfunktionen interpretiert werden.

Somit kann der Erwartungswert der A-posteriori-Wahrscheinlichkeitsdichtefunktion als eine gewichtete Summe von Basisfunktionen

$$\mathbb{E} \left\{ \underline{f}(\mathbf{x}_\star) \mid \underline{q}, X_\star, X \right\} = \sum_{i=0}^n \alpha_i^T \phi_i(\mathbf{x}_\star) + \underline{\mu}(\mathbf{x}_\star). \quad (4.37)$$

interpretiert werden, dabei werden die Gewichte mit

$$\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^T, \dots, \boldsymbol{\alpha}_{|\mathcal{Z}|}^T)^T = \mathbf{K}^{-1} (\mathbf{q} - \boldsymbol{\mu}_X) \quad (4.38)$$

berechnet.

### 4.3.6 Approximationsverfahren

Die Berechnung der A-posteriori-Kovarianz (4.35) und des A-posteriori-Erwartungswertes (4.34) erfordern die Invertierung der Kovarianzmatrix, deren Dimension quadratisch mit den Daten wächst. Beispielsweise hat die Matrix im Falle der Inferenz mit Punkten eine Größe von  $|\mathcal{Z}| \times |\mathcal{Z}|$  und im Falle der Inferenz mit Gradienten die Größe  $3|\mathcal{Z}| \times 3|\mathcal{Z}|$ . Die Invertierung hat den Aufwand  $\mathcal{O}(\bar{d}|\mathcal{Z}|^3)$ , wobei  $\bar{d}$  die durchschnittliche Dimension der Beobachtungsgrößen ist. Damit ist für große Datenmengen die exakte Berechnung nicht praktikabel. Ist die Kovarianzmatrix invertiert, kann die A-posteriori-Kovarianz in  $\mathcal{O}(\bar{d}|\mathcal{Z}|^2)$  evaluiert werden. In der Praxis kann bei einer großen Menge an Evaluationspunkten diese Operation ebenfalls aufwendig sein.

Für die praktische Anwendung mit großen Datenmengen ist somit eine Approximation erforderlich. Eine mögliche Methode besteht darin, nur eine kleine Menge der Trainingsdaten zur Schätzung der Funktion zu verwenden. Der Nachteil dieser Approximationsmethode ist, dass nicht alle Trainingsdaten für die Schätzung berücksichtigt werden und damit Information unberücksichtigt bleibt. Die Nyströmmethode [Wil01] approximiert die Kovarianzmatrix durch eine Matrix geringeren Rangs. Eine große Klasse von Methoden sind die sogenannten *Sparse-Approximation-Methoden*. Darunter fallen Arbeiten wie [Sne06] oder [See03]. Die Idee hierbei ist, die zu schätzende Funktion durch eine kleinere Menge von Punkten (die nicht notwendigerweise zu den Trainingsdaten gehören) anstatt den gesamten

Trainingsdaten zu repräsentieren. Diese Menge von Punkten hat in der Literatur viele unterschiedliche Bezeichnungen wie zum Beispiel Pseudopunkte, Stützvektoren oder induzierte Punkte. Letztere wird in der Arbeit [QC05] als eine universelle Bezeichnung dieser Punkte eingeführt einschließlich einer Approximationsmethodik, mit der viele der Sparse-Methoden vereinheitlicht beschrieben werden können. Diese Methodik wird im Folgenden in einer verallgemeinerten Form mittels beliebiger Messoperatoren vorgestellt und anschließend die in dieser Arbeit verwendete Approximationsinstanz beschrieben. Die Lösung wird weiterhin in eine Form gebracht, die es erlaubt, die Schätzung iterativ zu aktualisieren.

Die induzierten Punkte bestehen aus einer Menge von Punkten

$$\mathcal{U} = \{ \mathbf{u}_i \in \mathbb{R}^2 \mid i = 1, \dots, |\mathcal{U}| \}, \quad (4.39)$$

deren Operatoren mit  $\mathbf{U} = (\delta_{\mathbf{u}_1}, \dots, \delta_{\mathbf{u}_{|\mathcal{U}|}})$  und deren zugehörigen Funktionswerte mit dem Vektor  $\mathbf{f}_{\mathbf{U}}$  bezeichnet werden. Der Zusammenhang zwischen den Trainingspunkten, der Testpunkte und den induzierten Punkten ist durch

$$p(\mathbf{f}_{\mathbf{X}_*}, \mathbf{f}_{\mathbf{X}}) = \int p(\mathbf{f}_{\mathbf{X}_*}, \mathbf{f}_{\mathbf{X}}, \mathbf{f}_{\mathbf{U}}) d\mathbf{f}_{\mathbf{U}} \quad (4.40)$$

$$= \int p(\mathbf{f}_{\mathbf{X}_*}, \mathbf{f}_{\mathbf{X}} \mid \mathbf{f}_{\mathbf{U}}) p(\mathbf{f}_{\mathbf{U}}) d\mathbf{f}_{\mathbf{U}} \quad (4.41)$$

gegeben. Vielen Sparse-Approximations-Methoden liegt nun die fundamentale Annahme zu Grunde, dass die Trainings- und Testmenge bedingt unabhängig von den induzierten Punkten sind. Dadurch erhält man aus er Gleichung (4.41) weiter

$$p(\mathbf{f}_{\mathbf{X}_*}, \mathbf{f}_{\mathbf{X}}) \approx \int p(\mathbf{f}_{\mathbf{X}_*} \mid \mathbf{f}_{\mathbf{U}}) p(\mathbf{f}_{\mathbf{X}} \mid \mathbf{f}_{\mathbf{U}}) p(\mathbf{f}_{\mathbf{U}}) d\mathbf{f}_{\mathbf{U}} \quad (4.42)$$

und mittels weiterer Umformung die Wahrscheinlichkeitsdichtefunktion

$$p(\mathbf{f}_{X_*} | \mathbf{f}_X) = \frac{p(\mathbf{f}_{X_*}, \mathbf{f}_X)}{p(\mathbf{f}_X)} \quad (4.43)$$

$$\approx \int p(\mathbf{f}_{X_*} | \mathbf{f}_U) p(\mathbf{f}_U | \mathbf{f}_X) d\mathbf{f}_U. \quad (4.44)$$

In der obigen Gleichung (4.44) sieht man, dass Trainingsphase und Evaluierungsphase getrennt betrachtet werden können. Aus den Trainingsdaten kann zunächst die Wahrscheinlichkeitsdichtefunktion der induzierten Punkte berechnet werden. Sie fasst verlustbehaftet die gesamte Information der Trainingsdaten zusammen. Schließlich kann (ohne Verwendung der Trainingsdaten) die Wahrscheinlichkeitsdichtefunktion der Evaluierungspunkte berechnet werden. Der erste Faktor im Integral kann hierbei als Glättung der Werte der induzierten Punkte interpretiert werden. Die exakte Lösung für beide Terme ist

$$p(\mathbf{f}_U | \mathbf{f}_X) = \mathcal{N}(\mathbf{f}_U; \mathbf{K}_{UX} \mathbf{K}^{-1} \mathbf{f}_X, \mathbf{K}_{UU} - \mathbf{K}_{UX} \mathbf{K}^{-1} \mathbf{K}_{XU}) \quad (4.45)$$

$$p(\mathbf{f}_{X_*} | \mathbf{f}_U) = \mathcal{N}(\mathbf{f}_{X_*}; \mathbf{K}_{X_*U} \mathbf{K}_{UU}^{-1} \mathbf{f}_U, \Sigma_{X_*|U}) \quad (4.46)$$

$$\Sigma_{X_*|U} = \mathbf{K}_{X_*X_*} - \mathbf{K}_{X_*U} \mathbf{K}_{UU}^{-1} \mathbf{K}_{UX_*} \quad (4.47)$$

$$\mathbf{K} = \mathbf{K}_{XX} + \mathbf{W}. \quad (4.48)$$

Zunächst ist nichts gewonnen, da die Gleichung (4.45) ebenfalls die Invertierung der Kovarianzmatrix bezüglich der Trainingsdaten erfordert. Die Approximationsverfahren unterscheiden sich nun darin, wie sie die beiden Wahrscheinlichkeitsdichtefunktionen (4.45) und (4.46) approximieren.

In dieser Arbeit wird für die Wahrscheinlichkeitsdichtefunktion (4.45) folgender Ansatz zur Approximation genutzt. Der Gauß-Prozess wird durch eine Funktion  $\tilde{f}$  approximiert, der die Funktionswerte mittels der induzierten Punkte berechnet. Es wird gezeigt, dass diese Funktion ebenfalls ein

Gauß-Prozess ist, allerdings mit einer degenerierten Kovarianzmatrix, das heißt einer Kovarianzmatrix, die eine endliche Entwicklung mittels Basisfunktionen besitzt. Dann kann der Rang der Kovarianzmatrix nicht größer als jene endliche Anzahl sein, womit sich der Aufwand der Invertierung der Kovarianzmatrix reduzieren lässt. Die Approximation  $\underline{\tilde{f}}$  berechnet sich aus einer Linearkombination von Kernen an den induzierten Punkten

$$\underline{\tilde{f}}(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x})\underline{\alpha} + \mu(\mathbf{x}) \quad (4.49)$$

mit  $\mathbf{\Phi}(\mathbf{x}) = \mathbf{k} \cdot \mathbf{u}(\mathbf{x})$ . Für die Wahrscheinlichkeitsdichtefunktion der Gewichte  $\underline{\alpha}$  wird gefordert, dass die Marginaldichte an den induzierten Punkten dem exakten Wert des zu approximierenden A-priori-Gauß-Prozesses entspricht:  $\underline{\tilde{f}}_{\underline{U}} \sim \mathcal{N}(\underline{\mu}_U, \mathbf{K}_{UU})$ . So kann sichergestellt werden, dass die Approximation an den induzierten Punkten exakt ist. Aus der Forderung erhält man für den Erwartungswert der Gewichte

$$\mathbb{E} \left\{ \underline{\tilde{f}}_{\underline{U}} \right\} = \mathbf{K}_{UU} \mathbb{E} \left\{ \underline{\alpha} \right\} + \underline{\mu}_U = \underline{\mu}_U \Rightarrow \mathbb{E} \left\{ \underline{\alpha} \right\} = \mathbf{0}, \quad (4.50)$$

für die Kovarianz

$$\text{Cov} \left\{ \underline{\tilde{f}}_{\underline{U}} \right\} = \mathbf{K}_{UU} \text{Cov} \left\{ \underline{\alpha} \right\} \mathbf{K}_{UU}^T = \mathbf{K}_{UU} \Rightarrow \text{Cov} \left\{ \underline{\alpha} \right\} = \mathbf{K}_{UU}^{-1} \quad (4.51)$$

und insgesamt  $\underline{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{UU}^{-1})$ . Da  $\underline{\alpha}$  normalverteilt ist, ist die Zufallsfunktion  $\underline{\tilde{f}}$  ein Gauß-Prozess und deren Erwartungswertfunktion ist durch

$$\underline{\tilde{\mu}}(\mathbf{x}) = \mathbb{E} \left\{ \mathbf{\Phi}(\mathbf{x})\underline{\alpha} + \mu(\mathbf{x}) \right\} = \mu(\mathbf{x}) \quad (4.52)$$

und deren Kovarianzfunktion durch

$$\tilde{k}(\mathbf{x}_1, \mathbf{x}_2) = \text{Cov} \left\{ \tilde{\underline{f}}(\mathbf{x}_1), \tilde{\underline{f}}(\mathbf{x}_2) \right\} \quad (4.53)$$

$$= \mathbf{\Phi}(\mathbf{x}_1) \text{Cov} \left\{ \underline{\alpha} \right\} \mathbf{\Phi}^T(\mathbf{x}_2) \quad (4.54)$$

$$= \mathbf{\Phi}(\mathbf{x}_1) \mathbf{K}_{UU}^{-1} \mathbf{\Phi}^T(\mathbf{x}_2) \quad (4.55)$$

gegeben. Der Erwartungswert und die Kovarianzmatrix der A-posteriori-Wahrscheinlichkeitsdichtefunktion ergeben sich anschließend anhand der Gleichung (4.45) unter Verwendung der hergeleiteten approximierten Größen für die Kovarianzfunktion und Erwartungswertfunktion:

$$\boldsymbol{\mu}_a = \boldsymbol{\mu}_U + \mathbf{K}_{UX} \mathbf{K}_a^{-1} (\mathbf{q} - \boldsymbol{\mu}_X) \quad (4.56)$$

$$\boldsymbol{\Sigma}_a = \mathbf{K}_{UU} - \mathbf{K}_{UX} \mathbf{K}_a^{-1} \mathbf{K}_{XU} \quad (4.57)$$

$$\mathbf{K}_a = \mathbf{K}_{XU} \mathbf{K}_{UU}^{-1} \mathbf{K}_{UX} + \mathbf{W} . \quad (4.58)$$

Durch die Anwendung der Woodbury-Matrix-Identität [Pet12] können die Gleichungen weiter umgeformt werden, so dass zum einen nur eine Matrix mit den Dimensionen  $|\mathcal{U}| \times |\mathcal{U}|$  invertiert werden muss und zum anderen die Gleichungen iterativ berechnet werden können:

$$\boldsymbol{\mu}_a = \boldsymbol{\mu}_U + \mathbf{K}_{UU} \mathbf{V}_a^{-1} \mathbf{d} \quad (4.59)$$

$$\boldsymbol{\Sigma}_a = \mathbf{K}_{UU} \mathbf{V}_a^{-1} \mathbf{K}_{UU} \quad (4.60)$$

$$\mathbf{V}_a = \mathbf{K}_{UU} + \mathbf{K}_{UX} \mathbf{W}^{-1} \mathbf{K}_{XU} \quad (4.61)$$

$$\mathbf{d} = \mathbf{K}_{UX} \mathbf{W}^{-1} (\mathbf{q} - \boldsymbol{\mu}_X) . \quad (4.62)$$

Eine detaillierte Herleitung der Gleichungen ist in Anhang A.2 gegeben.

In der Evaluierungsphase wird mittels der Wahrscheinlichkeitsdichtefunktion (4.46) und der Gleichung (4.44) die Evaluationspunkte ausgewertet. Hier kann die gleiche Approximation wie (4.59)-(4.62) verwendet werden.

Die Herleitung ist analog, dabei muss nur  $X_\star$  anstatt  $U$  und  $U$  anstatt  $X$  verwendet werden. In Abbildung 4.2 ist das Ergebnis einer Schätzung eines Kugelausschnitts mittels eines Gauß-Prozesses visualisiert. Als Messoperatoren wurden das Evaluationsfunktional und der Gradientenoperator verwendet, mit denen jeweils der Höhenwert und die Neigungen extrahiert wurden. Es wurden 500 Höhenwerte und 500 Neigungswerte genutzt. Zur approximativen Inferenz wurden  $25 \times 25$  induzierten Punkte in einem Gitter gewählt. Es ist ein deutlicher Zusammenhang zwischen hoher Varianz und hohem Fehler zu erkennen.

### Iterative Berechnung

Während der Planung werden Messungen zur Schätzung hinzugefügt oder einzelne Messungen separat variiert. Hier ist es nützlich, wenn die Schätzung iterativ erfolgen kann. Die Formeln (4.61) und (4.62) befinden sich schon in einer Form, in der sie sich leicht iterativ berechnen lassen. Zur iterativen Berechnung wird  $X$  in die einzelnen Elemente aufgespalten in einelementige Sequenzen von Messoperatoren  $X_i = (T_i)$ .

Die Größen  $V_k$  und  $d_k$  werden zunächst mittels

$$V_0 = K_{UU} \quad (4.63)$$

$$d_0 = \mathbf{0} \quad (4.64)$$

initialisiert. Wie aus (4.61) und (4.62) leicht ersichtlich ist, kann die Messung  $k + 1$  wie folgt hinzugefügt werden:

$$d_{k+1} = d_k + K_{UX_k} W_i^{-1} (q_k - \mu_{X_k}) \quad (4.65)$$

$$V_{k+1} = V_k + K_{UX_k} W_i^{-1} K_{X_k U} . \quad (4.66)$$

Zur Auswertung des Erwartungswertes und der Kovarianzmatrix muss  $V_k$  einmal invertiert werden. Dies hat einen Aufwand von  $O(|\mathcal{U}|^3)$ .

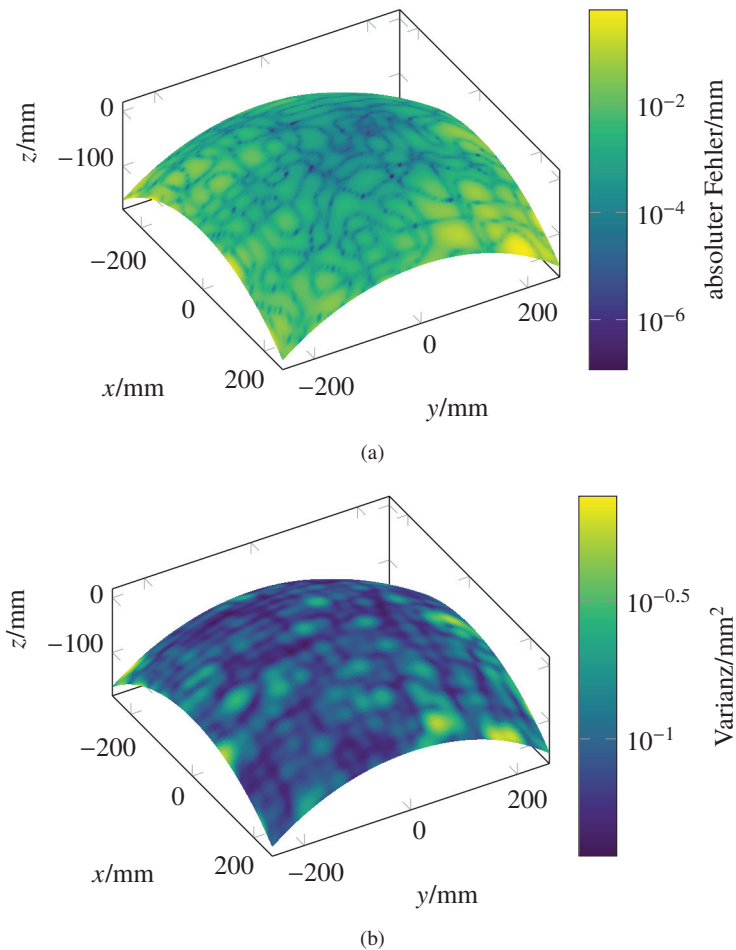


Abbildung 4.2: Rekonstruierter Ausschnitt einer Kugeloberfläche anhand eines Gaußprozesses mittels Höhenwerten und Neigungen. Farblich dargestellt sind (a) der absolute Fehler und (b) die Varianz der Schätzung in logarithmischer Skala.



### 4.3.7 Inferenz bei unbekanntem Referenzmodell und bekannten Regularisierungspunkten

Ist bei der Deflektometrie keine Referenzoberfläche gegeben, so ist es, wie in Unterabschnitt 3.6.2 diskutiert, nicht möglich zu bestimmen an welchem Ort der Gradienten gemessen wurde. Stattdessen erhält man ein Gradientenfeld im gesamten Sichtbereich der Kamera. Ist mindestens ein Punkt auf der Oberfläche bekannt, so lässt sich die Position der Oberfläche dennoch schätzen. In [Wer11] wird ein iteratives Verfahren für die Schätzung der Oberfläche in diesem Fall angegeben. Die Methode ist äquivalent zur Minimierung eines Funktionals, welches die quadratische Distanz zwischen den von der Messung induzierten Gradienten und der gesuchten Oberfläche berechnet unter den Nebenbedingungen, dass die Oberfläche nahe an den Regularisierungspunkten verläuft. Da Gauß-Prozesse, wie sie hier vorgestellt wurden, unterschiedliche Messoperatoren erlauben, lässt sich eine ähnliche Herangehensweise erzielen, indem die Regularisierungspunkte als das Resultat eines zusätzlichen Messoperators modelliert werden.

Seien  $\theta_{1:n}$  die Sensorkonfigurationen, mit denen gemessen wurde und seien  $I_{1:n}(u, v)$  die zugehörigen gemessenen deflektometrischen Registrierungen. Sei zusätzlich eine Menge von Regularisierungspunkten

$$\mathcal{Z}_r = \{(\delta_{x_i}, h_i) \mid i \in 1, 2, \dots, |\mathcal{Z}_r|\}$$
 (4.67)

gegeben, zum Beispiel die Höhe am Rand der Oberfläche. Die Oberfläche kann durch folgendes iteratives Verfahren geschätzt werden. In dem Iterationsschritt  $k$  liegt eine Schätzung  $\underline{f}_k$  der Oberfläche vor. Für den Iterationsschritt  $k = 0$  wird, falls kein bestimmtes Vorwissen vorliegt, die A-priori-Verteilung  $\underline{f} \sim \mathcal{GP}(0, k)$  mit geeigneter Kovarianzfunktion gewählt.

Der Erwartungswert der Schätzung  $E\left\{\underline{f}_{-k}\right\}$  dient im aktuellen Iterationsschritt  $k$  als Referenzoberfläche mittels der die Berechnung von Gradienten linearisiert wird; siehe Unterabschnitt 3.6.2. Mittels der linearen Regression wird anschließend die Schätzung der Oberfläche für den nächsten Schritt aktualisiert. Im Detail werden zunächst aus den deflektometrischen Registrierungen  $I_{1:n}$  aktualisierte Messungen anhand der aktuell geschätzten Oberfläche  $E\left\{\underline{f}_{-k}\right\}$  und dem Simulator berechnet:

$$\mathcal{Z}_{k+1} = \bigcup_{i=1}^n h_n(E\left\{\underline{f}_{-k}\right\}, I_i, \theta_i) \cup \mathcal{Z}_r. \quad (4.68)$$

Der Simulator wird über die mengenwertige Messabbildung  $h_n$  (siehe Abschnitt 3.1) beschrieben. Anhand aktualisierten Messdaten wird eine neue Schätzung berechnet

$$\underline{f}_{-k+1} \sim \mathcal{GP}(\mu_g, \Sigma_g), \quad (4.69)$$

deren Erwartungswert im nächsten Schritt als Referenzoberfläche verwendet wird. Der Algorithmus terminiert, wenn der Erwartungswert konvergiert. Die Unsicherheit der Schätzung wird aus der Schätzung des letzten Iterationsschritts extrahiert. Im Vergleich zur Schätzung mit Referenzoberfläche  $\mathcal{S}_r$  hängt diese implizit vom Erwartungswert des Gauß-Prozesses ab. Außerdem handelt es sich nur um einen approximativen Wert, da die Nichtlinearität der Messgleichung nicht berücksichtigt wird. In Abbildung 4.3 ist das Ergebnis einer Schätzung mittels der vorgestellten Methode visualisiert. Als Regularisierungspunkte wurde 40 Punkte am Rand der Oberfläche gewählt. Als Kovarianzfunktion dienten kubische B-Splines. Der Algorithmus terminiert nach 10 Iterationen; siehe Abbildung 4.4. Abbildung 4.5 zeigt die Schätzung der Oberfläche nur mittels der Regularisierungspunkte und nach einem Iterationsschritt.

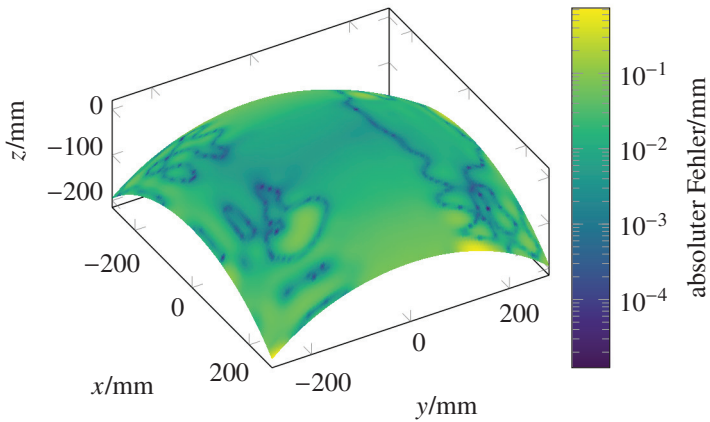


Abbildung 4.3: Absoluter Fehler der Schätzung mittels Gauß-Prozess und ohne Referenzoberfläche in logarithmischer Skala nach zehn Iterationen.

### 4.3.8 Gütekriterien

Alle im Kapitel 2 vorgestellten kovarianzbasierten Gütemaße können angewandt werden. Dabei muss beachtet werden, dass beim Gauß-Prozess die Kovarianzmatrix durch die Kovarianzfunktion zu ersetzen ist. Die Varianz an einem Punkt berechnet sich aus

$$V(\mathbf{x}) = \text{Cov} \left\{ \underline{f}(\mathbf{x}), \underline{f}(\mathbf{x}) \mid \mathbf{X}, \mathbf{q} \right\} \quad (4.70)$$

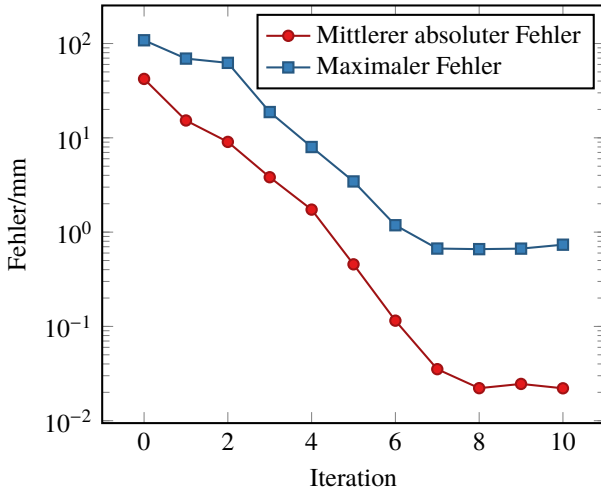


Abbildung 4.4: Mittlerer absoluter und maximaler Fehler der Schätzung mittels Gauß-Prozess ohne Referenzoberfläche in Abhängigkeit des Iterationsschritts.

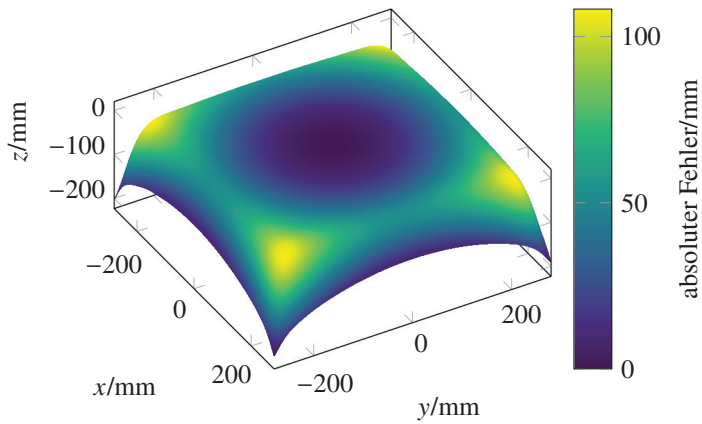
Hiermit lässt sich die totale Varianz durch die Integration der Varianzen an jedem Punkt

$$\int V(\mathbf{x}) \, d\mathbf{x} = \int \int (f_{\star} - E(f_{\star}))^2 p(f_{\star} | f_{\mathbf{X}}, \mathbf{X}, \mathbf{x}_{\star}) \, df_{\star} \, d\mathbf{x}_{\star} \quad (4.71)$$

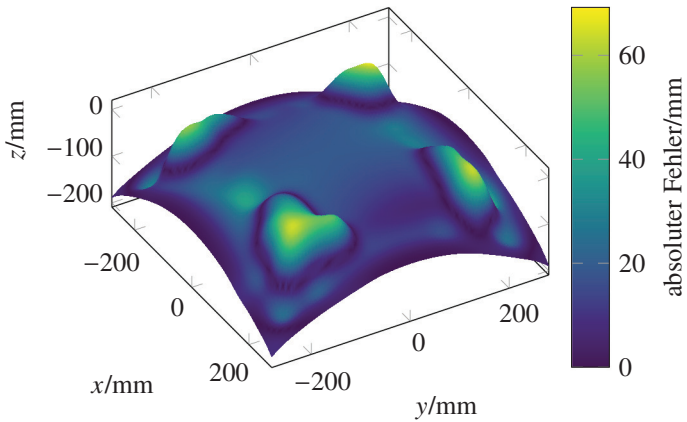
bestimmen. Diese Größe kann als gesamte Unsicherheit über der geschätzten Höhe interpretiert werden. Die maximale Varianz berechnet sich aus dem Optimierungsproblem

$$\max_x \text{Cov} \{ \underline{f}(\mathbf{x}), \underline{f}(\mathbf{x}) | \mathbf{X}, \mathbf{y} \} = \max_x \int (f_{\star} - E(f_{\star}))^2 p(f_{\star} | f_{\mathbf{X}}) \, df_{\star} .$$

Die Berechnung des größten Eigenwerts in funktioneller Form ist aufwendig und eignet sich nicht für die Praxis. Stattdessen wird der Gauß-Prozess diskretisiert. Die Kovarianzmatrix der so erzeugten Normalverteilung wird genutzt, um den größten Eigenwert zu bestimmen.



(a) Schätzung mit Höhendaten am Rand



(b) Schätzung nach einer Iteration

Abbildung 4.5: Rekonstruierter Ausschnitt einer Kugeloberfläche anhand eines Gauß-Prozesses mit Punkten und Gradienten ohne Referenzoberfläche. Zu sehen ist (a) der Erwartungswert der Oberflächenschätzung mit nur Höhendaten und (b) nach einer Iteration.

### 4.3.9 Modellauswahl

Die Wahl einer Kovarianzfunktion und seiner Parameter, der sogenannten *Hyperparameter* gehört dem Problem der Modellauswahl (engl. Model Selection) an. Die Modellauswahl ist ein aktives Forschungsgebiet. Es lassen sich daher eine Vielzahl von Methoden in der Literatur finden, von denen hier zwei genannt seien. Bei der Methode anhand der *Kreuzvalidierung* wird über einem diskreten Gitter im Parameterraum gesucht, wobei die Gütefunktion eine Abschätzung des Generalisierungsfehlers anhand der Kreuzvalidierung ist. Dieses Verfahren wird häufig zum Beispiel für SVMs verwendet und hat den Nachteil, dass es einen hohen Aufwand besitzt. Eine andere Methodik ist die *marginale Likelihood*  $p(\mathbf{y} | \mathbf{X})$  zu maximieren. In beiden Fällen geschieht die Modellauswahl auf Basis der Daten.

Für den Fall der Planung allerdings müssen die Parameter gewählt werden, bevor Daten vorhanden sind. Der Fehler einer Schätzung lässt sich in zwei Komponenten aufteilen, den *Approximationsfehler* und den *Schätzfehler*. Der Schätzfehler gibt den Kostenunterschied zwischen der vom Schätzer berechneten Funktion und der im Modell bestmöglichen Funktion an. Der Approximationsfehler gibt den Kostenunterschied zwischen der im Modell bestmöglichen Funktion und der bestmöglichen Funktion über alle Funktionen an. Der Approximationsfehler ist damit ein Maß für die Güte des Modells unabhängig von den Daten. In dieser Arbeit wird vorgeschlagen die Parameter über ein experimentelles Verfahren zu bestimmen mit dem Ziel den Approximationsfehler zu allen relevanten Funktionen gering zu halten. Wir beschränken uns auf den Fall einer Schätzung mit einer kubischen Spline-Kovarianzfunktion und die Wahl der induzierten Punkte in einem regelmäßigen Gitter. Der zu wählende Parameter in diesem Fall ist der Gitterabstand. Der Parameter soll so gewählt werden, dass der Approximationsfehler für die relevanten Funktionen niedrig ist. Die relevanten Funktionen sind eine Klasse von Funktionen  $\mathcal{D}$ , die die Defekte repräsentieren. Der

Erwartungswert des A-posteriori Gauß-Prozesses kann auch durch die Minimierung der Summe des quadratischen Fehlers und der Norm der Funktion in einem speziellen durch die Kovarianzfunktion erzeugten Hilbertraum bestimmt werden. Für eine steigende Anzahl an Messdaten geht der Einfluss des Regularisierers gegen Null [Ras05] (Kapitel 7). Aus diesem Grund ist es ausreichend nur den quadratischen Fehlerterm zu betrachten.

Sei  $T$  der Gitterabstand und  $\mathbf{u}_i(T)$  der  $i$ -te Punkt im Gitter, dann lässt sich der maximale quadratische Approximationsfehler in Abhängigkeit von  $T$  mittels des Ausdrucks

$$e_{\mathcal{D}}(T) = \max_{f \in \mathcal{D}} \int \left( f(\mathbf{x}) - \sum_{i=1}^{|\mathcal{U}|} \alpha_i k_{\text{cb}}(\mathbf{x}, \mathbf{u}_i(T)) \right)^2 d\mathbf{x} \quad (4.72)$$

berechnen, wobei  $k_{\text{cb}}$  die kubische B-Spline Kovarianzfunktion darstellt. Sei im Folgenden die Klasse von Defekten dargestellt durch die Funktionen

$$\mathcal{D} = \left\{ e^{-\frac{(\mathbf{x}-\mathbf{t})^2}{2L_d^2}} \mid \mathbf{t} \in \mathbb{R}^2, L_d \in \mathbb{R}_+ \right\} \quad (4.73)$$

wobei  $L_d$  die Defektbreite beeinflusst und  $\mathbf{t}$  die Position des Defekts ist. Da die Höhe des Defekts lediglich die Skalierung der Gewichte  $\alpha_1, \dots, \alpha_{\mathcal{D}}$  proportional beeinflusst, wurde sie hier auf eins normiert.

Für den eindimensionalen Fall ist der Approximationsfehler  $e_{\mathcal{D}}(T)$  für unterschiedliche Fehlerbreiten  $L_d$  in Abbildung 4.6 aufgetragen. Der Parameter  $T$  kann aus diesem Graphen abhängig von dem noch vertretbaren Fehler gewählt werden. Bei einer Schranke von beispielsweise  $10^{-3}$  muss für Defekte mit mindestens  $L_d = 30$  ein Gitterabstand kleiner 18 gewählt werden. Diese Methode eignet sich für Fehlermodelle mit wenigen Parametern, da der Aufwand für die Berechnung des Fehlers (4.72) mit der Dimension erheblich steigt.

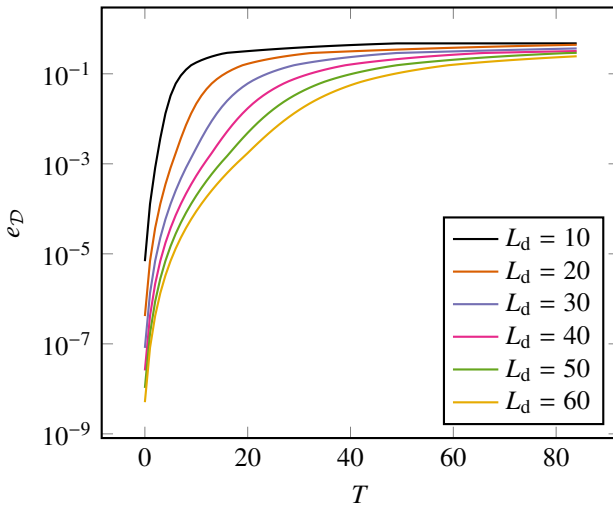


Abbildung 4.6: Darstellung des Approximationsfehlers  $e_{\mathcal{D}}$  in Abhängigkeit des Gitterabstandes  $T$  für Defekte mit  $L_d = 10, 20, \dots, 60$ .

## 4.4 Diskretisiertes Modell

Mit dem in Abschnitt 4.3 vorgestellten Modell wird die Höhe der zu inspizierenden Oberfläche geschätzt. Dies ist zeitaufwendig und auch nicht immer nötig. Je nach Inspektionsaufgabe sind relevante Merkmale, wie Neigungen, ausreichend. In diesem Abschnitt wird ein Modell vorgestellt, das die Oberfläche, anschaulich ausgedrückt, als eine Menge von kleinen Spiegeln repräsentiert. Hierzu wird die Oberfläche in Zellen partitioniert und jeder Zelle ein Spiegel zugeordnet, der über Parameter hinsichtlich Form und Lage beschrieben wird. Für eine probabilistische Beschreibung wird über diesen Parametern eine Verteilung definiert.



Sei  $\mathcal{P}_S = \{C_1, \dots, C_{|\mathcal{P}_S|}\}$  eine Partition der Oberfläche  $\mathcal{S}$ . Die Elemente  $C_i$  werden Zellen genannt. Die Oberfläche wird über eine Funktion  $\underline{f} : \mathcal{P}_S \rightarrow \mathbb{R}^d$  dargestellt, die jedem Element der Partition die Parameter der Spiegel zuordnet. Da die Partition aus einer endlichen Menge von Elementen besteht, kann die Wahrscheinlichkeitsdichtefunktion der Oberfläche  $\underline{f}$  über seine Verbundwahrscheinlichkeitsdichtefunktion beschrieben werden:  $\underline{f}_1, \dots, \underline{f}_{|\mathcal{P}_S|} \sim p(\underline{f}_1, \dots, \underline{f}_{|\mathcal{P}_S|})$  mit  $\underline{f}_i = \underline{f}(C_i)$ . Hier soll angenommen werden, dass die Parameter unabhängig sind, so dass ihre Wahrscheinlichkeitsdichtefunktion in ein Produkt von Marginaldichten über die Parameter  $p(\underline{f}_1, \dots, \underline{f}_{|\mathcal{P}_S|}) \approx p(\underline{f}_1) \cdot p(\underline{f}_2) \cdot \dots \cdot p(\underline{f}_{|\mathcal{P}_S|})$  zerfällt. Die Annahme der Unabhängigkeit wird wie folgt gerechtfertigt: Durch das Modellieren der A-priori-Wahrscheinlichkeitsdichtefunktion als das Produkt der Marginaldichten wird eine konservative Annahme über die Glattheit der Oberfläche und der Defekte gemacht. Damit erreicht der berechnete Plan eine gewisse Unabhängigkeit bezüglich dieser, da die Unsicherheitsabschätzung in diesem Fall ebenfalls konservativ ist. Während der Fusion von Messdaten kann es zu Abhängigkeiten zwischen den einzelnen Parametern kommen, wenn die Beobachtungen von mehreren Zellen abhängen. Das kann vorkommen, falls die Unschärfescheibe des Lichtstrahls auf der Oberfläche größer ist als eine Zelle. Werden sie unberücksichtigt gelassen, kommt es zu einer Unterschätzung der Varianz. Um diesen Effekt zu reduzieren, wird die Unsicherheitsscheibe während der Planung berücksichtigt. Die Strahlen, die eine zu große Unschärfescheibe besitzen, werden als unzulässig definiert. Schließlich führt die Unabhängigkeitsannahme zu einer Aufwandsreduzierung, weil die Zellen unabhängig voneinander aktualisiert werden können. Ohne Unabhängigkeitsannahme eignet sich die Modellierung der Verteilung als ein Markow-Zufallsfelder.

### 4.4.1 Ebene Spiegel

Das vorgestellte allgemeine Modell soll in diesem Unterabschnitt konkretisiert werden. Als Form der Spiegel wird eine Ebene gewählt. Sie modelliert die Neigung der Oberfläche in der Zelle. Die Position der Zellen wird im Zentrum der Zelle fixiert. Das Oberflächenmodell wird mit  $\mathbf{n} : \mathcal{P}_S \rightarrow \mathbb{R}^3$  bezeichnet und ordnet jedem Spiegel seinen Normalenvektor zu. Da die Zellen unabhängig voneinander betrachtet werden können, wird im Folgenden auf die Angabe des Zellenindex verzichtet.

Zunächst wird jeder Zelle  $\mathcal{C}$  ein Referenznormalenvektor zugeordnet. Er berechnet sich aus dem Durchschnitt der Normalenvektoren in der Zelle

$$\mathbf{n}_r \propto \int_{\mathcal{C}} \hat{\mathbf{n}}_{S_r}(\mathbf{r}) \, d\mathbf{r}, \quad (4.74)$$

wobei  $\hat{\mathbf{n}}_{S_r}(\mathbf{x})$  die Normale der Referenzoberfläche am Punkt  $\mathbf{x}$  ist. Unter der Annahme, dass das Beobachtungsruschen niedrig ist und der zu messende Normalenvektor nur wenig von dem Referenznormalenvektor abweicht, lassen sich die Rauschterme auf der Ebene senkrecht zum Referenznormalenvektor  $\mathbf{n}_r$  formulieren. Die A-priori-Wahrscheinlichkeitsdichtefunktion wird durch eine Normalverteilung

$$p_0(\mathbf{n}) = \mathcal{N}\left(\mathbf{n}; \hat{\mathbf{n}}_r, \mathbf{A}\Sigma_P\mathbf{A}^T\right) \quad (4.75)$$

modelliert, wobei  $\Sigma_P \in \mathbb{R}^{2 \times 2}$  die A-priori-Kovarianzmatrix und  $\mathbf{A}^T \in \mathbb{R}^{2 \times 3}$  der Projektor auf die Ebene senkrecht zur Referenznormalenvektor  $\hat{\mathbf{n}}_r$  ist mit  $\mathbf{A}^T\mathbf{A} = \mathbf{I}_{2 \times 2}$ .

Die Beobachtungen bestehen aus den Normalenvektoren, deren zugehörige Messstellen in die Zelle  $\mathcal{C}$  fallen. Der Zusammenhang zwischen dem beobachteten Normalenvektor  $\mathbf{n}_o$  und dem zu messenden Normalenvektor  $\mathbf{n}$  lässt sich über eine Taylorreihenentwicklung erster Ordnung an der Stelle  $\underline{\mathbf{w}} = \mathbf{0}$

approximieren. Durch Ausnutzen der Eigenschaft  $\mathbf{A}\mathbf{n} = 0$  des Projektors  $\mathbf{A}$  ergibt sich hierfür

$$\underline{\mathbf{n}}_o = \frac{\mathbf{n} + \mathbf{A}\mathbf{w}}{\|\mathbf{n} + \mathbf{A}\mathbf{w}\|} \approx \mathbf{n} + \mathbf{A}\mathbf{w}, \quad (4.76)$$

wobei das Rauschen gemäß  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_o)$  mit  $\Sigma_o \in \mathbb{R}^{2 \times 2}$  verteilt ist. Zusammenfassend ergibt sich für die bedingte Dichte

$$p(\mathbf{n}_o | \mathbf{n}) = \mathcal{N}(\mathbf{n}_o; \mathbf{0}, \mathbf{A}\Sigma_o\mathbf{A}^T). \quad (4.77)$$

Liegen für die Zelle  $\mathcal{C}$  die Beobachtungen

$$\mathcal{Z}_C = \left\{ (\mathbf{n}_1, \Sigma_o^1), \dots, (\mathbf{n}_{|\mathcal{Z}_C|}, \Sigma_o^{|\mathcal{Z}_C|}) \right\} \quad (4.78)$$

bestehend aus einer Menge von Paaren aus gemessener Normalen und Unsicherheiten vor, so kann mittels der Anwendung der Inferenzformel für normalverteilte Größen aus Anhang A.1.4 und den Gleichungen (4.75) und (4.77) die A-posteriori-Wahrscheinlichkeitsdichtefunktion wie folgt iterativ berechnet werden:

$$p(\mathbf{n} | \mathbf{n}_1, \dots, \mathbf{n}_k, \mathbf{n}_{k+1}) = \mathcal{N}(\mathbf{n}; \boldsymbol{\mu}_n^{k+1}, \Sigma_n^{k+1}) \quad (4.79)$$

mit

$$\boldsymbol{\mu}_n^{k+1} = \boldsymbol{\mu}_n^k + \Sigma_n^k \left( \mathbf{A}\Sigma_o^k\mathbf{A}^T + \Sigma_n^k \right)^{-1} (\mathbf{n}_k - \boldsymbol{\mu}_n^k) \quad (4.80)$$

und

$$\Sigma_n^{k+1} = \Sigma_n^k - \Sigma_n^k \left( \mathbf{A}\Sigma_o^k\mathbf{A}^T + \Sigma_n^k \right)^{-1} \Sigma_n^k. \quad (4.81)$$

Da die Kovarianzmatrizen  $\Sigma_o^k$  und  $\Sigma_p$  nur auf der Ebene senkrecht zum Referenznormalenvektor definiert sind, haben die Matrizen  $\Sigma_n^k$  nur einen Rang von zwei und besitzen demnach keine Inverse. Statt der Inverse in den Gleichungen (4.80) und (4.81) wird eine Pseudoinverse in dem von den Spalten von  $A$  aufgespannten Raum berechnet.

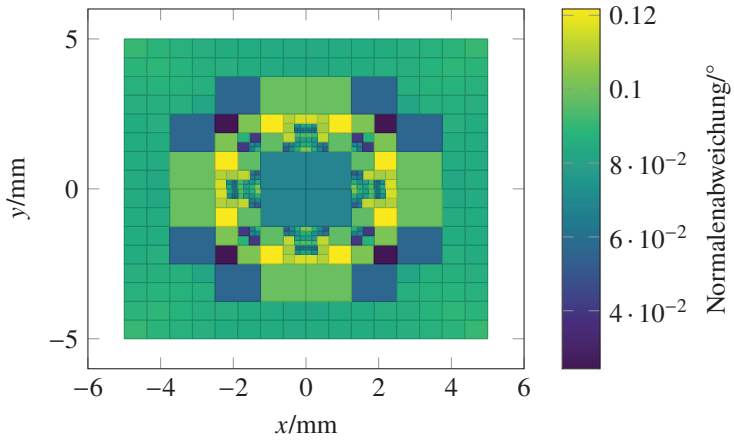
### Diskretisierungskriterien

Die Zellgröße ist ein entscheidender Parameter im Modell. Sie beeinflusst die Auflösung der Oberflächenmodellierung und sollte außerdem so gewählt werden, dass die Annahme über die Form der Spiegel gültig ist. Deshalb ist es nötig, sie abhängig von den lokalen Eigenschaften der Oberfläche zu wählen. Für das Aufteilen der Domäne wird ein hierarchischer Ansatz gewählt. Die Domäne wird anhand eines Quadrees iterativ zerlegt. Sollte die Domäne der Oberfläche nicht rechteckig sein, wird das kleinste Rechteck gewählt, in das die Domäne passt. Sollten die Seitenlängen der Domäne sich zu stark unterscheiden, kann eine Anfangsaufteilung durch ein Gitter gewählt werden, deren Zellen gleiche Seitenlängen besitzen und anschließend jede Zelle einzeln anhand eines Quadrees zerlegt werden. Der Algorithmus teilt die Zellen soweit auf, bis entweder eine Zelle ganz in einem nicht definierten Bereich liegt oder gewisse Terminierungskriterien erfüllt sind. Zellen, die nicht im Definitionsbereich der Oberfläche liegen, werden als solche markiert.

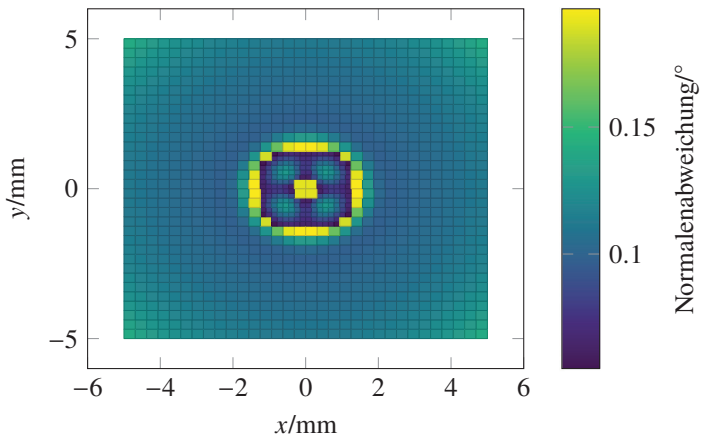
Zwei Terminierungskriterien werden betrachtet. Das erste Kriterium kontrolliert die Größe der Fläche, die die Zelle auf der Oberfläche einnimmt. Hierzu wird gefordert, dass die Zellfläche einen Schwellenwert  $A_{\max}$  nicht überschreiten darf

$$\int_C d\sigma \leq A_{\max}, \quad (4.82)$$

wobei  $d\sigma$  das Oberflächenelement der Referenzoberfläche ist. Das Flächenkriterium reicht zur Festsetzung der Oberflächenauflösung aus, da durch die



(a) Normalenkriterium



(b) Flächenkriterium

Abbildung 4.7: Diskretisierung einer Oberfläche mit Gauß-Delle im Zentrum anhand der beiden vorgestellten Kriterien.

Aufteilung sichergestellt wird, dass das Seitenverhältnis der Zelle approximativ eins ist. Sie sollte so gewählt werden, dass der kleinstmögliche relevante Defekt noch beobachtet werden kann. Die Ausdehnung der Zelle kann anhand des Abtasttheorems gewählt werden, das besagt, dass die Abtastrate mindestens die doppelte Frequenz wie die maximale Frequenz des Signals betragen muss.

Das zweite Kriterium stellt sicher, dass die Oberfläche in diesem Zellbereich auch als Ebene approximiert werden kann. Dazu wird gefordert, dass die maximale Abweichung der Normalenvektoren in der Zelle zu dem Referenznormalenvektor unter einem Schwellenwert  $\phi_{\max}$  liegt:

$$\forall \mathbf{r} \in \mathcal{C} : \mathbf{n}_r^T \hat{\mathbf{n}}_{S_r}(\mathbf{r}) < \phi_{\max} . \quad (4.83)$$

Die Summe aus dem Schwellenwert  $\phi_{\max}$  und der maximal gewünschte Standardabweichung für die Normalen  $\sqrt{U_{\text{suff}}}$  muss niedriger als die kleinste zu detektierende Neigungsabweichung sein.

In Abbildung 4.7 ist die Aufteilung einer Oberfläche mit einer Gauß-Delle im Zentrum anhand der beiden Kriterien dargestellt. Der Schwellenwert der Fläche ist auf  $A_{\max} = 0.12 \text{ mm}^2$  festgesetzt und die Winkelgenauigkeit auf  $\phi_{\max} = 0.12^\circ$ . Durch das Flächenkriterium können Zellen mit hohen Krümmungen entstehen, während durch das Normalenkriterium alleine Flächen entstehen können, die beliebig groß sind. Eine Kombination beider ist erforderlich. Ein Nachteil der Quadtreezerlegung ist, dass eine höhere Zerlegung entstehen kann, als gefordert, da eine Zelle bei noch nicht erfüllten Kriterium gevierteilt wird, unabhängig davon wie groß der Abstand zum Schwellenwert ist. Beim Flächenkriterium können so Zellen mit einer vierfach kleineren Fläche entstehen. Eine alternative Zerlegung ist die Zerlegung der Domäne in Dreiecke wie sie bei FEM Methoden stattfinden. Hier gibt es Methoden der adaptiven Aufteilung, die Kriterien, wie die Größe, Form

und Anzahl der Dreiecke berücksichtigen. Für einen Überblick siehe zum Beispiel [HL88].

### Gütemaße

Die Unsicherheit der Schätzung der Parameter einer einzelnen Zelle kann aus den A-posteriori-Verteilungen extrahiert werden. Die Kombination dieser Unsicherheiten zu einer Gesamtunsicherheit der Schätzung der Oberfläche muss gewichtet erfolgen, da die Zellen eine unterschiedliche Fläche besitzen. Die Varianz der Normalen einem Punkt  $\mathbf{r} \in \mathcal{S}_r$  ist durch

$$V(\mathbf{r}) = \sum_{i=1}^{|\mathcal{P}_S|} \mathbf{1}_{\mathbf{r} \in \mathcal{C}_i(\mathbf{r})} \text{Var} \{ \underline{\mathbf{n}}(\mathcal{C}_i) \} \quad (4.84)$$

gegeben. Da die Varianzen der Normalen in einer Zelle konstant sind, ergibt sich für die totale Varianz

$$\begin{aligned} U_T(b_k) &= \int V(\mathbf{r}) d\sigma \\ &= \sum_{i=1}^{|\mathcal{P}_S|} \int \mathbf{1}_{\mathbf{r} \in \mathcal{C}_i(\mathbf{r})} d\sigma \text{Var} \{ \underline{\mathbf{n}}(\mathcal{C}_i) \} \\ &= \sum_{i=0}^{|\mathcal{P}_S|} A_i \text{Var} \{ \underline{\mathbf{n}}(\mathcal{C}_i) \}, \end{aligned}$$

wobei  $A_i$  die Fläche der Zelle  $\mathcal{C}_i$  ist. Für die maximale Varianz wird keine Gewichtung benötigt. Sie berechnet sich aus

$$U(b_k) = \max_{\mathbf{x}} V(\mathbf{x}) = \max_{i=1, \dots, |\mathcal{P}_S|} \text{Var} \{ \underline{\mathbf{n}}(\mathcal{C}_i) \} .$$

## 4.5 Zusammenfassung

In diesem Kapitel wurden zwei Oberflächenmodelle vorgestellt. Sie werden während der Planung zur Fusionierung der Messungen in ein einheitliches Modell und zur Extraktion von Unsicherheit zur Bestimmung der Güte der Pläne verwendet.

Das erste vorgestellte Modell modelliert die Oberfläche als ein Gauß-Prozess. Ein Gauß-Prozess beschreibt eine Verteilung über Funktionen und ist vollständig durch die Definition einer Erwartungswertfunktion und einer Kovarianzfunktion beschrieben. Es lassen sich beliebige über lineare Operatoren beschriebene Messungen in das Modell fusionieren. Da der Aufwand der Inferenz mittels Gauß-Prozess kubisch mit den Beobachtungen wächst, wird eine Approximation vorgeschlagen, die die Oberfläche mittels einer endlichen Anzahl an sogenannten induzierten Punkten repräsentiert. Durch die Approximation wird allerdings gleichzeitig eine Glattheitsannahme über die Defekte getroffen, die beispielsweise bei der Position der induzierten Punkte in einem Gitter von der Gitterbreite abhängt.

Das zweite Modell, das diskrete Modell, stellt die Oberfläche als eine Menge kleiner Spiegel dar. Die Oberfläche wird in eine Menge von Zellen partitioniert, in der jeder Spiegel positioniert ist. Die Spiegel werden durch eine zu ihren Nachbarn unabhängige Wahrscheinlichkeitsverteilung über ihrer Form beschrieben. Speziell werden ebene Spiegel betrachtet mit der Neigung als Parameter. Die Wahrscheinlichkeitsverteilung wird als eine Normalverteilung im Tangentialraum modelliert. Es wird eine adaptive Diskretisierung vorgeschlagen, die sicherstellt, dass die maximale Fläche der Zellen sowie die Neigungsabweichung zum Referenznormalenvektor beschränkt sind. Damit lassen sich die Oberflächenauflösung und der A-Priori Neigungsfehler festsetzen.



## 5 Planungsverfahren

In Kapitel 2 wurde ein probabilistisches Rahmenwerk zur automatischen Planung einer Inspektionsaufgabe vorgestellt und die einzelnen Komponenten anschließend in den nachfolgenden Kapiteln konkretisiert. Dieses Kapitel setzt hiermit fort und stellt die letzte Komponente, das Planungsverfahren, vor. Das Ziel der Planung ist eine Folge von Sensorkonfigurationen zu finden, die die Unsicherheit bezüglich dem Wissen über die zu inspizierende Oberfläche auf eine gewünschte maximale Unsicherheit  $U_{\text{suff}}$  minimiert und gleichzeitig niedrige Kosten (zum Beispiel Anzahl der Messungen oder Messzeit) besitzt. Insbesondere ist die Anzahl der Messungen ebenfalls gesucht und bestimmt maßgeblich die Kosten. Das Optimierungsproblem lässt sich in der Form

$$n^*, \theta_{0:n-1}^* = \arg \min_{n, \theta_{0:n-1} \in \Theta^n} J_n(b_0, \theta_{0:n-1}) \quad (5.1)$$

formulieren, wobei  $b_0$  die A-priori-Verteilung über der Oberfläche und  $\Theta^n \subset \mathbb{R}^{nd}$  die zulässige Menge der Konfigurationen ist. Die Zielfunktion

$$J_n(b_0, \theta_{0:n-1}) = \sum_{i=0}^{n-1} \rho(b_i, \theta_i) + \rho_E(b_n) \quad (5.2)$$

mit

$$b_{i+1} = B(b_i, \theta_i, z_{i+1}) \quad (5.3)$$

besteht aus der Summe aller Schrittkosten  $\rho(b_i, \theta_i)$  und den Terminalkosten  $\rho_E(b_n)$ . Die Schrittkosten wurden in Gleichung (2.18) eingeführt und setzen sich aus der Unsicherheit der Oberfläche und den Kosten der Aktionen zusammen. Die Terminalkosten werden hier eingeführt, um sicherzustellen, dass der berechnete Plan genügend Information über die Oberfläche sammelt und die Informationsbeschaffungsphase beendet werden kann. Ein hoher Bestrafungsterm wird aufaddiert falls die maximale Varianz in  $b_n$  die in der Spezifikation geforderte Unsicherheit  $U_{\text{suff}}$  nicht erreicht, andernfalls sind die Terminalkosten Null:

$$\rho_E(b_k) = \begin{cases} U(b_0) & \text{falls } U_M(b_k) > U_{\text{suff}} \\ 0 & \text{sonst.} \end{cases} \quad (5.4)$$

Bei der Funktion  $B$  handelt es sich um die Transitionsfunktion des Belief-MDP; siehe Gleichung (2.8).

Bei (5.1) handelt es sich um ein nichtlineares Optimierungsproblem mit im Allgemeinen mehreren lokalen Minima. Die Dimension der zu optimierenden Variable  $\theta_{0:n-1}$  ist als eine Folge von Konfigurationen abhängig von der Anzahl der Messungen  $n$ . Eine einzelne Evaluation der Zielfunktion ist zeitaufwendig, da für jede Konfiguration eine Simulation der Messung an der Oberfläche und eine anschließende Fusion in das Schätzmodell erfolgen muss. Um die Komplexität des Planungsproblems zu reduzieren, werden zwei approximative Verfahren untersucht. Die Greedy-Planung approximiert das Optimierungsproblem (5.1) durch eine Folge von einfachen Optimierungsproblemen, die jeweils nur einen Zeitschritt im Voraus planen. Dies wird in Abschnitt 5.3 vorgestellt. In Abschnitt 5.4 wird eine weitere Methode vorgestellt, die in einem zweistufigen Verfahren auf der oberen Ebene über die Anzahl der Messungen optimiert und auf der unteren Ebene das Optimierungsproblem (5.1) bei fester Anzahl der Messungen löst.

Bevor auf die Planungsalgorithmen eingegangen wird, werden die Nebenbedingungen in Abschnitt 5.1 näher erläutert. Die Planungsalgorithmen reduzieren intern das Planungsproblem auf die Minimierung einzelner Sensorkonfigurationen über die zugehörigen Schrittkosten. Abschnitt 5.2 geht auf die verwendeten Optimierungsalgorithmen ein.

## 5.1 Nebenbedingungen

Nicht alle Konfigurationen aus  $\mathbb{R}^d$  sind erlaubt. Zum einen gibt es die natürlichen Nebenbedingungen, die durch die Domäne der Oberfläche und der Einschränkung der Winkeln entstehen, wie in dem Abschnitt 3.5 definiert. Zum anderen kann das Handhabungswerkzeug nur Positionen anfahren, die in seinem Arbeitsraum sind. Eine weitere Einschränkung kann durch hohe Lasten am Arm entstehen. Je nach Gewicht des Sensors können hohe Drehmomente an den Motoren entstehen, wenn der Arm ausgestreckt ist. Diese Positionen sollte deshalb ausgeschlossen werden. Mögliche Kollisionen zwischen Kamera, Schirm und Oberfläche schränken die Konfigurationen zusätzlich ein.

Konfigurationen, die beim Ausführen der zugehörigen Messung keine Stelle der Oberfläche beobachten, sind gemäß der Kostenfunktion zulässig. Die Berechnung der Schrittkosten liefert in diesem Fall die Kosten für den Aufwand und die Unsicherheit bleibt unverändert. Dennoch erweist es sich als vorteilhaft, diese Konfigurationen mit leerem Messbereich aus der Menge der zulässigen Konfigurationen auszuschließen, da die Gütefunktion über diese Menge konstant ist und es sich hierbei um eine große Menge handelt. Bei wiederholtem zufälligen Ziehen von Konfigurationen aus einer Gleichverteilung unter Berücksichtigung der natürlichen Einschränkungen, liegt die relative Anzahl von Konfigurationen mit leerem Messbereich bei über 90 Prozent. Dies führt zu Problemen bei Optimierungsalgorithmen,

insbesondere diejenigen, die zufallsbasierte Lösungskandidaten auswählen. Es können zwei Fälle unterschieden werden, die zu einem leeren Messbereich führen. In dem ersten Fall beobachtet die Kamera keinen Bereich der Oberfläche, in dem zweiten Fall kann die Reflexion des Bildschirms nicht beobachtet werden. Der erste Fall wird durch die Parametrisierung im Abschnitt 3.5 ausgeschlossen. Der zweite Fall muss über eine Simulation der Messung identifiziert werden. Effizient lässt sich die Prüfung auf Gültigkeit dadurch erreichen, indem nur wenige Strahlen bei der Simulation verfolgt werden. Hierdurch werden aber mehr Konfigurationen ausgeschlossen als nötig. Bei einer Messung mit kleinem Sichtbereich kann es vorkommen, dass eine hierfür geeignete Konfiguration als unzulässig identifiziert wird. Im Fall eines uneingeschränkten Sensors und keinen zu starken Einschränkungen der Positionierbarkeit gibt es aber immer eine andere Bildschirmposition, die den Sichtbereich zulässig macht. Im Fall eines kompakten Sensors oder starken Einschränkungen kann es sein, dass dies nicht möglich ist. Dann lässt sich aber durch eine zweite vollständige Simulation prüfen, ob eine zulässige Messung vorliegt. Bei zulässigen Messungen kann das Ergebnis der Simulation zur Evaluation der Schrittkosten weiterverwendet werden.

## 5.2 Optimierungsalgorithmen

Die Wahl eines geeigneten Optimierungsalgorithmus hängt von den Eigenschaften der Schrittkostenfunktion ab. Die hier betrachteten Schrittkostenfunktionen hängen nichtlinear von der Sensorkonfiguration ab. Sie besitzen viele lokale Minima, zum Beispiel in dem Fall, in dem bereits Messungen in den Zustand fusioniert worden sind. Hier existieren auch Plateaus, das heißt Bereiche im Konfigurationsraum, an denen die Zielfunktion sich nicht oder nur sehr schwach ändert. Dies ist immer der Fall, wenn Stellen beobachtet werden, die eine niedrige Unsicherheit besitzen, weil sie schon in einem früheren Schritt vermessen wurden. Außerdem hängt die

Differenzierbarkeit der Zielfunktion vom verwendeten Oberflächenmodell ab. Beispielsweise ist das Spiegelmodell aufgrund der Diskretisierung der Oberfläche nicht differenzierbar. Das Gauß-Prozess Modell dagegen ist kontinuierlich und differenzierbar. Aber auch hier kann es zu Schwierigkeiten kommen, beispielsweise an Stellen, an denen die in der Simulation verfolgten Lichtstrahlen die Oberfläche berühren.

Aufgrund dieser Schwierigkeiten werden ableitungsfreie Methoden für die Planung in der Deflektometrie untersucht. Diese zeichnen sich dadurch aus, dass sie nur Funktionsevaluationen verwenden. Bei Optimierungsalgorithmen wird zwischen lokalen und globalen Algorithmen unterschieden. Lokale Algorithmen konvergieren bei einem gegebenen Startwert gegen ein lokales Optimum, während globale Algorithmen versuchen das globale Optimum zu finden. Hier wird aufgrund der vorhandenen lokalen Minima in der Zielfunktion ein globaler Optimierungsalgorithmus bevorzugt. Weiterhin wird die Bedingung gestellt, dass der Algorithmus mit wenigen Funktionsevaluationen auskommen muss, da wie oben erwähnt die Evaluation der Zielfunktion aufwendig ist.

Eine Klasse von Optimierungsalgorithmen, die ableitungsfrei sind und unter welche auch genetische Algorithmen fallen, sind evolutionären Algorithmen. Diese Algorithmen sind von der Evolutionstheorie der Biologie motiviert. Sie erzeugen eine Population von Lösungskandidaten, die durch evolutionäre Operatoren wie Selektion, Rekombination und Mutation in neue, bessere Generationen entwickelt werden. Eine interessante Unterklasse bilden die Estimation of Distribution Algorithmen (EDA) [Hau11]. Sie stellen die Population als ein probabilistisches Modell dar, aus der Lösungskandidaten gezogen werden. Diese Verteilung wird iterativ entwickelt, so dass idealerweise die Wahrscheinlichkeit, Lösungskandidaten in der Umgebung des Optimums vorzuschlagen, steigt. Ein Nachteil dieser Klasse von Verfahren ist, dass sie viele Funktionsevaluationen benötigen.

Ein globaler Optimierungsalgorithmus, der im Vergleich zu den oben genannten, bereits mit weniger Iterationen gute Ergebnisse liefert, ist der Simulated-Annealing Algorithmus. Dieser wird in Unterabschnitt 5.2.1 vorgestellt und für die deflektometrische Oberflächeninspektion angepasst. Ein weiterer verwendeter Optimierungsalgorithmus ist der Nelder-Mead Algorithmus; siehe Unterabschnitt 5.2.2. Hierbei handelt es sich um einen lokalen Optimierungsalgorithmus. Er wird als ein Nachoptimierer verwendet, um das Ergebnis des Simulated-Annealing Algorithmus weiter zu verbessern.

### 5.2.1 Simulated-Annealing

Das Simulated-Annealing-Verfahren [Kir83] ist ein heuristisches Meta-Optimierungsverfahren. Motiviert ist das Verfahren durch den Abkühlvorgang von Metallen, bei dem sich durch langsames Abkühlen die Atome in einen energieärmeren Zustand positionieren. Im Simulated-Annealing Algorithmus beeinflusst die Temperatur die Wahrscheinlichkeit mit dem sich der Funktionswert in einem Schritt verschlechtern kann. Das steht im Gegensatz zu herkömmlichen deterministischen Optimierungsverfahren, wie dem Gradientenabstiegsverfahren, bei dem in jedem Schritt eine Lösung mit niedrigerem Funktionswert gefunden wird. Dies dient dazu aus lokalen Minima entfliehen zu können.

Das Simulated-Annealing-Verfahren ist ein iteratives Verfahren. Es speichert in jedem Schritt das bisher beste gefundene Ergebnis sowie eine aktuelle Stelle  $\theta$ , die iterativ verändert wird. In jedem Schritt wird gemäß einer Perturbationsfunktion aus der Nachbarschaft der vorherigen Stelle ein neuer zufälliger Lösungskandidat  $\theta_{\text{prop}}$  gewählt. Dieser Vorgang kann als das Ziehen einer Stichprobe aus einer bedingten Dichte  $p_{\text{prop}}(\cdot | \theta)$  modelliert werden. Sei  $J : \mathbb{R}^d \rightarrow \mathbb{R}$  die zu optimierende Zielfunktion, weiter wird der Wert  $J(\theta_{\text{prop}})$  mit dem vorherigen Wert  $J(\theta)$  verglichen und die

zugehörige Stelle abhängig von der Temperatur  $T$  mit einer gewissen Akzeptanzwahrscheinlichkeit  $p_a(J(\theta_{\text{prop}}), J(\theta), T)$  akzeptiert. Der Schritt endet mit der Reduzierung der Temperatur. Der Algorithmus terminiert, wenn die Endtemperatur  $T_e$  erreicht worden ist. In Algorithmus 1 ist das Simulated-Annealing-Verfahren als Pseudocode gegeben.

---

**Algorithm 1** Simulated-Annealing-Verfahren
 

---

```

 $T \leftarrow T_0$ 
while  $T > T_e$  do
   $\theta_{\text{prop}} \sim p_{\text{prop}}(\cdot \mid \theta)$ 
  if akzeptiere  $\theta_{\text{prop}}$  mit  $p_a(J(\theta_{\text{prop}}), J(\theta), T)$  then
     $\theta \leftarrow \theta_{\text{prop}}$ 
  end if
  reduziere  $T$ 
end while
return  $\theta$ 

```

---

Der Algorithmus enthält die drei Komponenten, Pertubationsfunktion, Akzeptanzfunktion und Temperaturplan, die geeignet gewählt werden müssen. Im „Standard“ Simulated-Annealing Verfahren wird für die Akzeptanzwahrscheinlichkeit die Boltzmannverteilung

$$p_a(E', E, T) = \begin{cases} e^{-\frac{E'-E}{T}} & \text{wenn } E' > E \\ 1 & \text{sonst} \end{cases} \quad (5.5)$$

gewählt. Für den Temperaturplan können unterschiedliche Abstiege, wie zum Beispiel ein linearer oder ein exponentieller gewählt werden.

**Wahl der Pertubationsfunktion**

Die Pertubationsfunktion erzeugt in Abhängigkeit der aktuellen Stelle eine zufällige neue Evaluationsstelle unter Zuhilfenahme der bedingten Dichte

$p_{\text{prop}}(\cdot | \theta)$ . Sie muss eine Abhängigkeit bezüglich der aktuellen Stelle besitzen, da sich das Verfahren sonst zu einer zufälligen Suche reduziert. Somit ist beispielsweise eine Gleichverteilung auf alle möglichen Konfigurationen keine gute Wahl. Für gewöhnlich wird bei einer kontinuierlichen Zielgröße zunächst eine zufällige Richtung gewählt und anschließend ein zufälliger Schritt in diese Richtung vollzogen. In der Deflektometrie führt die Wahl einer beliebigen Richtung zu einer unnötig hohen Anzahl an Evaluationen, wobei viele Konfigurationen niedrige Werte besitzen. Durch Ausnutzen der Anwendungsdomäne können die Richtungen eingeschränkt werden, um mit hoher Wahrscheinlichkeit relevante Konfigurationen zu erzeugen. Dabei kann auch die Information über den Zustand, wie beispielsweise die Varianz an jedem Ort auf der Oberfläche genutzt werden. Im Folgenden wird eine modulare bedingte Dichte vorgestellt, die es erlaubt relativ einfach Wissen in die Optimierung einzubringen. Sie muss dabei nicht in expliziter Form vorliegen, sondern in einer Form, in der effizient Stichproben gezogen werden können. Sie besteht aus einer Menge von Regeln  $\mathcal{R} = \{R_1, \dots, R_{|\mathcal{R}|}\}$ , wobei jede Regel  $R_i$  ein Tupel der Form  $R_i = (\Omega_i, p_i(\theta' | \theta), \alpha_i)$  ist mit

- $\Omega_i \subset \mathbb{R}^d$  die Menge aller Konfigurationen, bei der die Regel  $R_i$  angewendet werden kann,
- $p_i(\theta' | \theta)$  eine bedingte Wahrscheinlichkeitsdichte, aus der die neue Stelle zufällig gezogen werden kann und
- $\alpha_i \in \mathbb{R}$  ein Gewicht, das die Priorität der Regel beschreibt.

Wenn  $\Omega_i = \mathbb{R}^d$  dann ist die Bedingung trivial erfüllt und die Regel kann immer angewendet werden. Soll nun aus  $p_i(\theta' | \theta)$  von  $\theta$  aus eine neue Konfiguration zufällig gezogen werden, müssen zunächst alle Regeln bestimmt werden, die angewandt werden können. Dies ist der Fall wenn  $\theta \in \Omega_i$ . Sei die Menge aller in diesem Fall anwendbaren Regeln  $\mathcal{R}_\theta = \{i | \theta \in \Omega_i\}$ . Für alle  $i \in \mathcal{R}_\theta$  wird die Regel  $R_i$  mit der Wahrscheinlichkeit  $p_i = \frac{\alpha_i}{\sum_{j \in \mathcal{R}_\theta} \alpha_j}$



und die Konfiguration schließlich aus der Wahrscheinlichkeitsverteilung der Regel  $p_i(\theta'|\theta)$  gezogen.

### Regeln

Die Wahl der Regeln gibt die Richtung an, in denen der Algorithmus exploriert. Es werden Regeln ohne Bedingungen, also  $\Omega_i = \mathbb{R}^d$  betrachtet, deren Wahrscheinlichkeitsdichte über folgende Vorschrift beschrieben wird:

$$\theta_{\text{prop}} = \theta + \mathbf{A}\mathbf{v}, \quad (5.6)$$

wobei der Vektor  $\mathbf{v}$  eine Verschiebungsrichtung und die Skalierungsmatrix  $\mathbf{A} \in \mathbb{R}^{d \times d}$  eine zufällige Diagonalmatrix beschreiben. Die Diagonaleinträge der Matrix  $\mathbf{A}$  sind gemäß einer Gleichverteilung im Intervall  $[a_i, b_i]$  verteilt  $(\mathbf{A})_{ii} \sim U([a_i, b_i])$ . Die Intervalle  $[a_i, b_i]$  geben den Betrag der möglichen Skalierungen für die Komponenten an. Sei  $s \in \mathbb{R}^+$ , dann wird für die Verschiebungsintervalle  $[a_i, b_i]$  ein symmetrisches Intervall mit  $a_i = -s$  und  $b_i = s$  oder ein positives Intervall mit  $a_i = 0$  und  $b_i = s$  gewählt, je nachdem ob die Gegenrichtung mit einbezogen werden soll. Wenn  $\mathbf{v} = \mathbf{e}_i$  ist, wobei  $\mathbf{e}_i$  den  $i$ -te Einheitsvektor repräsentiert, dann beeinflusst die Regel nur eine Komponente der Sensorkonfiguration. Dabei wurden folgende Komponenten des Sensors zusammengefasst:

- Der Sichtpunkt  $\chi$  des Sensors,
- Elevation  $\vartheta_c$  und Azimuth  $\phi_c$  der Kamera
- Tilt  $\tau_c$  der Kamera
- Distanz  $d_c$  der Kamera
- Distanz  $d_l$  des Schirms
- Elevation  $\vartheta_1$  und Azimuth  $\phi_1$  des Schirms
- Orientierung  $\alpha_1, \beta_1$  und  $\gamma_1$  des Schirms

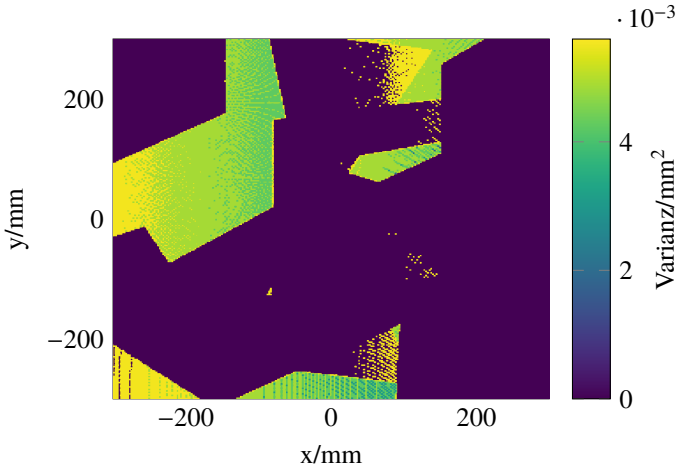
Diese Regeln beinhalten nur lokale Veränderungen der Komponenten. Wenn mehrere Messungen an einer Oberfläche ausgeführt worden sind, dann haben große teilweise unzusammenhängende Bereiche auf der Oberfläche eine niedrige Varianz. Für ein Beispiel siehe Abbildung 5.1a, die die Unsicherheit der Schätzung eines Abschnitts einer Kugeloberfläche nach 11 Messungen zeigt. Dieser Effekt führt dazu, dass die Optimierung der nächsten Sensor-konfiguration oft in flachen Bereichen konvergiert, weil sich die Varianz an gemessenen Positionen nicht oder nur schwach ändert. Dies erfordert eine Regel, die die Sensorkonfiguration so ändert, dass eine nicht beobachtete Region auf der Oberfläche beobachtet werden kann. In jedem Messschritt kann aus der aktuellen Schätzung eine *Unsicherheitskarte*  $U(\mathbf{x})$  aufgestellt werden, die in jedem Punkt die Unsicherheit angibt. Aus der Unsicherheitskarte der Schätzung wird die Wahrscheinlichkeitsdichtefunktion

$$p_u(\mathbf{x}) \propto \max(0, U(\mathbf{x}) - v_{\min}) \quad (5.7)$$

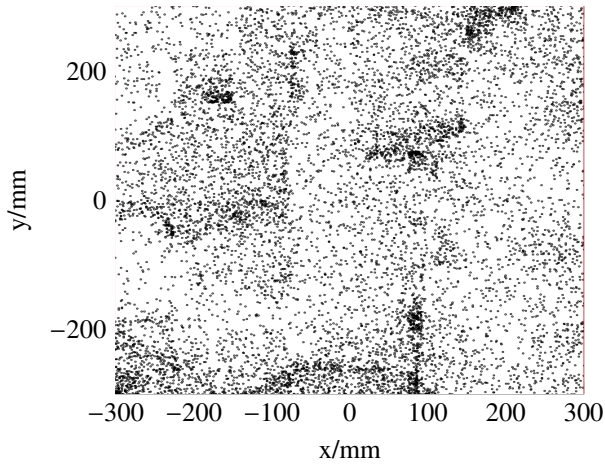
definiert, wobei  $v_{\min}$  für die minimale Unsicherheit steht, unterhalb derer die zugehörige Stelle nicht mehr gezogen werden darf. Der Wert von  $v_{\min}$  hängt von der zu erreichbaren Unsicherheit und der Inspektionsaufgabe ab. Für die hier vorgestellten Unsicherheitsmaße wird  $v_{\min} = U_{\text{suff}}$  gewählt. Die Wahrscheinlichkeitsverteilung (5.7) ist also an den Stellen größer als Null, die noch nicht zufriedenstellend beobachtet wurden. Die bedingte Dichte der Regel ist dann definiert mittels

$$\theta_{\text{prop}} = \begin{pmatrix} \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times (d-2)} \\ \mathbf{0}_{(d-2) \times 2} & \mathbf{I}_{(d-2) \times (d-2)} \end{pmatrix} \theta + \begin{pmatrix} \mathbf{x} \\ \mathbf{0}_{(d-2) \times 1} \end{pmatrix} \quad (5.8)$$

mit  $\mathbf{x} \sim p_u(\mathbf{x})$ . Die Regel überschreibt also die  $\chi$  Komponente und die neue Position ist unabhängig von dem aktuellen Lösungskandidaten. Abbildung 5.1b zeigt eine Stichprobe der Sichtpunkte aus der Verteilung definiert



(a) Varianz



(b) Samples

Abbildung 5.1: (a) Varianzverteilung nach 11 Messungen. Große unzusammenhängende Bereiche auf der Oberfläche haben eine niedrige Varianz. Die Wahrscheinlichkeit, dass die Pertubationsfunktion eine geeignete Perturbation durchführt, wird mit steigender Anzahl der Messungen niedriger. (b) Zufällige Menge von  $\chi$ , erzeugt durch einen Sampler, der unabhängig von der aktuellen Sensorkonfiguration proportional zur Varianz zieht.

durch die Unsicherheitskarte. Die Punkte in der Stichprobe liegen in Gebieten mit hoher Unsicherheit dichter zusammen.

Es bleibt nur noch die Gewichte für die Regeln zu wählen. Es lässt sich schon durch Ausprobieren die Signifikanz einiger Regeln feststellen. Beispielsweise hat die Distanz der Kamera und des Bildschirms großen Einfluss auf den Optimierungsverlauf.

Eine systematische Methode, ist die Schätzung der Gewichte basierend auf einer Folge von Daten  $\mathcal{D}_s = \{(i_k, v_k) \mid k \in \{1, \dots, m_a\}\}$  der Länge  $m_a$  bestehend aus dem Index der angewandten Regel  $i_k$  und dem Informationsgewinn  $v_k$ . Diese werden durch eine Zufallsbewegung im Raum der Sensorkonfigurationen anhand eines modifizierten Simulated-Annealing Algorithmus erzeugt. Der Startwert des Pfades wird zufällig aus der Menge der zulässigen Konfigurationen gewählt. Die Temperatur wird auf die Höchsttemperatur gesetzt, um unabhängig von dem Temperaturplan zu sein. Jeder Schritt in dem erzeugten Zufallspfad ist somit zulässig, d.h. erfüllt die Nebenbedingungen und wird durch die Akzeptanzfunktion akzeptiert. Die Perturbationsfunktion nutzt die oben genannten Regeln und es werden zunächst gleichverteilte Gewichte gewählt, um einen Bias zu vermeiden. Aus dieser Datenbasis können schließlich die Gewichte für die Regeln berechnet werden. Das Gewicht  $\alpha_j$  der  $j$ -ten Regel berechnet sich aus der Summe aller Informationsgewinne von einem Pfad zum nächsten, falls dieser positiv ist, der durch die Regel beigetragen wurde:

$$\alpha_j = \frac{\sum_{k=1}^{m_a} \delta_{i_k, j} v_k}{\sum_{k=1}^{m_a} v_k}. \quad (5.9)$$

Zum Vergleich der drei Perturbationsfunktionen: Perturbation mit zufälliger Richtung, regelbasierte Perturbation mit homogenen und optimierten Gewichten wurden Stichproben aus der besten erreichten relativen Varianz und die dafür benötigte Anzahl an Schritten mittels mehreren Optimierungsläufe erzeugt. Der Temperaturplan wurde so gewählt, dass die maximale Anzahl

an Schritten 200 beträgt. In Abbildung 5.2 sind die Stichprobenpunkte aufgetragen und ihre zugehörigen Statistiken in der Tabelle 5.1 zusammengefasst. Die Optimierung mit regelbasierter Perturbation erreicht eine im Vergleich zu der Perturbation mit zufälliger Richtung signifikant bessere Unsicherheitsreduktion. Unter den regelbasierten Perturbationsfunktionen ist die mit optimierten Gewichten im Schnitt besser als mit homogenen Gewichten.

Tabelle 5.1: Statistische Merkmale der besten relativen Varianz und die dafür benötigte Anzahl an Schritten, berechnet aus mehreren Optimierungsläufen jeweils mit der zufälligen Perturbation der Richtung, der Perturbation per Regeln mit homogenen und optimierten Gewichten.

<b>Perturbation</b>	Schritte (Median)	beste relative Varianz
zufällige Richtung	103	$0.88 \pm 0.06$
regelbasiert, homogen	156	$0.73 \pm 0.09$
regelbasiert, optimiert	163	<b><math>0.71 \pm 0.07</math></b>

### Berücksichtigung von Nebenbedingungen

Die Nebenbedingungen können zum einen berücksichtigt werden, indem die Perturbationsfunktion so gewählt wird, dass sie keine Kandidaten in unzulässigen Regionen vorschlägt. Zum anderen kann der Perturbation ein Test nachgestellt werden, der unzulässige Kandidaten identifiziert. Die Perturbationsfunktion wird solange ausgeführt bis ein Kandidat gefunden oder eine maximale Anzahl an Wiederholungen erreicht wird. Einfache Nebenbedingungen, wie Grenzbedingungen (box-constraints) können über erstere Methode berücksichtigt werden, komplexere wie Kollision zwischen Bildschirm und Oberfläche müssen explizit getestet werden. Bei Nebenbedingungen, die den Raum der zulässigen Sensorkonfigurationen in zwei oder mehrere unzusammenhängende Bereiche teilen, kann es vorkommen, dass eine Menge von Sensorkonfigurationen nicht erreicht werden kann. Dies

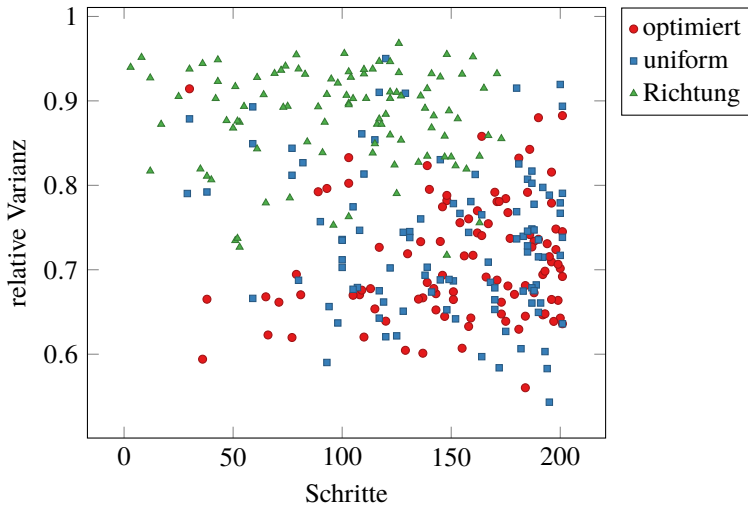


Abbildung 5.2: Stichproben der besten relativen Varianz und die dafür benötigte Anzahl an Schritten, berechnet aus mehreren Optimierungsläufen jeweils mit der zufälligen Perturbation der Richtung, der Perturbation per Regeln mit homogenen und optimierten Gewichten.

hängt nicht nur mit der Schrittgröße der Perturbation ab, sondern auch von den definierten Pertubationsrichtungen. Zur Lösung dieses Problems wird dem Simulated-Annealing-Algorithmus erlaubt in einer begrenzten Anzahl von Schritten den unzulässigen Bereich zu betreten. Wird keine zulässige Sensorkonfiguration gefunden, wird die aktuelle Position des Pfades auf die zuletzt zulässige Konfiguration zurückgesetzt.

### 5.2.2 Nelder-Mead

Das Nelder-Mead-Verfahren [Nel65], auch Downhill-Simplex-Verfahren genannt, ist eine Methode zur Optimierung von nichtlinearen Funktionen. Dazu verwendet sie nur Funktionsevaluationen also keine Gradienten. Die Konvergenz des Algorithmus ist linear. Der Algorithmus evaluiert in jedem

Schritt  $d+1$  Punkte, wobei  $d$  die Dimension des Suchraumes ist. Diese Punkte spannen ein Simplex auf. Durch Anwendung unterschiedlicher Regeln wird versucht diesen Simplex in Richtung Optimum zu transformieren. Die Hauptidee dabei ist es, den Punkt mit dem höchsten Funktionswert durch einen neuen Punkt mit niedrigerem Funktionswert zu ersetzen. Der Punkt mit dem bisherigen optimalen Funktionswert wird in jeder Iteration beibehalten. Die ursprüngliche Formulierung des Algorithmus unterstützt keine Nebenbedingungen. Die Autoren schlagen eine *Penalty Methode* vor, bei der ein Strafterm zur Zielfunktion addiert wird, der ungültige Konfigurationen bestraft. Es gibt auch verschiedene Variationen des Algorithmus, die Nebenbedingungen berücksichtigen; siehe zum Beispiel [Box65, Gui68].

## 5.3 Greedy-Planung

Bei der Greedy-Planung wird das Optimierungsproblem (5.1) durch eine Folge von Optimierungsproblemen niedriger Komplexität ersetzt:

$$J_k(b_k) = \min_{\theta_k} \rho(b_k, \theta_k), \quad k = 0, 1, \dots \quad (5.10)$$

In jedem Iterationsschritt der Planung wird nur über eine Sensorkonfiguration minimiert, in dem nur ein Schritt im Voraus geplant wird. Da die Sensorkonfigurationen iterativ berechnet werden, kann die Greedy-Planung auch online erfolgen. Dazu kann anhand der geplanten Sensorkonfiguration  $\theta_k$  eine reale Messung ausgeführt und zur Aktualisierung der Verteilung über der Oberfläche  $b_k$  genutzt werden. Dies hat den Vorteil, dass Unsicherheiten, die nicht mit modelliert wurden, kompensiert werden können. Allerdings ist dieses Vorgehen in der Praxis bei zeitrelevanten Aufgaben nicht geeignet, da der Zeitaufwand der Optimierung hoch ist. Abhilfe schafft hier, das Messsystem durch einen Simulator zu ersetzen. Dann kann die

Planung offline erfolgen. Als Resultat erhält man einen Plan, der schließlich online nur noch ausgeführt werden muss. Der Greedy-Algorithmus terminiert, wenn die maximale Varianz kleiner oder gleich der gewünschten Varianz  $U_{\text{suff}}$  ist.

### 5.3.1 Greedy-Planung mit Teildiskretisierung

Um lokale Minima zu vermeiden wird die Sensorkonfiguration teildiskretisiert [Ros12, Ros13]. Hierzu wird der Parameter  $\chi$  auf diskrete Punkte auf der Oberfläche eingeschränkt. Diese können in einem Raster in der Domäne der Oberfläche gewählt werden. Eine andere Möglichkeit ist die Domäne der Oberfläche in Regionen zu partitionieren und den Parameter  $\chi$  auf diese Regionen zu beschränken. Die Elemente der Partition repräsentieren die diskreten Aktionen. In beiden Fällen wird so eine Familie von Sensorkonfigurationenräumen definiert  $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_{|\mathcal{P}|}\}$  mit  $\mathcal{P}_i \subset \Theta$ . Die Selektion der nächsten Sensorkonfiguration erfolgt hierarchisch. Auf der äußeren Ebene wird aus der diskreten Familie  $\mathcal{P}$  eine Menge  $\mathcal{P}_i$  gewählt und aus dieser wird über eine Optimierung die Aktion mit minimalen Kosten gewählt:

$$J_k(b_0) = \min_{i=1, \dots, |\mathcal{P}|} \min_{\theta_k \in \mathcal{P}_i} \rho(b_k, \theta_k), \quad k = 0, 1, \dots, . \quad (5.11)$$

In der Implementierung dieses Verfahrens muss zunächst in der untersten Hierarchie in jeder Region unabhängig voneinander eine optimale Sensorkonfiguration gefunden werden. In der oberen Hierarchie wird aus der diskreten Menge der Sensorkonfigurationen, die mit den niedrigsten Kosten gewählt. Der Pseudocode für diese Prozedur ist in Algorithmus 2 gegeben.



**Algorithm 2** Naive Greedy-Optimierung mit Teildiskretisierung

---

```

 $b \leftarrow b_0$ 
for all  $k = 0 \dots$  bis Terminierung do
  for all Regionen  $i = 1 \dots |\mathcal{P}|$  do
     $\zeta_i^* \leftarrow \arg \min_{\theta} \rho(b, \theta)$  mit Nebenbedingungen:  $\theta \in \mathcal{P}_i$ 
  end for
   $j \leftarrow \arg \min_i \rho(b, \zeta_i^*)$ 
   $\theta_k \leftarrow \zeta_j^*$ 
   $\mathcal{Z} \leftarrow h(\mathcal{S}_r, \theta_k, \mathbf{0})$ 
   $b \leftarrow B(b, \theta_k, \mathcal{Z})$ 
end for

```

---

Der Algorithmus 2 berechnet die Sensorkonfigurationen für jede Region in jedem Zeitschritt neu. Dies erfordert einen hohen Zeitaufwand. Die Greedy-Planung kann beschleunigt werden, in dem das Wissen um die Optimierung aus den vorherigen Schritten berücksichtigt wird. So können untere Schranken für die einzelnen Kosten in den Regionen berechnet werden. Im Folgenden wird angenommen, dass die Kosten  $g_a$  für den Aufwand der Einstellung des Sensors konstant sind.

Für die Kosten einer beliebigen Sensorkonfiguration  $\theta$  gilt die Ungleichung:

$$\rho(b_{k+1}, \theta) \geq \rho(b_k, \theta). \quad (5.12)$$

Sie besagt, dass die Kosten für eine beliebige Sensorkonfiguration  $\theta$  steigen, je später sie eingesetzt wird. Dies liegt daran, dass die Sichtbereiche der vorherigen Messungen mit  $\theta$  Überschneidungen haben können und damit der erwartete Informationsgewinn sinkt. Mit dieser Ungleichung ist es möglich die Kosten des aktuellen Zeitschritts als eine untere Schranke für den nächsten Schritt zu wählen. Die Ungleichung kann nicht direkt für die diskreten Konfigurationen angewendet werden, da die optimale Konfiguration einer Region sich von einem Schritt zum nächsten ändern kann. Per Definition der

optimalen Konfiguration in einer Region  $i \in \{1, \dots, |\mathcal{P}|\}$  gilt jedoch, dass für alle  $\theta \in \mathcal{P}_i$ :

$$\rho(b_k, \theta) \geq \rho(b_k, \zeta_i^*), \quad (5.13)$$

wobei  $\zeta_i^* \in \Theta$  die optimale Konfiguration des Sensorkonfigurationsraumes  $\mathcal{P}_i$  ist. Zusammen kann schließlich gefolgert werden, dass:

$$\rho(b_{k+1}, \theta) \geq \rho(b_k, \theta) \geq \rho(b_k, \zeta_i^*) \quad (5.14)$$

für alle  $\theta \in \mathcal{P}_i$ . Durch Ausnutzen der Lokalität der Messungen kann die untere Schranke weiter verfeinert werden. Der Einflussbereich einer Messung ist der Bereich auf der geschätzten Oberfläche, der von den gemessenen Gradienten abhängt. Dieser kann größer als der Messbereich selbst sein und hängt von der Wahl des Oberflächenmodells ab. Wenn sich nun der Einflussbereich der Messung, die  $b_k$  nach  $b_{k+1}$  überführt, nicht mit dem Einflussbereich  $\zeta_i^*$  schneidet, dann ändert sich der Wert  $\rho(b_k, \zeta_i^*)$  nicht. Beim Gauß-Prozess Modell muss dabei die verwendete Kovarianzfunktion eine endliche Domäne besitzen oder zumindest lokal konzentriert sein und schnell nach außen hin abfallen. Im letzten Fall kann der Einfluss des Messbereichs auf einen endlichen Bereich approximiert werden.

Durch Zuhilfenahme der Ungleichung (5.14) und der Lokalität der Messung kann der Naive Greedy-Algorithmus modifiziert werden, um den Berechnungsaufwand zu reduzieren. Zunächst werden in einer Initialisierungsphase optimale Sensorkonfigurationen für alle Regionen berechnet. Der Algorithmus speichert die Kosten in einer Prioritätswarteschlange. Es wird zwischen zwei Arten von Werten unterschieden: exakte Werte berechnet durch  $\rho$  und untere Schranken. In jedem Zeitschritt wird aus der Prioritätswarteschlange die Sensorkonfiguration mit dem niedrigsten Wert selektiert. Wenn es sich um eine untere Schranke handelt, wird der exakte Wert berechnet und der aktualisierte Wert neu in die Prioritätsschlange eingefügt. Dieser Vorgang wird wiederholt bis ein exakter Wert gewählt wird. Die zugehörige Sensor-

konfiguration wird als die optimale in dem aktuellen Schritt gewählt. Alle Sensorkonfigurationen in der Prioritätsschlange, die den Einflussbereich der gewählten Sensorkonfiguration schneiden, werden als untere Schranken markiert. Ein Pseudocode ist in Algorithmus 3 gegeben. Es sei angemerkt, dass die Ungleichung (5.13) in der Praxis nicht gewährleistet werden kann, da durch Optimierungsalgorithmen nicht immer das globale Optimum gefunden wird. In diesem Fall gilt diese Ungleichung nicht und das Verfahren kann nur eine approximative Lösung finden.

---

**Algorithm 3** Schnelle Greedy-Planung
 

---

```

for all  $i = 0 \dots |\mathcal{P}|$  do
     $g_i \leftarrow 0$  ▷ Kosten für Partition  $i$ 
     $s_i \leftarrow \text{True}$  ▷ Kosten der Partition  $i$  sind eine untere Schranke
     $\zeta_i \leftarrow \mathbf{0}$  ▷ Beste Sensorkonfiguration für Partition  $i$ 
end for
 $b \leftarrow b_0$ 

for all  $k = 0 \dots$  bis Terminierung do
     $j \leftarrow \arg \min_i g_i$ 
    while  $s_j = \text{True}$  do
         $\zeta_j \leftarrow \arg \min_{\theta} \rho(b, \theta)$  Nebenbedingungen:  $\theta \in \mathcal{P}_i$ 
         $g_j \leftarrow \rho(b, \zeta_j)$ 
         $s_j \leftarrow \text{False}$ 
         $j \leftarrow \arg \min_i g_i$ 
    end while

     $\theta_k \leftarrow \zeta_j$ 
    for all  $i = 0 \dots |\mathcal{P}|$  do
        if  $\zeta_i$  schneidet Einflussbereich von  $\theta_k$  then
             $s_i \leftarrow \text{True}$ 
        end if
    end for
     $\mathcal{Z} \leftarrow h(\mathcal{S}_r, \theta_k, \mathbf{0})$ 
     $b \leftarrow B(b, \theta_k, \mathcal{Z})$ 
end for

```

---

In einem Experiment mit 36 Regionen wurde anhand des beschleunigten Greedy-Algorithmus ein Plan mit 16 Messungen berechnet. Die Regionen wurden erzeugt, indem die Sichtpunktkomponente auf einem Gitter von  $6 \times 6$  beschränkt wurde. Abbildung 5.3a zeigt die Anzahl der Optimierungen für die Regionen und Abbildung 5.3b das dazu korrespondierende Histogramm über die Anzahl der Optimierungen. Die maximale Anzahl an Optimierungen für eine Region liegt bei fünf. Im Vergleich dazu optimiert der Naive Greedy Algorithmus in jedem Schritt die Konfigurationen für alle Regionen neu und benötigt 16 Optimierungen pro Region. Die totale Aufwandsreduzierung liegt bei einem Faktor von 4.1.

## 5.4 Sweep-Planung

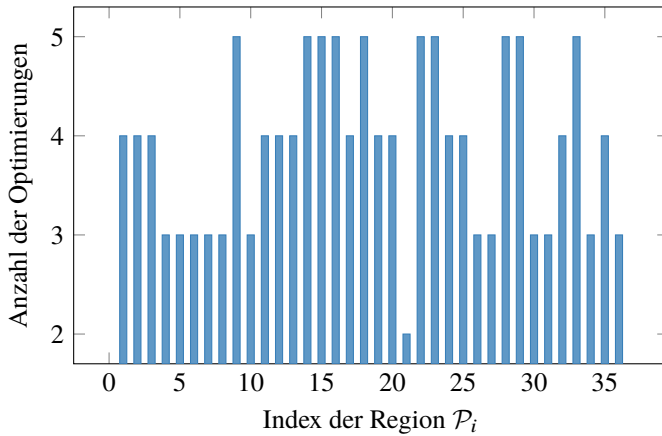
Eine Hürde bei der Minimierung des Optimierungsproblems (5.1) ist die unbekannte Anzahl der Messungen. Das Optimierungsproblem (5.1) kann aufgespalten werden zu:

$$J(b_0) = \min_n J_n(b_0), \quad (5.15)$$

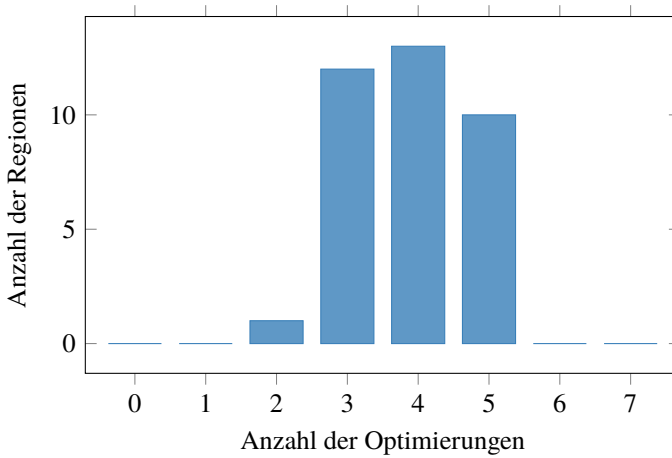
$$J_n(b_0) = \min_{\theta_{0:n-1}} \sum_{i=0}^{n-1} \rho(b_i, \theta_i) + \rho_E(b_n). \quad (5.16)$$

Das Optimierungsproblem (5.16) wird für feste Planlängen  $n$  gelöst und in Gleichung (5.15) wird die optimale Anzahl  $n^*$  gesucht.

Da die Terminalkosten im Optimierungsproblem (5.16) nur sicherstellen, dass der optimale Plan die Unsicherheit ausreichend reduziert und für die tatsächliche Minimierung hinderlich sind, werden sie aus der Zielfunktion herausgenommen und direkt als Endkriterium der Optimierung definiert. Für ein festes  $n$  gilt die Optimierung (5.16) als erfolgreich, falls die maximale Varianz des Oberflächenmodells  $b_n$  die geforderte Unsicherheit  $U_{\text{suff}}$  erreicht



(a) Varianz



(b) Histogramm

Abbildung 5.3: Aufwand der Optimierung: (a) zeigt die Anzahl der Optimierungen für die einzelnen Regionen und (b) das korrespondierende Histogramm über die Anzahl der Optimierungen.

oder unterschreitet. Die Optimierung schlägt fehl, wenn kein Plan gefunden wird, der dieses Endkriterium erfüllt. Dies wird über die üblichen Kriterien wie die Überschreitung einer maximalen Anzahl an Iterationen oder eine zu langsame Konvergenz festgestellt. In der Praxis lässt sich die Existenz eines Plans schneller feststellen als seine Nichtexistenz, da die Abbruchkriterien für das Abbrechen der Optimierung konservativ gewählt werden, um keine Fehlentscheidung zu treffen.

Die Suche kann auf mehrere Arten erfolgen. Bei einer linearen Suche wird  $n$  mit einer maximalen oder minimalen Anzahl an Messungen initialisiert. Eine maximale Anzahl an Messungen  $n_{\max}$  kann über die Greedy-Lösung berechnet werden. Eine Abschätzung einer minimalen Anzahl an benötigten Messungen kann über den Informationsgewinn der optimalen ersten Greedy-Messung erfolgen. Dividiert man ihren Wert durch die gesamte zu reduzierende Unsicherheit ergibt sich die Abschätzung. Der Aufwand für die lineare Suche ist  $\mathcal{O}(n_{\max})$ . Bei der binären Suche wird zwischen der minimalen und der maximalen Anzahl an benötigten Messungen gesucht. Der Aufwand für die binäre Suche beträgt  $\mathcal{O}((n_{\max} - n_{\min}) \log(n_{\max} - n_{\min}))$ .

In dieser Arbeit wird eine Suche mit einer festen Schrittweite  $m$  genutzt. Dabei wird von  $\lfloor \frac{n_{\max}}{m} \rfloor m, \dots, 3m, 2m, m$  absteigend in der festen Schrittweite  $m$  gesucht. Wird für die Planlänge  $k \cdot m$  ein Plan gefunden und für  $(k + 1) \cdot m$  kein Plan, so kann der Suchbereich in der Region zwischen diesen Planlängen eingeschränkt werden. In der Region wird linear weitergesucht. Der Vorteil dieser Suche ist, dass sie parallel ausgeführt werden kann. Dabei wird  $m$  kleiner gleich der Anzahl der möglichen Threads gewählt. Der Berechnungsaufwand beträgt  $\mathcal{O}(n_{\max}/m + m)$  im sequentiellen Fall und  $\mathcal{O}((n_{\max}/m + m)/m)$  im parallelen Fall.

Die lineare Suche von  $n_{\max}$  aus stößt nur einmal auf eine Planlänge, die keine Lösung hat. Die Suche mit fester Schrittweite nur zweimal. Bei der

binären Suche kann im Worst-Case jede betrachtete Planlänge bis auf das berechnete Optimum fehlschlagen.

Das Optimierungsproblem (5.16) für feste  $n$  hat eine Dimension von  $nd$ . Die direkte Optimierung mittels Simulated-Annealing kann erfolgen, indem in der Pertubationsfunktion eine Konfiguration zufällig ausgewählt und perturbiert wird. Der Aufwand für die Evaluation steigt jedoch mit wachsendem  $n$ . Um den Aufwand zu reduzieren, ist es sinnvoll, ausgewählte Konfigurationen über mehrere Schritte zu optimieren und die restlichen Konfigurationen konstant zu halten. Dies wird erreicht, indem der Optimierungsalgorithmus in mehreren Durchgängen die Zielfunktion minimiert; siehe Algorithmus 4. In jedem Durchgang wird eine Aktion gewählt und die Zielfunktion bezüglich dieser minimiert. Um lokale Minima zu vermeiden, wird hier nach einer fest vorgegebenen Anzahl an Schritten gestoppt. Die Wahl der Sensorkonfiguration für einen Durchgang kann zufällig geschehen oder die Durchläufe werden in Zyklen zusammengefasst, in denen jede Sensorkonfiguration einmal ausgewählt wird. Es werden so viele Durchgänge durchgeführt, bis die Terminierungsbedingung erfüllt ist [Ros14].

---

**Algorithm 4** Optimierung anhand von Durchgängen

---

```

for all Zyklen  $i = 0 \dots$  bis Terminierung do
  for all Messungen  $j = 0 \dots n$  do
     $b \leftarrow b_0$ 
    for all Messungen  $k \neq j$  do
       $\mathcal{Z} \leftarrow h(S_r, \theta_k, \mathbf{0})$ 
       $b \leftarrow B(b, \theta_k, \mathcal{Z})$ 
    end for
     $\theta_j \leftarrow \arg \min_{\theta} \rho(b, \theta)$ 
  end for
end for

```

---

## 5.5 Zusammenfassung

In diesem Kapitel wurden zwei Planungsverfahren, die Greedy-Planung und die Sweep-Planung für die Oberflächeninspektion vorgeschlagen. Die Planung wird als Minimierungsproblem über die Kosten, die aus der Unsicherheit und den Aktionskosten gemäß dem in Kapitel 2 vorgestellten Rahmenwerks bestehen, formuliert. Da das Optimierungsproblem eine hohe Dimension aufweist, wurden Verfahren vorgestellt, die intern einen Optimierer für jeweils eine Konfiguration nutzen.

Als Optimierungsverfahren wird das Simulated-Annealing-Verfahren eingesetzt. Die Pertubationsfunktion des Simulated-Annealing Verfahrens wird modular modelliert, so dass unterschiedliches Vorwissen, hauptsächlich in Form von Schrittrichtungen, eingeführt werden kann. Das Nelder-Mead-Verfahren wird als ein Nachoptimierer eingesetzt, um das Ergebnis weiter zu verfeinern.

Bei der Greedy-Planung wird das Optimierungsproblem in eine Sequenz von einfacheren Optimierungsproblemen zerlegt, die jeweils nur eine Schritt im Voraus planen. Dadurch sinkt pro Optimierung die Dimension der Zielgröße auf die einer Sensorkonfiguration. Um dem Problem der lokalen Minima zu entgehen, wird eine Teildiskretisierung der Sensorkonfiguration vorgeschlagen und hierfür eine effiziente Planung unter Ausnutzung einer Abschätzung einer oberen Schranke für die Unsicherheitsreduktion des nächsten Messschritts verwendet.

Die Sweep-Planung nutzt einen hierarchischen Ansatz, bei dem auf der höchsten Ebene über die Planlängen gesucht wird und auf der untersten Ebene die Optimierung für feste Planlängen erfolgt. Zur Reduzierung des Aufwandes wird die Optimierung in mehreren Durchläufen ausgeführt. Bei jedem Durchlauf werden alle Sensorkonfigurationen jeweils einzeln ausgewählt und für eine feste Anzahl von Schritten optimiert. Dabei werden die



restlichen Sensorkonfigurationen konstant gehalten. Die Suche der optimalen Planlänge kann auf unterschiedlichen Arten erfolgen. In dieser Arbeit wird eine Suche mit fester Schrittweite bevorzugt, die zunächst die möglichen Planlängen auf ein Intervall der Länge der Schrittweite einschränkt und anschließend im Intervall sucht. Der Vorteil dieser Suche ist, dass sie parallel erfolgen kann.



## **6 Evaluation der Planungsverfahren**

In diesem Abschnitt wird die Güte der Planungsverfahren anhand mehrerer Experimente evaluiert. Zunächst werden die Verfahren mit simulierten Ergebnissen evaluiert. In Abschnitt 6.1 wird der dafür verwendete Messaufbau beschrieben. Anschließend wird in Abschnitt 6.2 die Unsicherheitsmaße des Gauß-Prozess-Modells verglichen. In Abschnitt 6.3 wird die Planung mit Gauß-Prozess Modell und in Abschnitt 6.4 mit diskreten Modell evaluiert. In Abschnitt 6.5 wird die Wirkung der Approximation auf die Varianzreduktion und der Planung untersucht. Die Berechnung und Ausführung eines Plans für eine reale Oberfläche erfolgt abschließend in Abschnitt 6.6.

### **6.1 Testaufbau der simulationsbasierten Experimente**

#### **Sensor**

Bei den im Folgenden simulierten Experimenten wird ein Sensor bestehend aus einer Kamera und einem Bildschirm gewählt, wobei beide Komponenten unabhängig voneinander positioniert werden können. Die Dimension der Sensorkonfiguration beträgt damit zwölf. Es wird ein Bildschirm mit einer Bildschirmdiagonalen von 26 Zoll und einer Auflösung von  $1920 \times 1080$  verwendet. Die Kamera wird mit einer Sensorgröße von  $2.4 \text{ mm} \times 1.8 \text{ mm}$

und einer Brennweite von 5 mm simuliert. Die Bildweite wird gleich der Brennweite gesetzt, um einen möglichst hohen Schärfentiefe zu erhalten. Die Auflösung der Kamera wird abhängig vom Oberflächenmodell gewählt. Für das diskrete Modell wird eine Auflösung von  $1024 \times 768$  gewählt. Da das Gauß-Prozess Modell aufgrund der Approximation eine niedrige Ortsauflösung besitzt, wird hierfür eine kleinere Auflösung von  $800 \times 600$  gewählt. Die Standardabweichung der Registrierung auf dem Bildschirm wird konstant auf 0.5 mm gesetzt.

Die Parameter wurden so gewählt, dass eine sinnvolle Evaluation möglich ist, das heißt, dass eine Inspektion mehrere Messungen erfordert und dass einzelne Messungen genügend Information liefert. Sind die Unsicherheiten zu hoch, werden mehrere sich überlagernde Messungen geplant, was in der Praxis unerwünscht ist. In diesem Fall ist das Messsystem für die Inspektionsaufgabe nicht geeignet.

### Testoberflächen

Als Testoberfläche werden folgende Oberflächen verwendet; siehe auch Abbildung 6.1 und 6.2:

- Ein planarer Spiegel  $f_p(x, y) = h$  positioniert auf einer Höhe  $h$ .
- Ein konvexer Spiegel  $f_s(x, y) = \sqrt{r^2 - (x - x_c)^2 - (y - y_c)^2} + z_c$  mit dem Radius  $r$  und dem Mittelpunkt  $(x_c, y_c, z_c)$ .
- Ein konkaver Spiegel  $f_i(x, y) = -\sqrt{r^2 - (x - x_c)^2 - (y - y_c)^2} + z_c$ , der durch einen Parabolspiegel repräsentiert wird mit dem Radius  $r$  und dem Mittelpunkt  $(x_c, y_c, z_c)$ .

Für das Zentrum des konvexen und sowohl des konkaven Spiegels wird der Punkt  $(x_c, y_c, z_c) = (0, 0, 0)$  und für den Radius  $r = 500$  mm gewählt, wenn nichts anderes angegeben. Die Höhe des planaren Spiegels wird auf  $h = 0$  gesetzt. Die Oberflächen sind jeweils in einem Bereich von

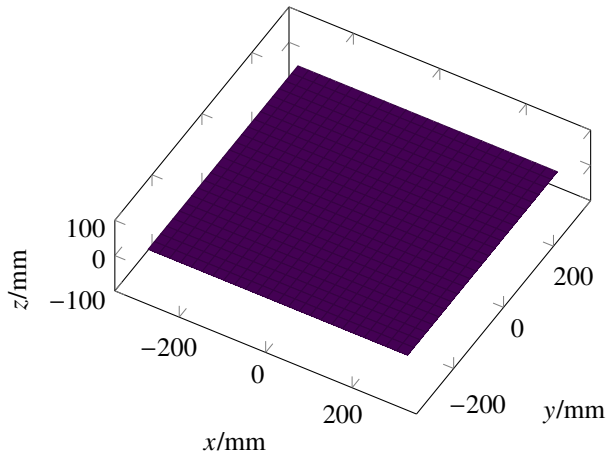


Abbildung 6.1: Ebene Testoberfläche für die simulationsbasierte Evaluation der Planungsalgorithmen

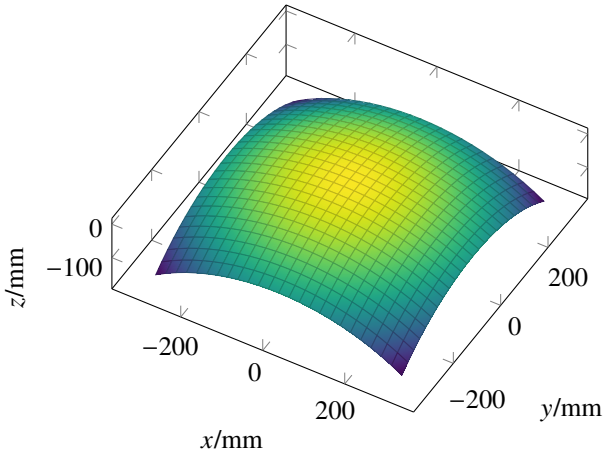
600 mm × 600 mm definiert. Diese Oberflächen werden als Referenzoberflächen während der Planung genutzt.

### Fehlermodelle

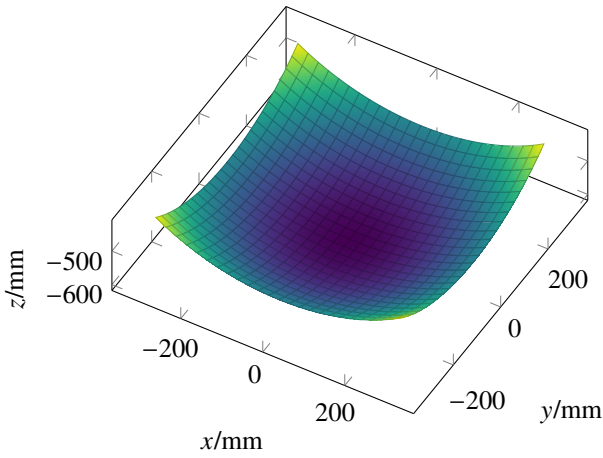
Zur Evaluation der Qualität der berechneten Pläne werden Testoberflächen mittels einer simulierten Inspektion untersucht. Die Testoberflächen werden aus der Referenzoberfläche mittels additiver Überlagerung eines Defektmodells  $f(\mathbf{x}) = f_r(\mathbf{x}) + f_e(\mathbf{x})$  gebildet. Der Defekt  $f_e$  wird als eine Gauß-Funktion modelliert:

$$f_e(x, y) = H_d e^{-\frac{(\|\mathbf{x} - \mathbf{x}_0\|)^2}{2L_d^2}}. \quad (6.1)$$

Anhand des Parameters  $\mathbf{x}_0$  lässt sich die Position des Fehlers setzen, die Parameter  $L_d$  und  $H_d$  beschreiben jeweils die Ausdehnung und die Höhe des Defektes. Mit negativem  $H_d$  werden Dellen und mit positiven Beulen



(a) Konvexer Spiegel



(b) Konkaver Spiegel

Abbildung 6.2: Testoberflächen für die simulationsbasierte Evaluation der Planungsalgorithmen

modelliert. Die maximalen Gradienten liegen auf dem Kreis  $\|\mathbf{x} - \mathbf{x}_0\| = L_d$  und der zugehörige Neigungswinkel ist

$$\arctan\left(\frac{H_d}{L_d\sqrt{e}}\right). \quad (6.2)$$

Insbesondere haben zwei Gauß-Funktionen den gleichen maximalen Neigungswinkel, wenn das Verhältnis von Höhe zu Defektbreite  $\frac{H_d}{L_d}$  übereinstimmt. Das erlaubt Defekte unterschiedlicher Größe aber mit gleichem maximalen Neigungswinkel zu modellieren.

### 6.1.1 Durchführung der simulationsbasierten Inspektion

Zur Detektion der Defekte auf den Testoberflächen wird das gleiche Modell verwendet, das für die Planung genutzt wird. Dies wird hier zur Evaluation der Verfahren genutzt. Generell ist es jedoch nicht notwendig die Detektion der Defekte auf den Modellen zu basieren. Beispielsweise kann eine Detektion gänzlich auf den deflektometrischen Registrierungen basieren. Im Detail werden die Messungen gemäß dem optimalen Plan und der spezifizierten Unsicherheit simuliert und die so gewonnenen Messdaten in das jeweilige Modell fusioniert.

Zur Prüfung wird die Abweichung mittels absoluter Differenz vom Erwartungswert der geschätzten Modelle und dem Referenzmodell berechnet. Beim Gauß-Prozess Modell wird die Höhenabweichung und beim diskreten Modell die Normalenabweichung zum Referenzmodell geschätzt. In beiden Fällen werden anschließend anhand eines Schwellenwertes Bereiche mit hoher Abweichung identifiziert und mittels eines Blob-Detektors die Zentren bestimmt. Durch Vergleich mit den wahren Defektpositionen wird die Trefferquote (engl. recall) und die Genauigkeit (engl. precision) bestimmt. Die Trefferquote ist definiert als das Verhältnis von der Anzahl der korrekt erkannten Defekte zu der Anzahl der Defekte. Die Genauigkeit ist definiert

als das Verhältnis von der Anzahl der korrekt erkannten Defekte zu der Anzahl der erkannten Defekte (unabhängig davon ob diese korrekt sind oder nicht). Da der Schwellenwert direkt auf den Abweichungen angewendet wird, geben Trefferquote und Genauigkeit Aufschluss über die Güte des Plans.<sup>1</sup>

## 6.2 Gütemaß des Gauß-Prozess Modells

Für das Gauß-Prozess Modell wurden im Abschnitt 4.3.8 vier Gütemaße vorgestellt: Die totale Varianz  $U_T$ , die Determinante des diskretisierten Gauß-Prozesses  $U_D$ , der maximale Eigenwert  $U_E$  und die maximale Varianz  $U_M$ . In diesem Abschnitt wird untersucht, welches Gütemaß zur Planung geeignet ist. Die maximale Varianz wird in den vorgestellten Planungsverfahren in der Terminierungsbedingung genutzt. Eine direkte Minimierung dieser ist allerdings nicht geeignet, da diese Größe über weite Bereiche des Konfigurationsraumes konstant ist. Aus diesem Grund wird sie in der folgenden Untersuchung nicht aufgeführt. Mit dem Gauß-Prozess-Modell wird anhand der Greedy-Planung ein Plan für die konvexe Oberfläche berechnet. Für den Radius wird  $r = 1400$  mm gewählt. Zur approximativen Inferenz werden  $25 \times 25$  induzierte Punkte in einem Gitter gewählt. Der Plan wird anschließend in einer Simulation ausgeführt und anhand der simulierten Messwerte die Oberfläche rekonstruiert. Da der Gauß-Prozess mit Neigungsbeobachtungen die Oberfläche bis auf einen konstanten Höhenversatz schätzt, wurde hier als Fehlermaß der minimale quadratische Fehler über der Höhe

$$E(\hat{f}, f) = \min_c \int (\hat{f}(x) + c - f(x))^2 dx \quad (6.3)$$

---

<sup>1</sup> Die Trefferquote ist, wie vielleicht erwartet, keine bezüglich eines fallenden Schwellenwerts monoton steigende Funktion, da der Schwellenwert am Anfang des Detektionsalgorithmus auf den Abweichungen angewendet wird und beispielsweise bei niedrigem Schwellenwert zu große Fehlerbereiche entstehen, deren Zentren falsch geschätzt werden.



gewählt. Dabei ist  $\hat{f}$  die Schätzung der Oberfläche und  $f$  die wahre Oberfläche. Die benötigte Anzahl von Messungen, um den obigen Fehler des Erwartungswertes unter einen Schwellenwert zu reduzieren, dient als Indiz für die Qualität des Gütemaßes. In Abbildung 6.3 wird der durchschnittliche Fehler  $E$  in Abhängigkeit der Anzahl der Messungen dargestellt. Im Vergleich schneidet hier das Maß  $U_T$  am besten ab. Eine ähnliche Performanz bietet das Maß  $U_D$ . Beide reduzieren den Fehler auf circa um den Faktor 1/10 innerhalb der ersten zehn Messungen. Der maximalen Eigenwert benötigt dafür dreizehn Messungen.

Im Vergleich zum Maß  $U_D$  bietet das Maß  $U_T$  den Vorteil, dass es eine anschauliche physikalische Interpretation, als Standardabweichung der Oberflächenhöhe besitzt. Das vereinfacht die Wahl der Parameter wie z.B. den Schwellenwert für die gewünschte maximale Varianz. Außerdem lässt sie sich effizient berechnen. Ein Nachteil der Minimierung des  $U_T$  ist, dass sie keine theoretische Aussage bezüglich der oberen Schranke der maximalen Varianz zulässt. In den folgenden Experimenten wird stets das Unsicherheitsmaß  $U_T$  für das Gauß-Prozess Modell verwendet.

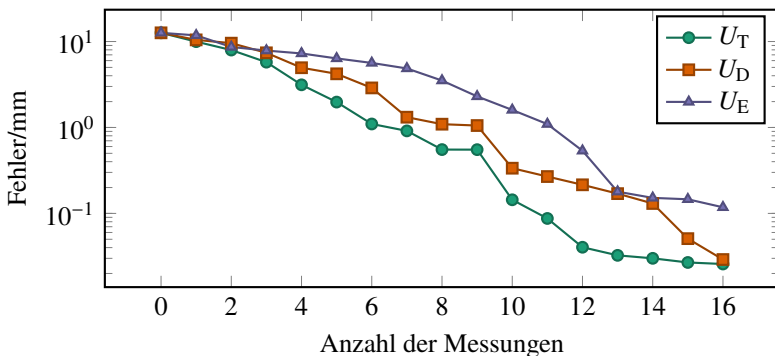


Abbildung 6.3: Durchschnittlicher Fehler für die Planung mittels Gauß-Prozess Modell bei Variation der Gütemaße.

### 6.3 Planungsverfahren anhand des Gauß-Prozess Modells

In diesem Abschnitt werden die Planungsalgorithmen mit dem Gauß-Prozess als Oberflächenmodell anhand von simulierten Experimenten evaluiert. Für das Gauß-Prozess-Modell wird für die Kovarianzfunktion der kubische B-Spline-Kern aus Abschnitt 4.10 verwendet. Zur approximativen Inferenz werden  $25 \times 25$  induzierte Punkte in einem Gitter gewählt. Die gewünschte maximale Unsicherheit wird auf  $U_{\text{suff}} = 0.1^2 \text{ mm}^2$  und die A-priori Unsicherheit auf  $\sigma_A^2 = 0.1^2 \text{ mm}^2$  gesetzt. Das Sweep-Verfahren nutzt die Suche mit fester Schrittweite mit  $m = 5$ . Zur Berechnung der Pläne wurde ein Rechner mit 16 GB RAM und zwei Intel Xeon E5645 Prozessoren mit einem Prozessortakt von 2.4 Ghz und je 6 Kernen verwendet.

In Abbildung 6.4 ist die relative durchschnittliche Varianz der geplanten Messung mittels Greedy-Planung in Abhängigkeit des Zeitschritts aufgetragen. Mit den ersten geplanten Messungen reduziert der Greedy-Algorithmus einen großen Anteil der Unsicherheit über der Oberfläche. Die Reduktion fällt mit steigender Anzahl an Messungen stark ab. Der Grund hierfür liegt darin, dass die verbleibende Unsicherheit in mehreren Bereichen örtlich auf der Oberfläche verteilt ist und dementsprechend viele Messungen benötigt werden, um diese abzudecken.

Die Ergebnisse der Sweep-Planung für die gefundene optimale Anzahl von Messungen sind in Abbildung 6.5 gegeben. Im Vergleich zum Greedy-Algorithmus erreicht der Sweep-Algorithmus ungefähr  $1.5 \times -2 \times$  kürzere Pläne. Auffallend bei beiden Planungsverfahren ist, dass die geplanten Messbereiche sich nicht überlappen. Dies liegt an der niedrigen Oberflächenauflösung und der damit einhergehenden Glattheitsannahme. Eine Zusammenfassung der Ergebnisse ist in der Tabelle 6.1 gegeben. Für die Sweep-Planung ist die Gesamtzeit der Suche über alle betrachteten Planlängen angegeben.

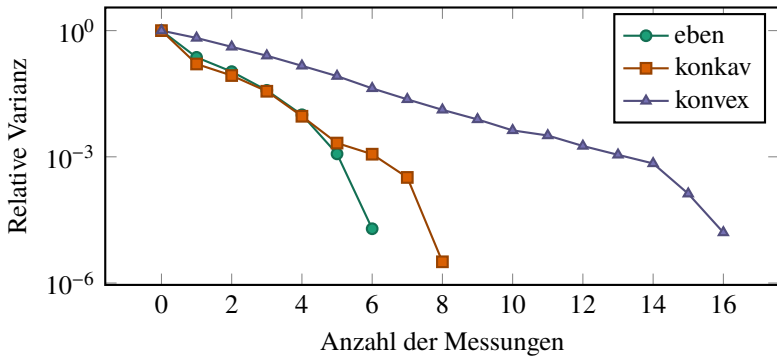


Abbildung 6.4: Verlauf der relativen Varianz der mittels Greedy-Planung und Gauß-Prozess Oberflächenmodell gefundenen Pläne für die Testoberflächen.

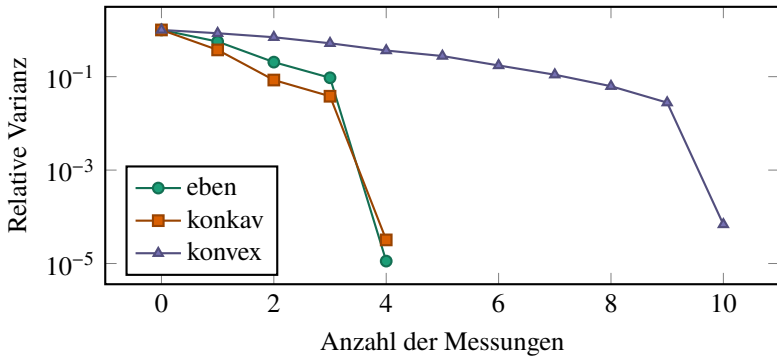


Abbildung 6.5: Verlauf der relativen Varianz der mittels Sweep-Planung und Gauß-Prozess Oberflächenmodell gefundenen Pläne für die Testoberflächen.

Hier wird die Zeit sowohl für die Suche in parallel und sequentiell (Summe über alle Planungen für die betrachteten Planlängen) angegeben.

Tabelle 6.1: Zusammenfassung der Ergebnisse der Greedy-Planung und der Sweep-Planung mit Gauß-Prozess Modell für die jeweiligen Testoberflächen. Aufgelistet sind das Planungsverfahren (PV), die Oberfläche, die optimale gefundene Planlänge  $n^*$ , die erreichte maximale Standardabweichung  $\sigma_{\max}$  sowie die totale Zeit für die Planung mit Suche im sequentiellen  $t_{\text{seq}}$  und parallelen  $t_{\text{par}}$  Fall. Zur Berechnung der Pläne wurde ein Rechner mit 16 GB RAM und zwei Intel Xeon E5645 Prozessoren mit einem Prozessortakt von 2.4 Ghz und je 6 Kernen verwendet.

PV	Oberfläche	$n^*$	$\sigma_{\max}/\text{mm}$	$t_{\text{seq}}/\text{h}$	$t_{\text{par}}/\text{h}$
greedy	eben	6	0.09	8.40	-
greedy	konvex	16	0.05	30.38	-
greedy	konkav	8	0.04	12.03	-
sweep	eben	4	0.07	72.10	46.30
sweep	konvex	10	0.08	165.93	87.92
sweep	konkav	4	0.10	78.77	70.97

### Simulierte Inspektion

Auf mehreren Testoberflächen wurden ein bis zwei additive Defekte mit zufälliger Höhe und konstanter Ausdehnung eingefügt und mittels des Verfahrens in 6.1.1 simulationsbasiert inspiziert. Aufgrund der groben Approximation eignet sich das Modell nur zur Inspektion von Defekten mit mittlerer bis globaler Ausdehnung. Bei einer Domänenausdehnung von 600 mm und 25 induzierenden Punkten pro Gitterrichtung beträgt der Gitterabstand circa 25 mm, wobei die induzierenden Punkte in einem größeren Bereich als die Domäne definiert sind. Der maximale Abstand der Punkte bei einer 4-er Nachbarschaft beträgt  $\sqrt{2} \cdot 25$  mm. Bei einem geforderten Approximationsfehler von  $10^{-3}$  mm wird gemäß der Abbildung 4.6 eine Defektausdehnung von 50 mm gewählt. Die Ergebnisse der simulierten Inspektion sind in Tabelle 6.2 aufgelistet. Für alle drei Oberflächen führt ein Schwellenwert von 0.025 mm zu einer hohen Trefferquote mit einer Genauigkeit von 1.0.

Tabelle 6.2: Trefferquote (T) und Genauigkeit (G) für die Inspektion der unterschiedlichen Testoberflächen in Abhängigkeit des Schwellenwertes.

Oberfläche		Schwellenwert/mm						
		0.0032	0.0065	0.0125	0.025	0.05	0.1	0.2
Eben	T	0.68	1.0	1.0	0.9	0.68	0.13	0.0
	G	0.02	0.10	0.69	1.0	1.0	1.0	1.0
Konkav	T	0.72	0.96	0.98	0.96	0.68	0.08	0.0
	G	0.03	0.10	0.72	1.0	1.0	1.0	1.0
Konvex	T	0.76	0.82	0.92	0.96	0.72	0.16	0.0
	G	0.21	0.35	0.92	1.0	1.0	1.0	1.0

## 6.4 Planungsverfahren anhand des diskreten Modells

In diesem Abschnitt werden die Planungsverfahren mit dem diskreten Modell als Oberflächenmodell anhand von simulierten Experimenten evaluiert. Die Oberflächen werden anhand des Verfahrens in Unterabschnitt 4.4.1 diskretisiert. Hierfür wird für den Flächenschwellenwert  $1.5^2 \text{ mm}^2$  und für den Schwellenwert der Normalenabweichung  $0.1^\circ$  gewählt. Für die A-priori Unsicherheit wird  $\sigma_A^2 = \tan(5^\circ)^2$  gewählt. Die gewünschte maximale Unsicherheit ist  $U_{\text{suff}} = \tan(0.1^\circ)^2$ . Das Sweep-Verfahren nutzt die Suche mit fester Schrittweite mit  $m = 5$ . Zur Berechnung der Pläne wurde derselbe Rechner wie in Abschnitt 6.3 verwendet.

In Abbildung 6.6 ist die relative durchschnittliche Varianz in Abhängigkeit der Anzahl der Messungen für die mittels Greedy-Planung berechneten Pläne dargestellt. Wie aus den in Abschnitt 6.3 genannten Gründen fällt auch hier die Unsicherheitsreduktion mit steigender Anzahl der Messungen stark ab. Die Ergebnisse der Sweep-Planung sind in Abbildung 6.7 dargestellt.

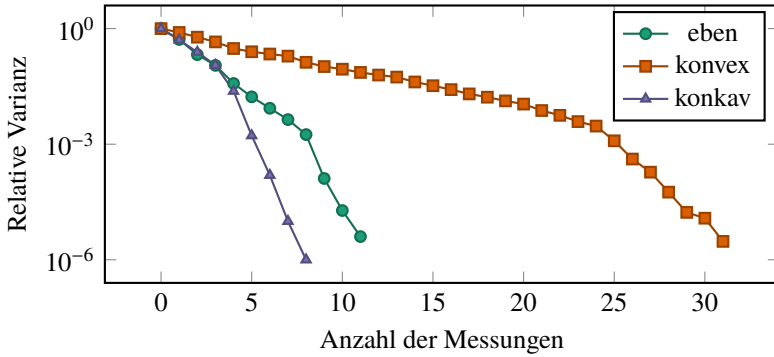


Abbildung 6.6: Verlauf der relativen Varianz der mittels Greedy-Planung und diskretem Modell gefundenen Pläne für die Testoberflächen.

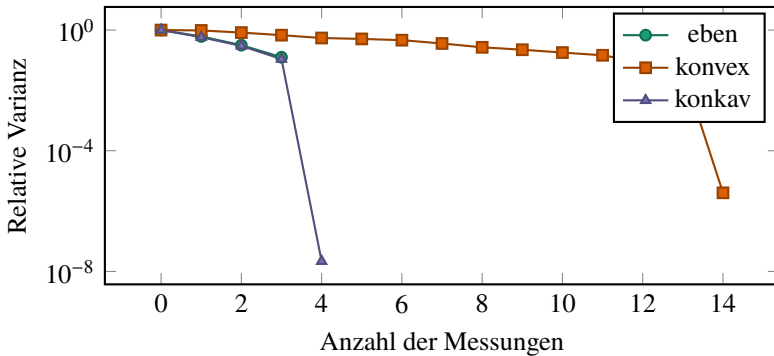


Abbildung 6.7: Verlauf der relativen Varianz der mittels Sweep-Planung und diskretem Modell gefundenen Pläne für die Testoberflächen.

Der Sweep Algorithmus berechnet im Vergleich zum Greedy-Algorithmus  $2 \times -2.75 \times$  kürzere Pläne. In der Tabelle 6.3 ist eine Zusammenfassung der interessanten Größen gegeben. Wie im vorherigen Abschnitt 6.3 ist auch hier für die Sweep-Planung die Gesamtzeit der Suche über die Planlängen für den parallelen und sequentiellen Fall angegeben.

Tabelle 6.3: Zusammenfassung der Ergebnisse der Greedy- und Sweep-Planung mit dem diskreten Modell für die jeweiligen Testoberflächen. Aufgelistet sind das Planungsverfahren (PV), die Oberfläche, die optimale gefundene Planlänge  $n^*$ , die erreichte maximale Standardabweichung  $\sigma_{\max}$  sowie die totale Zeit für die Planung mit Suche im sequentiellen  $t_{\text{seq}}$  und parallelen  $t_{\text{par}}$  Fall. Zur Berechnung der Pläne wurde ein Rechner mit 16 GB RAM und zwei Intel Xeon E5645 Prozessoren mit einem Prozessortakt von 2.4 Ghz und je 6 Kernen verwendet.

PV	Oberfläche	$n^*$	$\sigma_{\max}/^\circ$	$t_{\text{par}}/\text{h}$	$t_{\text{seq}}/\text{h}$
greedy	eben	11	0.095	2.43	-
greedy	konvex	31	0.084	5.83	-
greedy	konkav	8	0.084	1.92	-
sweep	eben	4	0.097	21.30	31.20
sweep	konvex	14	0.092	136.60	322.63
sweep	konkav	4	0.067	45.15	74.33

### Simulierte Inspektion

In diesem Unterabschnitt wird die Qualität des Plans hinsichtlich der Detektionsgüte evaluiert. Hierfür wurden wie in Unterabschnitt 6.1.1 dargelegt Testoberflächen mit additiv überlagerten Defekten betrachtet. Es werden zwei unterschiedliche Defektarten untersucht. Zum einen wird die Defektausdehnung variiert bei konstanter maximaler Neigung und zum anderen wird die maximale Neigung variiert bei konstanter Defektausdehnung. Bei Defekten mit konstanter maximaler Neigung wird die Ausdehnung im Intervall von 0.03 mm bis 1.0 mm variiert und bei Defekten mit konstanter Defektausdehnung wird die maximale Neigung im Intervall  $0.116^\circ$  bis  $1.4^\circ$  variiert. Für die konstante maximale Neigung wird  $20^\circ$  und für die konstante Defektausdehnung 3 mm gewählt. Eine Auswahl der Defekte mit beispielhafter Schwellenwertanwendung ist in Abbildung 6.8 und 6.9 dargestellt.

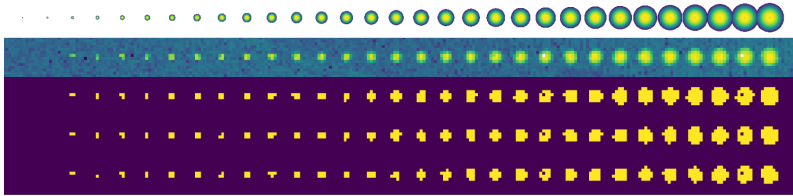


Abbildung 6.8: Schematische Darstellung der Defekte bei variierender Ausdehnung (von 0.03 mm bis 1 mm in 0.03 mm Schritten) und konstanter maximaler Neigung ( $20^\circ$ ). Von oben nach unten: Wahre Neigungsabweichung des Defekts in logarithmischer Skala, geschätzte Neigungsabweichung in logarithmischer Skala, Anwendung eines Schwellenwertfilters mit jeweils  $0.1^\circ$ ,  $0.2^\circ$  und  $0.3^\circ$ .

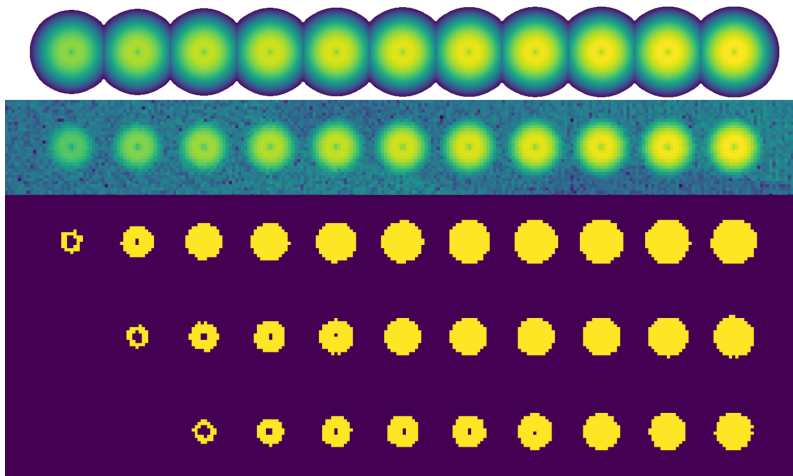


Abbildung 6.9: Schematische Darstellung der Defekte bei variierender maximaler Neigungsabweichung (von  $0.116^\circ$  bis  $1.4^\circ$  in  $0.116^\circ$  Schritten) und konstanter Ausdehnung von 3 mm. Von oben nach unten: Wahre Neigungsabweichung des Defekts in logarithmischer Skala, geschätzte Neigungsabweichung in logarithmischer Skala, Anwendung eines Schwellenwertfilters mit jeweils  $0.1^\circ$ ,  $0.2^\circ$  und  $0.3^\circ$ .

Zur statistischen Auswertung wurden Testoberflächen mit 100 zufällig verteilten Defektpositionen und den oben genannten Parametern erzeugt. Mittels dem Verfahren in 6.1.1 wurden die Trefferquote und die Genauigkeit bei



variierendem Schwellenwert registriert. Die Ergebnisse sind in der Tabelle 6.4 dargestellt. Die Trefferquote bei Defekten mit variierender Neigung sind stabil. Der Schwellenwert bei  $0.2^\circ$  liefert das beste Ergebnis über alle Defektarten mit zwei Ausnahmen. Bei der Ebene liegt der Schwellenwert bei  $0.1^\circ$ . Bei der konvexen Oberfläche mit variierender Breite erhält man einen bessere Genauigkeit mit einem Schwellenwert von  $0.6^\circ$ .

Der Schwellenwert von  $0.1^\circ$  bei der Oberfläche ohne Krümmung und der Wert  $0.2^\circ$  bei Oberflächen mit Krümmung stimmen mit den Angaben der geforderten Neigungsunsicherheit während der Planung überein. Bei Oberflächen mit Krümmung kommt zusätzlich noch die Abweichung der Normalen in der Zelle zur gewünschten maximalen Unsicherheit hinzu.

Obwohl mit einem Flächenschwellenwert von  $1.5^2 \text{ mm}^2$  geplant wurde, lassen sich Fehler mit kleineren Ausdehnungen detektieren, solange sie eine hohe Neigungsabweichung besitzen. Bei der konvexen Oberfläche lässt sich dabei nicht derselbe Schwellenwert für sehr kleine Defekte und Defekte mit niedriger Normalenabweichung nutzen.

## 6.5 Approximationsmöglichkeit

Da die Planung für ein Bauteil nur einmal offline ausgeführt wird, können auch längere Planungszeiten akzeptiert werden. Trotzdem ist die Reduktion der Planungszeit von Interesse solange die Planqualität nicht beeinflusst wird, um beispielsweise Kosten zu sparen. In diesem Abschnitt wird untersucht inwiefern sich eine Approximation der Simulation und Modelle auf die Berechnungszeit und Qualität der Planung auswirken. Für beide Modelle hängt der Aufwand für die Auswertung von der Anzahl der verfolgten Strahlen ab. Mit ihnen steigt zum einen der Simulationsaufwand und zum anderen der Aufwand zur Fusionierung der Daten in das Modell. Beim Gauß-Prozess Modell spielt zusätzlich die Anzahl der induzierten Punkte eine große Rolle,

da für jede Evaluation eine von dieser Größe abhängige Matrix invertiert werden muss. Um den Aufwand zu reduzieren, kann die Auflösung während der Simulation reduziert werden. Dies führt aber zu weniger Messpunkten was im Allgemeinen zu einer Überschätzung der Unsicherheit führt. Um dem entgegen zu wirken, muss gleichzeitig das Modell der Oberfläche vergrößert werden. Beim Gauß-Prozess Modell handelt es sich bei der Vergrößerung um eine Reduzierung der Anzahl der induzierten Punkte und bei dem diskreten Modell um eine Vergrößerung des Flächenschwellenwerts. Dabei wird beim diskreten Modell der Neigungsschwellenwert in der Approximation nicht gesetzt.

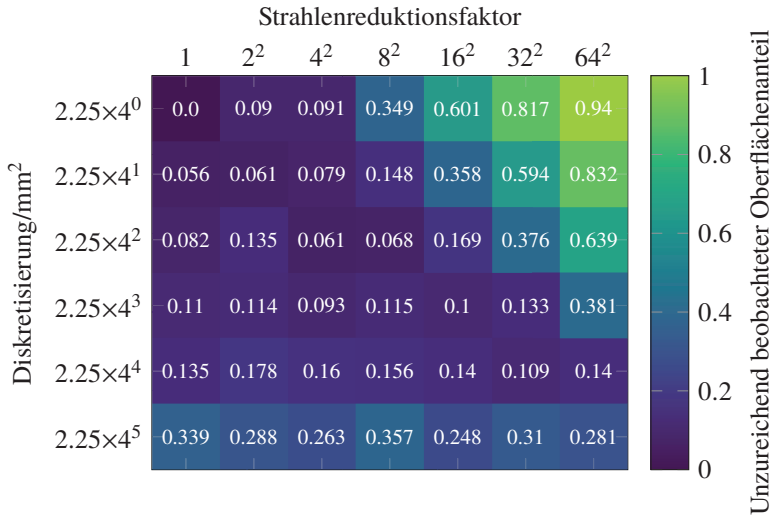
Tabelle 6.4: Trefferquote (T) und Genauigkeit (G) für die Inspektion der unterschiedlichen Testoberflächen mit den Defektarten Ausdehnung und Neigung in Abhängigkeit des Schwellenwerts.

<b>Oberfläche</b>	<b>Schwellenwert/mm</b>							
		0.1°	0.2°	0.3°	0.4°	0.5°	0.6°	0.7°
Eben-Breite	T	0.87	0.87	0.84	0.83	0.82	0.81	0.81
	G	0.98	0.98	0.95	0.95	0.94	0.94	0.94
Eben-Neigung	T	0.98	0.85	0.75	0.68	0.57	0.51	0.49
	G	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Konvex-Breite	T	0.99	0.98	0.97	0.97	0.95	0.95	0.93
	G	0.17	0.78	0.78	0.84	0.86	0.94	0.93
Konvex-Neigung	T	1.0	0.99	0.83	0.76	0.67	0.54	0.5
	G	0.45	0.99	1.0	1.0	1.0	1.0	1.0
Konkav-Breite	T	0.95	0.94	0.92	0.94	0.94	0.93	0.92
	G	0.15	0.98	0.98	0.99	0.99	0.99	0.99
Konkav-Neigung	T	1.0	0.97	0.79	0.73	0.64	0.53	0.49
	G	0.60	1.0	1.0	1.0	1.0	1.0	1.0

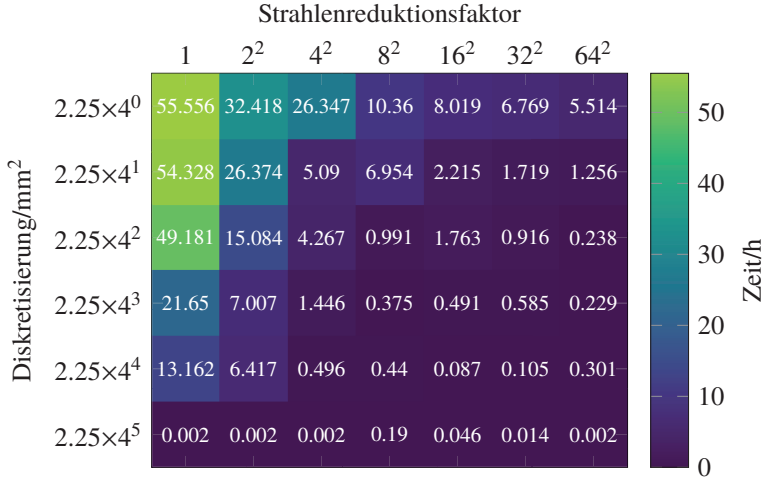
Für jede Kombination aus Vergrößerung und Strahlenreduktionsfaktor wurde mittels der Sweep-Planung mit den in den Abschnitten 6.3 und 6.4 berechneten optimalen Planlängen für die konvexe Oberfläche ein approximativer Plan berechnet. Zur Bewertung des approximativen Plans wurde dieser mit dem Ausgangsmodell (dem zu approximierenden Modell) ausgeführt. Beim Gauß-Prozess Modell wird als Bewertungsgröße direkt die erreichte maximale Standardabweichung der Höhe betrachtet. Beim diskreten Modell bleibt jeweils ein Oberflächenanteil der nicht beobachtet wird, so dass die maximale Standardabweichung keine Aussagekraft hat (Sie ist immer gleich des A-priori Wertes). In diesem Fall wird stattdessen der Oberflächenanteil betrachtet, der unzureichend beobachtet wurde, das heißt der mit der Fläche gewichtete Anteil an Zellen, dessen Standardabweichung der gemessenen Normale größer ist als die geforderte. Zur Berechnung der Pläne wurde derselbe Rechner wie in Abschnitt 6.3 verwendet.

Bei beiden Modellen sinkt die Zeit für die Berechnung des Plans signifikant mit der Reduktion der Kameraauflösung; siehe Abbildung 6.10b und 6.11b. Allerdings ist sie keine monoton fallende Funktion bezüglich des Strahlenreduktionsfaktors. Ist der Strahlenreduktionsfaktor relativ zur Diskretisierung zu hoch, kann es zu einer Erhöhung der Berechnungszeit kommen, da der Planungsalgorithmus mehr Durchläufe braucht. Die Berechnungszeit sinkt ebenfalls mit der Oberflächenauflösung: Beim diskreten Modell sinkt sie indirekt mit der Größe der Fläche, da weniger Flächenelemente beobachtet werden müssen und damit die Anzahl der Durchläufe sinkt und beim Gauß-Prozess Modell sinkt sie mit der Anzahl der induzierten Punkte, da die Dimension der Matrix in Gleichung (4.60) verkleinert wird.

In Abbildung 6.11a ist die erreichte maximale Standardabweichung für die Matrix aus Reduktionsfaktor der Kameraauflösung und der Vergrößerung für den Gauß-Prozess aufgetragen. Eine Reduzierung nur der Anzahl der Strahlen lässt sich bis auf das  $64^2$ -fache ausführen ohne die Spezifikation

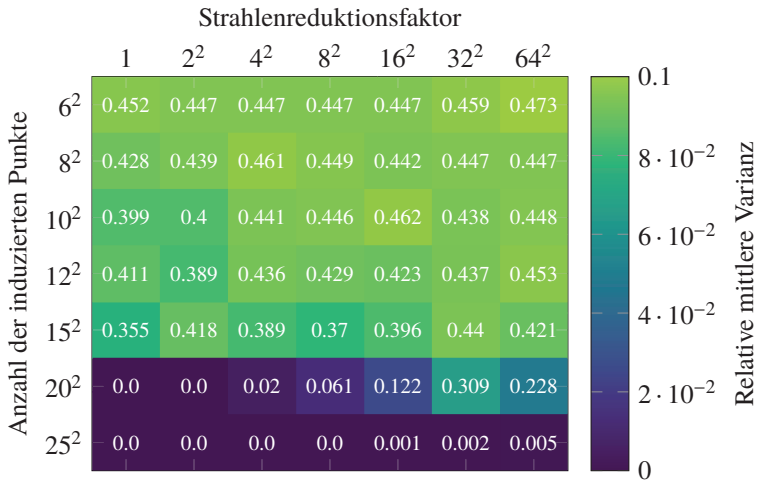


(a)

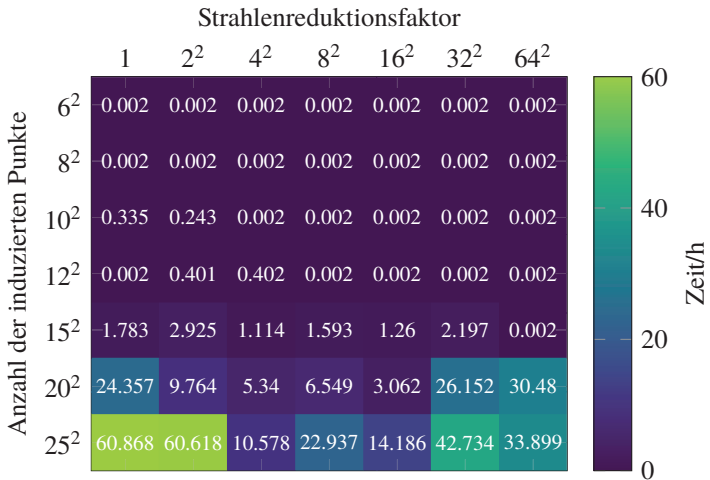


(b)

Abbildung 6.10: (a) Unzureichend beobachteter Oberflächenanteil und (b) Berechnungszeit der approximativ berechneten Pläne für das diskrete Modell.



(a)



(b)

Abbildung 6.11: (a) Maximale Varianz und (b) Berechnungszeit der approximativ berechneten Pläne für den Gauß-Prozess.

von  $U_{\text{suff}} = 0.1^2 \text{ mm}^2$  zu verletzen. Die Berechnungszeit reduziert sich dabei mindestens auf die Hälfte. Dies lässt sich damit erklären, dass das Oberflächenmodell eine niedrige Auflösung besitzt und die Planqualität nicht durch die Messauflösung, sondern durch die möglichen maximalen Messbereiche beschränkt ist. Die Reduktion der induzierten Punkte führt zu einem schnellen Qualitätsverlust der Pläne. Bereits eine Anzahl von  $15^2$  induzierten Punkten führt zu Plänen, die circa eine viermal höhere maximale Unsicherheit besitzen als die Spezifikation fordert. Das Ergebnis mit der niedrigsten Berechnungszeit bei gleichzeitiger Einhaltung der Spezifikation liefert eine Anzahl von  $20^2$  induzierten Punkten und einer Strahlenreduktion von  $4^2$ . Die Prädiktion geeigneter Parameter zur Approximation ist hieraus allerdings nicht ersichtlich und hängt sicherlich auch von der Oberfläche ab.

Beim diskreten Modell bleibt bei jeder Parameterkombination ein unzureichend beobachteter Oberflächenanteil; siehe Abbildung 6.11a. Bis zu einer Strahlenreduktion von  $32^2$  liegen die niedrigsten Werte entlang der Diagonalen und Nebendiagonalen. Werte auf der Diagonalen entsprechen demselben Faktor für die Strahlenreduktion und der Oberflächenvergrößerung.

## 6.6 Evaluation an einer realen Oberfläche

In diesem Abschnitt werden die Planungsverfahren an einer realen Oberfläche untersucht. Bei der Testoberfläche handelt es sich um eine Motorhaube für den Renault Twingo; siehe Abbildung 6.12. Eine Referenzoberfläche liegt in Form eines Mesh-Modells vor, dass aus einer vorhergehenden Rekonstruktion erstellt wurde. Aufgrund der Größe und Topologie des Objektes wird das diskrete Modell als Oberflächenmodell für die Berechnung des Plans gewählt.



Abbildung 6.12: Das zu untersuchende Karosserieteil ist eine Motorhaube für den Renault Twingo.

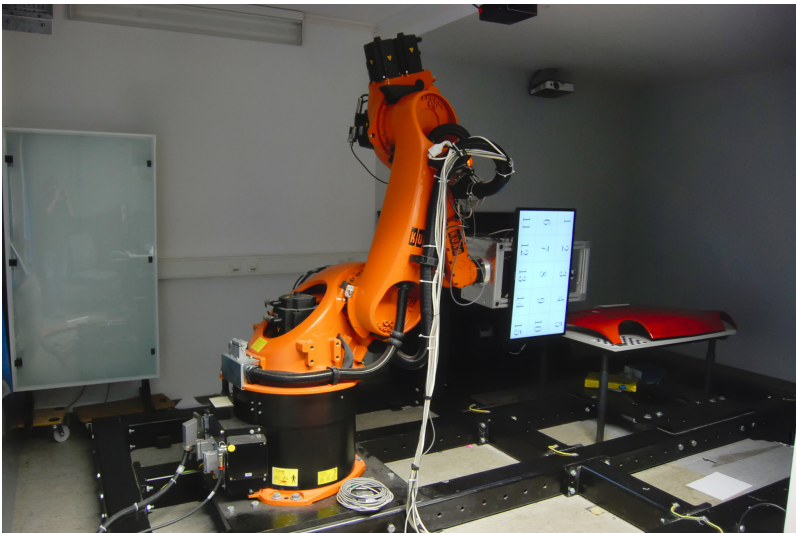


Abbildung 6.13: Messaufbau bestehend aus dem Industrieroboter KUKA KR 60 HA, kompaktem deflektometrischen Sensorkopf und Testoberfläche.

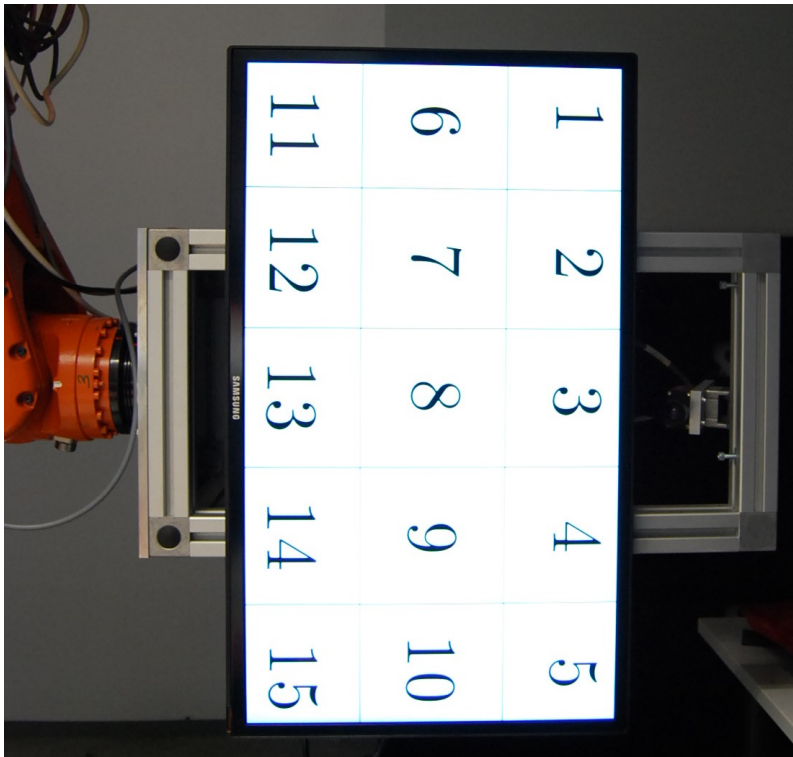


Abbildung 6.14: Nahaufnahme des verwendeten kompakten deflektometrischen Sensorkopfs bestehend aus Kamera und Bildschirm. Angezeigt wird ein Hilfsmuster zur Kalibrierung von Schirm-zu-Kamera.



### 6.6.1 Aufbau

Der Messaufbau, siehe Abbildung 6.13, besteht aus einem Industrieroboter mit einem kompakten deflektometrischen Sensor (Abbildung 6.14) und der Testoberfläche. Bei dem Industrieroboter handelt es sich um einen sechssachsigen KUKA KR 60 HA. Die Positionierwiederholgenauigkeit des Roboters ist mit 0.05 mm angegeben. Der Sensor hat eine Abmessung von 10 cm  $\times$  50 cm  $\times$  80 cm. Der benutzte LCD hat eine Bildschirmdiagonale von 30 Zoll und eine Auflösung von 1920 $\times$ 1080. Bei der verwendeten Kamera handelt es sich um eine AVR MANTA 123 mit einer Auflösung von 2000 $\times$ 1080. Es wird eine 5 mm Blende verwendet und die Bildweite gleich der Brennweite eingestellt. Das Testobjekt wird auf einem Inspektionstisch vor dem Roboter positioniert.

### 6.6.2 Kalibrierung

Die Planungsverfahren erfordern eine Bestimmung der Parameter und der relativen Lagen der Komponenten (Kamera-Schirm-Oberfläche) des Messsystems anhand mehrerer Kalibrierungen. Die relevanten Koordinatensysteme und ihre Transformationen untereinander sind in Abbildung 6.15 schematisch dargestellt. Bei den Koordinatensystemen handelt es sich um

- das Roboterbasiskoordinatensystem  $\mathcal{R}$  im Fuß des Roboters,
- das Flanschkoordinatensystem  $\mathcal{F}$  an der Abschlussfläche der letzten Roboterhauptachse,
- das Kamerakoordinatensystem  $\mathcal{C}$ ,
- das Schirmkoordinatensystem  $\mathcal{D}$ ,
- das Weltkoordinatensystem  $\mathcal{W}$ , das mittels eines Schachbrettmusters auf dem Inspektionstisch definiert wird,
- und das Objektkoordinatensystem  $\mathcal{O}$ .

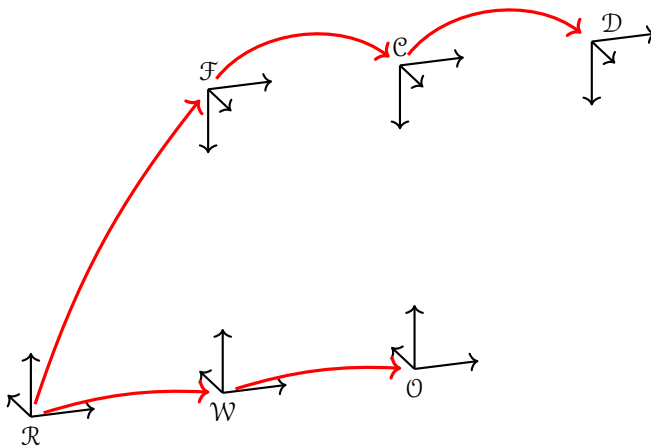


Abbildung 6.15: Darstellung der relevanten Koordinatensysteme und zu bestimmenden Transformationen zwischen diesen für die Kalibrierung des deflektometrischen Messsystems und Versuchsaufbaus.

Folgende Kalibrierungen müssen ausgeführt werden:

1. Kamera-Kalibrierung,
2. Hand-Auge-Kalibrierung (Bestimmung von  ${}^{\mathcal{F}}\mathbf{H}_{\mathcal{C}}$ ),
3. Kamera-zu-Bildschirm-Kalibrierung (Bestimmung von  ${}^{\mathcal{C}}\mathbf{H}_{\mathcal{D}}$ ),
4. Roboter-zu-Welt-Kalibrierung (Bestimmung von  ${}^{\mathcal{R}}\mathbf{H}_{\mathcal{W}}$ ),
5. Objekt-zu-Welt-Kalibrierung (Bestimmung von  ${}^{\mathcal{W}}\mathbf{H}_{\mathcal{O}}$ ).

Die Transformation  ${}^{\mathcal{R}}\mathbf{H}_{\mathcal{F}}$  ist eine Funktion bezüglich der Winkel an den Gelenkachsen des Roboters. Diese wird über die Robotersteuerung bereitgestellt. Im Folgenden werden die verwendeten Kalibrierungsverfahren kurz erläutert. Für eine weitere ausführlichere Beschreibung siehe die entsprechenden Referenzen in den Unterabschnitten.

### Kamerakalibrierung

Die Funktion der Kamerakalibrierung ist die Bestimmung der intrinsischen Parameter der Kamera. Die Kamerakalibrierung erfolgt anhand einer Implementierung von OpenCV [Its14, Its15]. Diese basiert auf den Methoden von Zhang [Zha00] und Heikkila [Hei97]. Durch mehrere Aufnahmen eines planaren Kalibrierungsmusters, meist einem Schachbrettmuster, wird jeweils eine Homographie zwischen den Kalibrierungsmusterpunkten und den abgebildeten Punkten auf der Kamerabildebene geschätzt. Aus ihr lassen sich die intrinsischen Kameraparameter mittels eines geschlossenen Ausdruck berechnen, unter der Annahme, dass die Verzerrungsparameter Null sind. Die Koeffizienten der Verzerrung werden anschließend mittels der Methode der kleinsten Quadrate bestimmt. Diese so gewonnenen Parameter der Kamera werden schließlich anhand einer nichtlinearen Minimierung des Rückprojektionsfehlers verfeinert.

### Hand-Auge-Kalibrierung

Mittels der Hand-Auge-Kalibrierung wird die räumliche Lage der Kamera in Relation zum sogenannten Flansch-Koordinatensystem des Roboters bestimmt. Eine Methode der Kalibrierung besteht darin mit dem Roboter unterschiedliche Positionen anzufahren und die dadurch induzierte Bewegung der Kamera zu beobachten. Daraus lässt sich die gesuchte Größe bestimmen.

Bei  $i = 1, \dots, m_h$  angefahrenen Positionen erhält man aus der Robotersteuerung die Lagen des Flansch-Koordinatensystems  ${}^{\mathcal{R}}\mathbf{H}_{\mathcal{F}_i}$  relativ zur Roboterbasis. Die Lagen der Kamera relativ zur Welt lassen sich bei bekannten intrinsischen Kameraparametern mittels dem Schachbrettmuster bestimmen:  ${}^{\mathcal{C}_i}\mathbf{H}_{\mathcal{W}}$ . Die gesuchte Größe ist die Lage der Kamera relativ zum Flansch  ${}^{\mathcal{F}}\mathbf{H}_{\mathcal{C}}$ . Diese Größe hängt nicht von der angefahrenen Position  $i$  ab, da die Lage der Kamera relativ zum Flansch fix bleibt. Insbesondere sind

${}^{\mathcal{F}}_i \mathbf{H}_{\mathcal{E}_i} = {}^{\mathcal{F}}_j \mathbf{H}_{\mathcal{E}_j} = {}^{\mathcal{F}} \mathbf{H}_{\mathcal{E}}$  für alle  $i, j$ . Nun wird aus den gegebenen Größen die Transformation von den Positionen  $i$  zu  $j$  jeweils für den Flansch  ${}^{\mathcal{F}}_i \mathbf{H}_{\mathcal{F}_j} = ({}^{\mathcal{R}} \mathbf{H}_{\mathcal{F}_i})^{-1} {}^{\mathcal{R}} \mathbf{H}_{\mathcal{F}_j}$  und für die Kamera  ${}^{\mathcal{E}_i} \mathbf{H}_{\mathcal{E}_j} = {}^{\mathcal{E}_i} \mathbf{H}_{\mathcal{W}} ({}^{\mathcal{E}_j} \mathbf{H}_{\mathcal{W}})^{-1}$  berechnet. Hiermit ergibt sich der Zusammenhang

$${}^{\mathcal{F}}_i \mathbf{H}_{\mathcal{F}_j} {}^{\mathcal{F}} \mathbf{H}_{\mathcal{E}} = {}^{\mathcal{F}} \mathbf{H}_{\mathcal{E}} {}^{\mathcal{E}_i} \mathbf{H}_{\mathcal{E}_j}. \quad (6.4)$$

Diese Gleichung hat die Form  $\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{B}$  mit  $\mathbf{A}, \mathbf{B} \in \text{SE}(3)$  und unbekanntem  $\mathbf{X} \in \text{SE}(3)$ , wofür in der Literatur zahlreiche Lösungsvorschläge [Fas05, Par94, Str06] existieren. Die Lösung dieser Gleichung ist uneindeutig, weshalb im rauschfreien Fall mindestens zwei unabhängige Positionspaare benötigt werden [Shi89]. In dieser Arbeit wird für die Lösung der Gleichung (6.4) der klassische Algorithmus von Tsai und Lenz [Tsa89] verwendet. Dieser separiert die Rotations- und Translationskomponenten und berechnet separat zunächst die Rotationskomponente und anschließend die Translationskomponente.

### Roboter-zu-Welt-Kalibrierung

Die Bestimmung des Weltkoordinatensystems  $\mathcal{W}$  relativ zum Roboterbasiskoordinatensystem  $\mathcal{R}$  wird mittels einer nichtlinearen Optimierung berechnet. Hierzu werden wie im vorhergehenden Unterabschnitt Positionen  $i = 1, \dots, m_h$  angefahren. Die Robotersteuerung liefert die Lagen des Flanschs relativ zum Roboterbasiskoordinatensystem  ${}^{\mathcal{F}} \mathbf{H}_{\mathcal{R}_i}$ . Die Lagen der Kamera  ${}^{\mathcal{W}} \mathbf{H}_{\mathcal{E}_i}$  relativ zur Welt werden mittels Schachbrettmuster bestimmt. Außerdem werden die bekannten  $m_p$  Punkte des Schachbrettmusters relativ zum Weltkoordinatensystem  ${}^{\mathcal{W}} \mathbf{p}_j \in \mathbb{R}^4$  aufgezeichnet. Seien die aufgezeichneten Punkte auf der Bildebene mit  $\tilde{\mathbf{p}}_{ij} \in \mathcal{E}_C$  mit  $j = 1, \dots, m_p$  bezeichnet. Hiermit lässt sich der Rückprojektionsfehler für die  $i$ -te Ansicht durch den Ausdruck

$$e_i(\mathbf{H}) = \sum_{j=1}^{m_p} \|\tilde{\mathbf{p}}_{ij} - \Pi(\mathcal{C} \mathbf{H}_{\mathcal{F}} \mathcal{F}_i \mathbf{H}_{\mathcal{R}} \cdot \mathbf{H} \cdot \mathcal{W} \mathbf{p}_j)\| \quad (6.5)$$

für ein Weltkoordinatensystem  $\mathbf{H} \in SE(3)$  bestimmen. Bei der Gleichung (6.5) werden die bekannten Punkte des Schachbrettmusters über die Koordinatensysteme  $\mathcal{W}, \mathcal{R}, \mathcal{F}_i$  und  $\mathcal{C}_i$  in das Kamerakoordinatensystem transformiert, anschließend auf den Kamerasensor projiziert und schließlich mit den entsprechenden beobachteten Punkten verglichen. Insgesamt berechnet sich die gesuchte Größe durch die Optimierung des Rückprojektionsfehlers über alle Ansichten:

$$\mathcal{R} \mathbf{H}_{\mathcal{W}} = \arg \min_{\mathbf{H} \in SE(3)} \sum_{i=1}^{m_h} e_i^2(\mathbf{H}). \quad (6.6)$$

Als Startwert für die Optimierung wurde eine erste Abschätzung mittels der ersten Ansicht gewählt:

$$\mathcal{R} \tilde{\mathbf{H}}_{\mathcal{W}} = \mathcal{R}_1 \mathbf{H}_{\mathcal{F}_1} \mathcal{F} \mathbf{H}_{\mathcal{C}} (\mathcal{W} \mathbf{H}_{\mathcal{C}_1})^{-1}. \quad (6.7)$$

Die angefahrenen Positionen aus der der Hand-Auge-Kalibrierung wurden für diese Kalibrierung wiederverwendet.

### Schirm-zu-Kamera Kalibrierung

Die Bestimmung der Lage der Kamera relativ zum Schirm erfordert die Nutzung eines Spiegels, da der Schirm nicht im Sichtbereich der Kamera liegt. Zur Kalibrierung wird das Verfahren aus [Rod10] verwendet, mit dem Unterschied, dass als Kalibriermuster keine Schachbrettmuster, sondern Phasenschiebemuster verwendet werden. Auf dem Bildschirm wird eine Sequenz von Phasenschiebemustern gezeigt und mit der Kamera ihr Spiegelbild beobachtet. Dies liefert eine deflektometrische Registrierung, die genutzt werden kann, um die Lage einer virtuellen Kamera relativ zum

Schirm zu berechnen. Die virtuelle Kamera sieht das gleiche Bild wie die reale Kamera, beobachtet den Punkt allerdings direkt. Seien  $\mathcal{C}_r$  das Koordinatensystem der realen Kamera sowie  $\pi_1, \dots, \pi_{m_s}$  mehrere Spiegel, über die der Bildschirm beobachtet wird mit ihrer jeweiligen Spiegelabbildung  $S_i$  und  $\tilde{\mathcal{C}}_i$  die Koordinatensysteme der virtuellen Kameras. Beobachtet werden die Lage der virtuellen Kameras relativ zum Bildschirm  ${}^{\mathcal{D}}\mathbf{H}_{\tilde{\mathcal{C}}_i}$ . Die homogene Matrix der virtuellen Kamera relativ zur realen Kamera ist durch

$${}^{\mathcal{C}_r}\mathbf{H}_{\tilde{\mathcal{C}}_i} = S_i \quad (6.8)$$

gegeben. Die Spiegelabbildungen haben die Eigenschaft, dass  $S_i = S_i^{-1}$  ist. Daraus ergibt sich

$$\tilde{\mathcal{C}}_0\mathbf{H}_{\tilde{\mathcal{C}}_i} = ({}^{\mathcal{C}_r}\mathbf{H}_{\tilde{\mathcal{C}}_0})^{-1} {}^{\mathcal{C}_r}\mathbf{H}_{\tilde{\mathcal{C}}_i} = S_0 S_i, \quad (6.9)$$

wobei sich die relativen Lagen der virtuellen Kameras untereinander durch

$$\tilde{\mathcal{C}}_0\mathbf{H}_{\tilde{\mathcal{C}}_i} = ({}^{\mathcal{D}}\mathbf{H}_{\tilde{\mathcal{C}}_0})^{-1} {}^{\mathcal{D}}\mathbf{H}_{\tilde{\mathcal{C}}_i} \quad (6.10)$$

bestimmen lassen. Die Gleichung (6.9) stellt einen Zusammenhang zwischen den unbekanntem Spiegellagen und den bekannten virtuellen Kameralagen her und dient als Grundlage zur Bestimmung eines Gleichungssystems mit der die Spiegellagen und daraus anschließend die Lage der realen Kamera sowie des Bildschirms bestimmt werden können.

### Objekt-zu-Welt-Kalibrierung

Die Lage der Testoberfläche relativ zum Weltkoordinatensystem ist die letzte Komponente, die für die Simulation der Messung benötigt wird. Der Planungsalgorithmus ist relativ robust gegenüber dieser Größe, weil die Konfigurationen relativ zum Objekt geplant werden und damit unabhängig vom Weltkoordinatensystem sind. Nur die Nebenbedingungen hängen von

der relativen Position des Testobjekts zur Welt ab. Beispielsweise kann die Veränderung der Lage des Testobjektes dazu führen, dass geplante Positionen vom Roboter nicht mehr angefahren werden können, weil sie außerhalb seines Arbeitsraumes liegen.

Aufgrund dieser schwachen Abhängigkeit wird für die Planung eine grobe Abschätzung der Lage  ${}^W\tilde{\mathbf{H}}_S$  genutzt, um den optimalen Plan zu berechnen. Die Abschätzung muss genau genug sein, so dass eine anschließende Korrektur eine geplante Konfiguration nicht ungültig macht.

Für die Ausführung des Plans ist allerdings eine genauere Abschätzung der Lage notwendig. Hierfür wird die Lage des Testobjektes mittels Rückprojektionsfehler verfeinert. Sei  $\theta_{0:n-1}$  der optimale Plan, der mittels der Abschätzung  ${}^W\tilde{\mathbf{H}}_S$  berechnet wurde. Seien  $\mathbf{l}_{1:n}(\mathbf{H})$  deflektometrische Messungen, die anhand des Simulators mittels dem optimalen Plan und der Lage der Oberfläche  $\mathbf{H}$  bestimmt werden. Seien  $\mathbf{l}'_{1:n}$  die realen Messungen, die mittels des optimalen Plans aufgenommen wurden. Da die Schätzung der Lage der Prüfoberfläche Fehler enthält, weichen die simulierten deflektometrischen Messungen  $\mathbf{l}_{1:n}({}^W\tilde{\mathbf{H}}_S)$  von den realen  $\mathbf{l}'_{1:n}$  ab. Für die Bestimmung der Objektposition wird das Optimierungsproblem

$${}^W\mathbf{H}_S = \arg \min_{\mathbf{H} \in \text{SE}(3)} \sum_{i=1}^n \|\mathbf{l}'_i - \mathbf{l}_i(\mathbf{H})\|^2 \quad (6.11)$$

mit einer geeigneten Norm minimiert. Hier wurde für die Norm

$$\|\mathbf{l}\|^2 = \sum_{u,v} \|\mathbf{l}(u,v)\|^2 \quad (6.12)$$

gewählt. Zur Optimierung wurde der Simulated-Annealing Algorithmus verwendet. Als Startwert diente die Ausgangsabschätzung  ${}^W\tilde{\mathbf{H}}_S$ .

### 6.6.3 Nebenbedingungen

Da durch die Kinematik des Roboters nicht jede Position anfahrbar ist, muss dies in den Nebenbedingungen berücksichtigt werden. Zum Beispiel führen Positionen, die zu nah am Roboter ausgeführt werden, zu kleinen Achsenwinkeln, die nicht angefahren werden können. Außerdem kann aufgrund des längs angebrachten Sensors, dieser zum Beispiel nicht beliebig um die  $z$ -Achse gedreht werden. Zusätzlich führte das hohe Sensorgewicht bei ausgestrecktem Roboterarm zu hohen Drehmomenten in den Gelenken, die zu einer weiteren Verkleinerung des Arbeitsraumes führen.

Eingeschränkt durch die Reichweite in Längsrichtung ist es nicht möglich, die gesamte Oberfläche zu inspizieren. Die Oberfläche wird so positioniert, dass die erste Hälfte vermessen werden kann. Die Nebenbedingungen werden so gewählt, dass es möglich ist einen Plan für die gesamte Oberfläche zu berechnen. Anschließend werden nur die anfahrbaren Konfigurationen zur Inspektion genutzt.

Der Arbeitsraum wird durch zwei Nebenbedingungen approximiert:

- Die Orientierung des Sensors um die vertikalen Achse wird zwischen  $-90^\circ$  und  $90^\circ$  beschränkt.
- Die erlaubte Position des Sensors wird auf einen achsenorientierten Quader beschränkt. Der Quader wird so gewählt, dass in longitudinaler Richtung die gesamte Oberfläche enthalten ist und prinzipiell inspiziert werden kann. Praktisch kann der Roboter trotz ausgestrecktem Arm nicht die zweite Hälfte der Oberfläche erreichen. In lateraler Richtung werden die Einschränkungen so gewählt, dass seitliche Positionen noch durch den Roboter anfahrbar sind.

Hohe Drehmomentlasten können durch diese Nebenbedingungen eingedämmt, aber nicht vollständig eliminiert werden. Diese treten allerdings nur bei ausgestrecktem Arm an der Grenze des Arbeitsraumes auf.



### 6.6.4 Durchführung und Ergebnisse der Planung

Als Referenzoberfläche dient ein rekonstruiertes Meshmodell der Motorhaube. Ein Glättungsfilter wird auf die Oberfläche angewendet, um hohe, aufgrund der Rekonstruktion entstandenen Frequenzanteile, zu reduzieren. Anschließend wird die Oberfläche in das in Abschnitt 4.4 vorgestellte Modell überführt. Für den Flächenschwellenwert wird  $A_{\max} = 4.0 \text{ mm}^2$  und für die Normalenabweichung  $\phi_{\max} = 0.1^\circ$  gewählt. Die A-priori Unsicherheit beträgt  $\sigma_A^2 = \tan(5^\circ)^2$ . Die gewünschte maximale Unsicherheit ist  $U_{\text{suff}} = \tan(0.1^\circ)^2$ . Für die Standardabweichung der Unsicherheit der Registrierung wurde durchschnittlich 0.86 mm gemessen. Das Sweep-Verfahren nutzt die Suche mit fester Schrittweite mit  $m = 5$ . Die Wahl des Startwertes für das Sweep-Verfahren stellte sich für diese Oberfläche als wichtig heraus. Die Startwerte wurden so ausgewählt, dass die Sichtpunkte homogen auf der Oberfläche verteilt liegen und die restlichen Parameter so, dass bei einer ebenen Fläche ein großer Messbereich vorliegt. Die Varianz der Messung für den Startplan mit 40 Messungen ist in Abbildung 6.16 gegeben.

Zunächst wird das Messsystem mit den beschriebenen Verfahren kalibriert. Anschließend wird ein Greedy-Plan berechnet und auf Basis des Ergebnisses mittels Sweep-Verfahren der optimale Plan berechnet. Der optimale Plan wird schließlich mittels des kalibrierten Messsystems ausgeführt und die Oberfläche inspiziert.

Bei einer Anwendung in der Praxis muss zusätzlich eine Rundtour durch die Messpunkte berechnet werden, die Kriterien wie Zeit oder Energie minimiert, falls diese nicht in der Kostenfunktion berücksichtigt wurden.

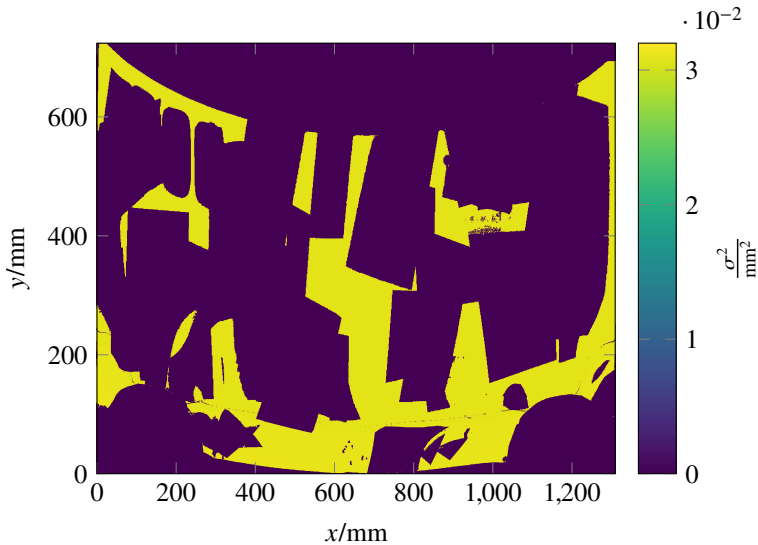


Abbildung 6.16: Varianz nach der Messung mit dem zufälligen Startplan des Sweep-Verfahrens für einen Planlänge von 40.

Dies ist ein klassisches Problem in der Informatik für das mehrere exakte und heuristische Lösungsansätze existieren [Lap92].

Die relative Varianz der Pläne, die mittels Greedy- und Sweep-Verfahren berechnet wurden und die aus der Suche des Sweep-Verfahrens resultierenden Pläne, sind in Abbildung 6.17 gegeben. Die Greedy-Planung berechnet einen Plan mit 62 Konfigurationen. Nach 33 Messungen sind bereits 80% der Oberfläche hinreichend untersucht. Die Sweep Planung berechnet einen um den Faktor 1.4 kürzeren Plan mit 40 Konfigurationen. Die Sweep-Planung sucht dabei insgesamt über 7 Planlängen; siehe Tabelle 6.5.

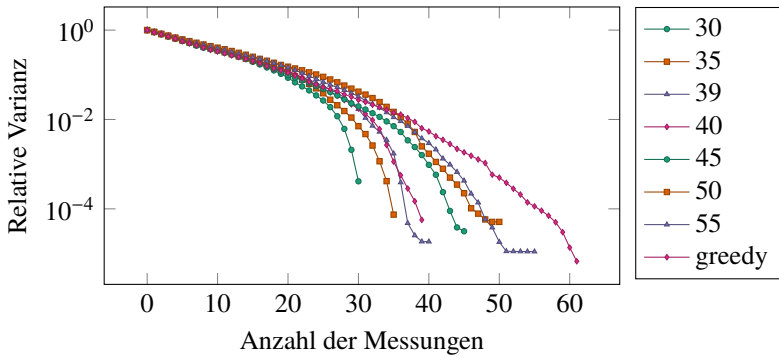
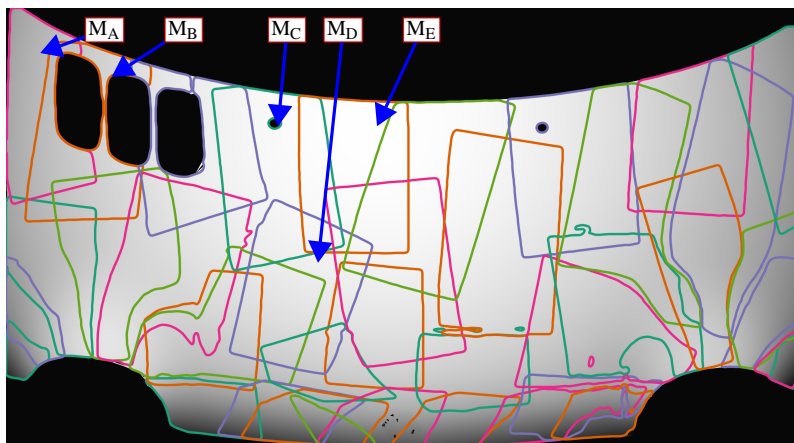


Abbildung 6.17: Verlauf der durchschnittlichen relativen Varianz für Planlängen 30, 35, 39, 45, 50 und 55 initiiert durch die Gridsuche/Sweep-Planung und der Greedy-Planung

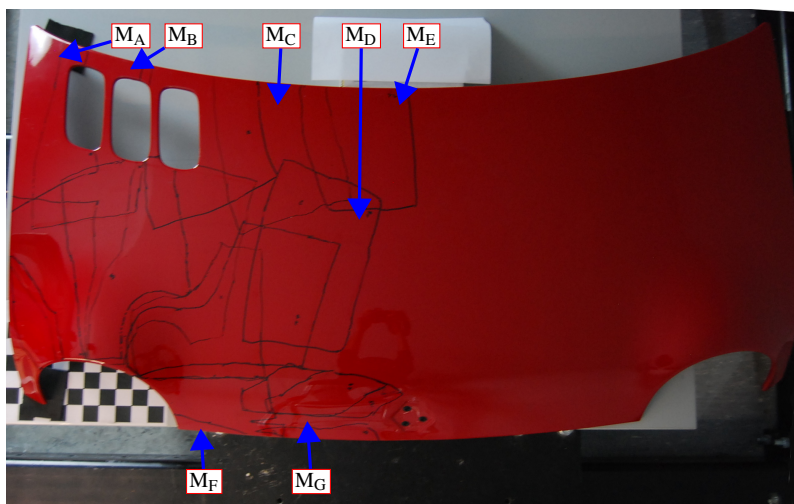
Tabelle 6.5: Ergebnisse der Gridsuche/Sweep-Planung. Die als optimal bestimmte Planlänge beträgt 40.

$n$	Varianz/ $10^{-5}$	#Sweep	Total/h	Sweep/h	Config/min
30	41.32	22	57.16	2.60	5.20
35	7.44	11	31.02	2.82	4.83
39	5.67	10	28.40	2.84	4.37
40	1.85	8	24.65	3.08	4.62
45	3.15	4	15.77	3.94	5.25
50	5.11	2	8.42	4.21	5.06
55	1.11	2	8.79	4.40	4.80

Bei der Ausführung des mit Sweep-Verfahren berechneten Plans wurden nur die Konfigurationen genutzt, die vom Roboter anfahrbar waren. Zunächst werden die Messbereiche der realen Messungen und die mittels Simulation berechneten Messungen untersucht; siehe Abbildung 6.18. Es ist zu



(a)



(b)

Abbildung 6.18: (a) Messbereiche des vom Sweep-Verfahren berechneten optimalen Plans. Zur besseren Erkennung wurden die Konturen geglättet und in unterschiedlichen Farben dargestellt. (b) Messbereiche der real ausgeführten Messung.

erkennen, dass es Abweichungen zu den geplanten Messungen gibt. Messbereiche  $M_A$  und  $M_B$  stimmen gut überein, Messbereiche  $M_C$  und  $M_D$  sind in der real ausgeführten Messung gestaucht, während Messbereich  $M_E$  gestreckter ist. Es kann damit nicht mit Sicherheit gesagt werden, ob die Spezifikation der Normalenunsicherheit an allen Stellen erfüllt ist. Da der Plan Überlappungen enthält, ist zumindest die Abdeckung der Oberfläche in leicht gekrümmten Bereichen nicht verletzt. Vorne im Krümmungsbereich der Motorhaube (Bereiche  $M_F$  und  $M_G$ ) gibt es allerdings auch Stellen, die vom Sensor nicht erfasst werden. Der Grund für die Abweichung liegt zum einen in der Positionierungsgenauigkeit des Roboters und an der Kalibrierungsgenauigkeit. Die Positionierungsgenauigkeit des Roboters wird vom Hersteller nicht angegeben, da üblicherweise die Anfahrpunkte eingelernt werden und hierfür die Wiederholgenauigkeit ausschlaggebend ist. Eine weitere Ungenauigkeit entsteht durch die Last des Sensors, die bei der Bahnplanung im Roboter nicht berücksichtigt wird. Einen wesentlichen Einfluss auf die Abweichung der Messbereiche hat die Orientierung des Sensors. Beispielsweise führt eine Abweichung der Orientierung der Kamera von  $1^\circ$  bei einem Abstand von 1 m auf eine Abweichung der Messposition von circa 1.7 cm. Dies kann allerdings alleine noch nicht die Abweichung von teilweise circa 5 cm erklären. Da insbesondere Messbereiche  $M_A$  und  $M_B$  an einer markanten Stelle der Oberfläche gut mit der Simulation übereinstimmen, ist eine Ungenauigkeit der Oberflächenorientierung naheliegend.

Mittels einer visuellen Inspektion der Oberfläche konnten fünf lokale Defekte im gesamten Messbereich identifiziert werden; siehe Abbildung 6.19. Abbildungen 6.21, 6.22 und 6.20 zeigen, dass diese, bis auf Defekt  $D_B$ , sowohl in den Streifenbildern als auch in der Registrierung detektiert werden können. Defekt  $D_B$  konnte nicht erkannt werden, da zum einen die Normalenabweichung gering ist und zum anderen die aufgrund des sogenannten *Orangenhaut-Effekts*, einer künstlich erzeugen Welligkeit der Oberfläche, der Defekt schwer zu erkennen ist.



Abbildung 6.19: Defektpositionen auf der Oberfläche

## 6.7 Zusammenfassung

In diesem Kapitel wurden die Oberflächenmodelle und Planungsverfahren evaluiert. Zunächst wurden die Gütefunktionen des Gauß-Prozess-Modells untersucht. Hier zeigte sich, dass sowohl die Spur als auch die Determinante für die Planung geeignet ist. Da die Spur sich effizient berechnen lässt und intuitiv ist, wurde sie für den Rest der Evaluation verwendet. Anschließend wurden Pläne mittels der Greedy- und Sweep-Planung unter Verwendung jeweils des Gauß-Prozess-Modells und des diskreten Modells berechnet. Die Greedy-Planung erreicht in den ersten Messungen hohe Überdeckung. Da aber nur einen Schritt im Voraus geplant wird, entstehen mehrere verstreute Bereiche mit niedriger Varianz, was die Planlänge insgesamt erhöht.

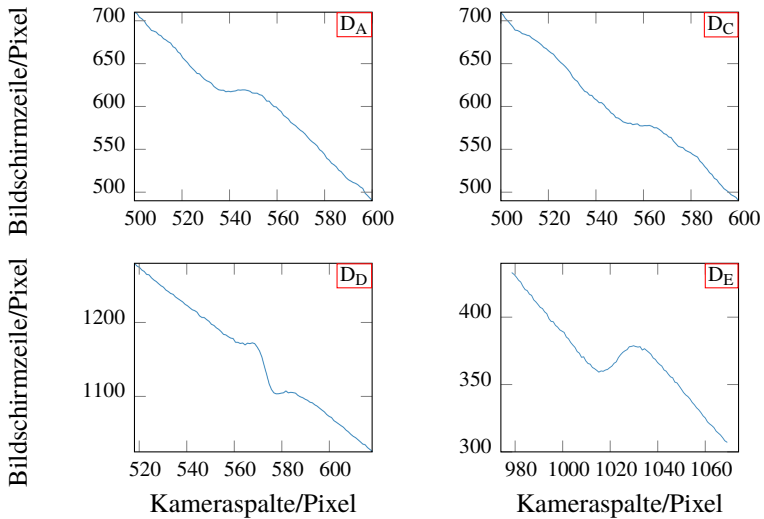


Abbildung 6.20: Schnitt entlang der Zeilen durch die deflektometrische Registrierung an den jeweiligen Defektpositionen.

Die Sweep-Planung berechnete im Vergleich zur Greedy-Planung  $1.4\text{-}2.75 \times$  kürzere Pläne, bei Erhöhung des Berechnungsaufwandes. Die Qualität der Sweep-Planung wurde anhand einer simulationsbasierten Oberflächeninspektion für beide Modelle evaluiert. Dazu wurden Testoberflächen mit additiven gaußförmigen Defekten überlagert und mit den berechneten Plänen inspiziert. Beim diskreten Modell wurden Testoberflächen mit Defekten variierender Defektausdehnung und maximaler Neigung verwendet. Dabei konnten hohe Trefferquoten und Genauigkeiten für Defektausdehnungen unter der geplanten Oberflächenauflösung erreicht werden. Die Neigungsempfindlichkeit entsprach dem in der Spezifikation angegebenen Grenze. Das Gauß-Prozess-Modell ist aufgrund der Approximation mittels induzierten Punkten für größere Defektausdehnungen geeignet. Hier wurde die Defektausdehnung konstant gehalten und die Defekthöhe variiert. Auch hier konnten hohe Werte unter der spezifizierten Höhenunsicherheit erreicht werden.

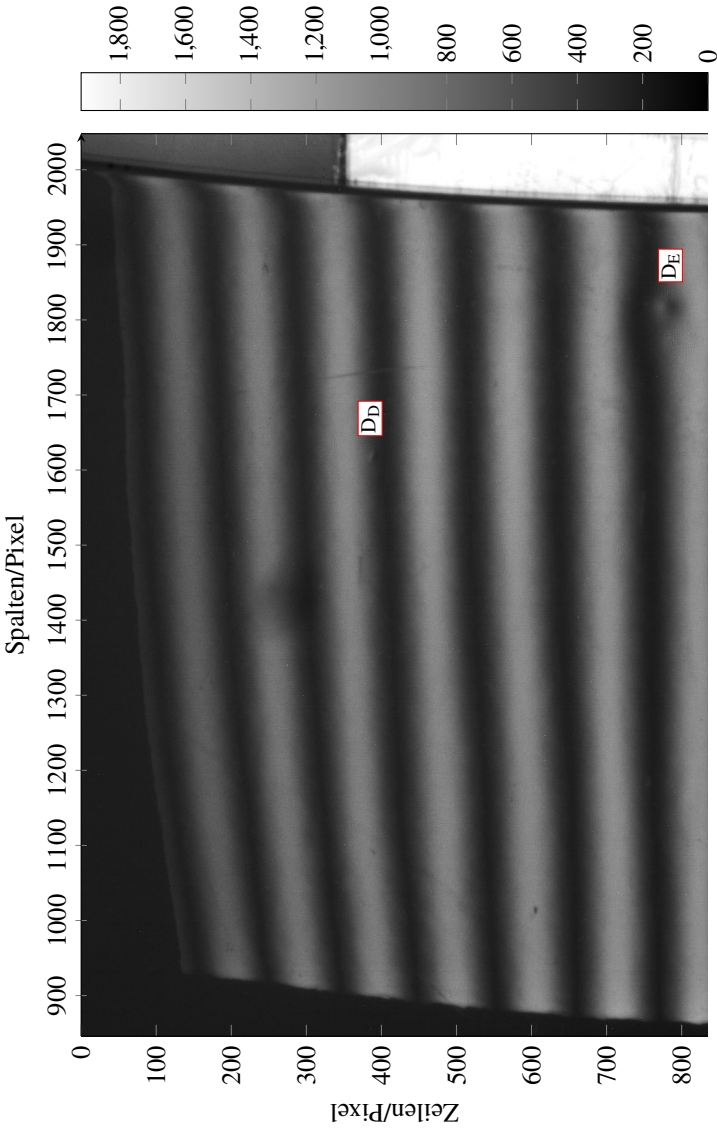
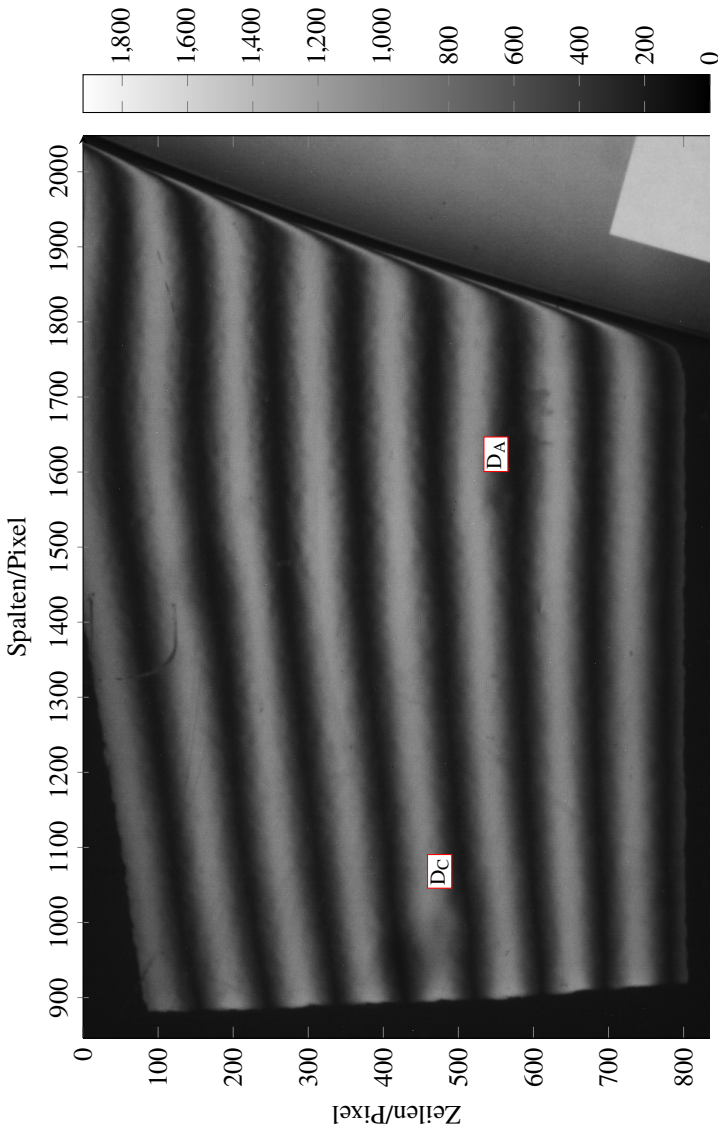


Abbildung 6.21: Defekte  $D_D$  und  $D_E$  sind in den Streifenbildern erkennbar.



Abbildung 6.22: Defekte  $D_A$ ,  $D_C$  sind in den Streifenbildern erkennbar.

Weiterhin wurden die Approximationsmöglichkeiten der Oberflächenmodelle untersucht. Zur Approximation diente die Reduktion der Anzahl der verfolgten Strahlen. Da dies zu einer Überschätzung der Unsicherheit führt, wurde gleichzeitig die Oberflächenauflösung reduziert. Beim diskreten Modell über den Flächenschwellenwert und beim Gauß-Prozess-Modell über die Anzahl der induzierten Punkte. Bei letzterem sinkt dadurch der Aufwand signifikant, da für die Auswertung der Unsicherheit eine zur Anzahl der induzierten Punkte quadratisch wachsende Matrix invertiert werden muss. Für verschiedene Kombinationen aus Oberflächenauflösung und Strahlenduktionsfaktor wurden für die konvexe Oberfläche Pläne mit der optimalen Planlänge, die aus den vorherigen Experimenten mittels Sweep-Planung bestimmt wurde, berechnet. Anschließend wurde der Plan verwendet, um die Unsicherheit jeweils mit dem approximierten Modell und mit dem Ausgangsmodell zu berechnen. Aus den Beobachtungen über die Approximation lässt sich schließen, dass beim Gauß-Prozess-Modell aufgrund der bereits niedrigen Oberflächenauflösung durch die induzierten Punkte die Anzahl der Strahlen reduziert werden kann. Dabei sinkt die Berechnungszeit nicht gemäß dem Strahlenreduktionsfaktor und steigt bei hohen Reduktionsfaktoren, da die Anzahl der Durchläufe bei der Sweep-Planung steigt. Das Modell verhält sich sensitiver gegenüber einer Reduktion der induzierten Punkte. Die Planungszeit reduziert sich allerdings signifikant. In dem Experiment konnte mit der besten Approximation ( $20^2$  induzierten Punkte und ein Strahlenreduktionsfaktor von  $4^2$ ) eine Aufwandreduktion um das 11-fache erreicht werden. Die Prädiktion dieser Parameter ist allerdings nicht ersichtlich und hängt sicherlich auch von der Oberfläche ab. Für das diskrete Modell blieb bei jeder verwendeten Kombination aus Oberflächenauflösung und Strahlenanzahl ein Oberflächenanteil der nicht ausreichend, also nicht gemäß der gewünschten maximalen Unsicherheit, beobachtet wurde. In dem Experiment lagen die niedrigsten Werte entlang der Diagonalen und Nebendiagonalen. Für Werte auf der Diagonalen wurde derselbe Faktor für die

Strahlenreduktion und der Oberflächenvergrößerung genutzt. Allerdings gilt diese Beobachtung nicht für beliebig hohe Faktoren. Dabei ist mit einem unzureichend beobachteten Oberflächenanteil von 6% und 14% zu rechnen, wobei dies wiederum von der Oberfläche abhängt.

In einem abschließenden Experiment wurden die Planungsverfahren an einer realen Oberfläche, einer Motorhaube des Renault Twingo, evaluiert. Aufgrund der Größe und Topologie der Oberfläche wurde das diskrete Oberflächenmodell verwendet. Die Sweep-Planung berechnet im Vergleich zur Greedy-Planung einen um den Faktor 1.4 kürzeren Plan mit 40 Messungen. Um die Qualität des Plans hinsichtlich der Detektionsgüte zu untersuchen, wurden mittels einer visuellen Inspektion Defekte auf der Oberfläche detektiert und anschließend untersucht, inwiefern diese in den geplanten Messungen beobachtbar sind. Da mit dem menschlichen Auge Defekte weit unter der gesetzten Toleranz von  $0.1^\circ$  detektiert werden können und die maximale Abweichung der gefundenen Defekte unbekannt ist, konnte nur eine qualitative Evaluation erfolgen. Insgesamt konnten mittels der Messungen, die mit dem Sweep-Plan ausgeführt wurden vier von fünf Defekten detektiert werden. Ein Defekt konnte aufgrund der niedrigen Abweichung und des Rauschens nicht detektiert werden.



# 7 Zusammenfassung und Ausblick

## 7.1 Zusammenfassung

In dieser Arbeit wurde ein Ansatz zur Planung der deflektometrischen Oberflächeninspektion untersucht, der Unsicherheiten während des Messvorgangs mitberücksichtigt. Hierzu wurde die Inspektion von Oberflächen mittels eines probabilistischen Rahmenwerks modelliert, das die Berücksichtigungen der Unsicherheit der Messung und gleichzeitige Betrachtung von Wissen über die Oberfläche ermöglicht. In dieser Formulierung wird die Inspektion als ein sequentielles probabilistisches Entscheidungsproblem modelliert. Für die Planung wird die Informationsbeschaffung von der Inspektionsentscheidung entkoppelt. Die Informationsbeschaffungsstrategie führt Messungen aus, bis die Unsicherheit bezüglich der Oberfläche einen Schwellenwert unterschreitet. Anschließend wird die Prüfentscheidung getroffen. Aufgrund der Unsicherheit im System muss zum einen eine Verteilung über der Oberfläche als Zustand genutzt werden und zum anderen erfordert die exakte Lösung des Problems eine Rückkopplung, das heißt, dass der Strategie nach jeder Messung der aktualisierte Zustand zugeführt werden muss. Da die Berechnung einer Strategie bei hohen Zustandsdimensionen aufwendig ist und alternativ eine Approximation mittels Open-Loop oder Modellprädiktiver Planung einen erheblichen Planungsaufwand während den Messungen erzeugen würde, wurde auf die Rückführung verzichtet. Unter dem Gesichtspunkt, dass eine Oberflächeninspektion meist in einer kontrollierbaren Umgebung

stattfindet und die Unsicherheiten niedrig gehalten werden können, wurden Planungsverfahren zur Berechnung eines deterministischen Plans betrachtet. Mittels eines Simulators wird aus einer gegebenen Sensorkonfiguration und einer Referenzoberfläche die zugehörige Messung simuliert und die Messdaten in ein probabilistisches Oberflächenmodell fusioniert. Für das Oberflächenmodell sowie die Fusion der Messungen in diesen wurden zwei Modelle vorgestellt.

Das Gauß-Prozess Modell modelliert die Abweichung der Oberflächenhöhe zur Referenzhöhe als ein Gauß-Prozess. Sie erlaubt Messoperationen über beliebige lineare Operatoren zu modellieren und in das Modell einfließen zu lassen. Aufgrund des hohen Berechnungsaufwandes wurden approximative Inferenzverfahren betrachtet.

Das diskrete Modell modelliert die Oberfläche als eine Menge von kleinen Spiegeln mit einer Wahrscheinlichkeitsverteilung über deren Form. Speziell wurden ebene Spiegel betrachtet mit den Neigungen als Parametern. Das Modell ist für Referenzoberflächen mit beliebigen Topologien anwendbar. Im Vergleich zum Gauß-Prozess Modell ist der Fusionsaufwand gering.

Schließlich wurden zwei Planungsverfahren vorgestellt. Bei der Greedy-Planung wird das Optimierungsproblem mit einer Sequenz von Optimierungsproblemen approximiert, die jeweils nur für einen Zeitschritt bezüglich einer Sensorkonfiguration optimieren.

Die Sweep-Planung ist eine Abwandlung hiervon, die in einem hierarchischen zweistufigen Prozess einen Plan berechnet. Im äußeren Prozess wird eine feste Anzahl von Messungen gesucht. Dabei wird der innere Prozess verwendet, der bei fester Anzahl an Messungen mittels mehreren Durchläufen jeweils eine Sensorkonfiguration auswählt und diese optimiert. Zur Optimierung der einzelnen Sensorkonfigurationen wird das Simulated-Annealing-Verfahren verwendet. Hierzu wurde die Pertubationsfunktion für die deflektometrische Anwendung angepasst.

Anhand von simulationsbasierten Experimenten wurden die Oberflächenmodelle und Planungsverfahren evaluiert. Die Sweep-Planung erreichte im Vergleich zur Greedy-Planung circa  $1.4 - 2.75 \times$  kürzere Pläne. Anhand simulationsbasierter Inspektionen konnte gezeigt werden, dass die Sweep-Planung die geplante Spezifikation einhält. Aufgrund des hohen Berechnungsaufwandes wurde weiter untersucht, inwiefern sich eine Approximation der Simulation und Modelle auf die Berechnungszeit und Qualität der Planung auswirkt. Die Approximation erfolgt durch die Reduktion der Anzahl der verfolgten Strahlen und gleichzeitige Vergrößerung der Oberflächenmodelle. Beim Gauß-Prozess konnte, da dieser bereits eine niedrige Oberflächenauflösung besitzt, die Anzahl der Strahlen bei gleichbleibender Anzahl an induzierten Punkten reduziert werden, ohne die Spezifikation zu verletzen. Beim Sweep-Verfahren führt eine Approximation zu einem Oberflächenanteil, der nicht ausreichend beobachtet wird. Dieser lag bei den Experimenten im Bereich von 6%-14% für einen approximativ äquivalenten Faktor für Strahlenreduktion und Oberflächenauflösungsreduktion.

Schließlich wurde anhand einer Motorhaube für den Renault Twingo die Anwendung der Planungsverfahren mit dem diskreten Modell an einer realen Oberfläche untersucht und seine Eignung festgestellt. Abweichungen der gemessenen Messbereiche von den berechneten Messbereichen lagen an der Kalibrierungs- bzw. Positionierungsgenauigkeit des Handhabungsgerätes, die mit aufwändigeren Mitteln reduziert werden können. Insbesondere ist eine präzise Referenzoberfläche für die Bestimmung der Position der Testoberfläche und eine Kalibrierung des Handhabungsgerätes mit Last vorteilhaft.

## 7.2 Ausblick

In diesem Abschnitt werden mögliche neue Forschungsrichtungen und Erweiterungen vorgeschlagen.

### **Positionierungsgenauigkeit**

In dieser Arbeit wurde angenommen, dass die Position der Oberfläche sowie die Kalibrierungsparameter mit einer ausreichend niedrigen Unsicherheit bekannt sind und dass das Handhabungsgerät mit einer beliebigen Genauigkeit positioniert werden kann. Dadurch kann die Unsicherheit in der Richtung der verfolgten Lichtstrahlen vernachlässigt werden. Allerdings kann bei geforderten hohen Oberflächenauflösungen das Erreichen der dafür geforderten Unsicherheiten und Positioniergenauigkeit mit viel Aufwand verbunden sein. Wird dann auf eine genaue Kalibrierung und Positionierung verzichtet, wirken sich die Abweichungen des mittels ausgeführten Plans beobachteten Messbereiche und des mittels simulierten Plans berechneten Messbereiche auf die Planqualität aus. In diesem Fall reicht die Berechnung eines zustandsunabhängigen Plans nicht aus, sondern erfordert entweder eine Strategie mit Zustandsrückführung oder eine Neuplanung nach jedem Messschritt mit aktualisiertem Zustand. Dies hat die wie oben diskutierten Nachteile. Alternativ wäre eine Berechnung eines konservativen Plans denkbar, der nur den trotz Unsicherheit sicher beobachteten Messbereich berücksichtigt. Dadurch würden ausreichend Überdeckungen der Messbereiche geplant, so dass trotz Positionierungsunsicherheiten die Spezifikation eingehalten werden könnte.

### **Infrarotdeflektometrie**

In der Automobilindustrie ist es oftmals von Interesse, Defekte auf Karosserieteile schon vor der Lackierung zu detektieren. Dies kann zwar mit herkömmlichen Triangulationsverfahren erreicht werden, allerdings reicht deren Empfindlichkeit häufig nicht aus, da nach der Lackierung schon kleine Neigungsabweichungen mit steigendem Abstand deutlich gesehen werden. Die Infrarotdeflektometrie [Höf17] weitet den Anwendungsbereich der Deflektometrie von spiegelnden Oberflächen auf raue Oberflächen aus.



Aufgrund der höheren Wellenlänge des Infrarotlichts wird diese an rauen Oberflächen reflektiert, womit sich das Prinzip der Deflektometrie anwenden lässt. Mittels eines Infrarotmustergenerators und einer Infrarotkamera lässt sich die Oberfläche nach dem deflektometrischen Prinzip inspizieren. Ein Problem ist dabei einen geeigneten Mustergenerator zu entwerfen. Eine technisch einfache Umsetzung ist die Nutzung eines heißen Drahtes [Höf16]. Eine Umsetzung der Planungsverfahren für solch ein Messsystem führt zu neuen Herausforderungen. Eine Besonderheit des Messsystems ist, dass einzelne Messung anhand des Drahtes für die Bestimmung der Normale uneindeutig sind. Die möglichen Normalen reduzieren sich auf einen Pfad, womit die Approximation mittels einer Normalverteilung als unangemessen erscheint. Da die Messung mit Draht nur einen kleinen Messbereich besitzt, erfordert die Inspektion von Oberflächen viele Messungen, was hohe Ansprüche an das Optimierungsverfahren stellt. Hier ist es möglicherweise sinnvoll den Messsensor entlang von Pfaden zu führen und eine Parametrisierung als Sensorkonfiguration zu nutzen.

### **Planung für die Rekonstruktion**

Eine weitere denkbare Erweiterung wäre die Planung bei unbekannter Referenzoberfläche. Anwendung wäre hierbei beispielsweise die Erfassung der geometrischen Form einer spiegelnden Oberfläche. Herausforderung sind hier zum einen die probabilistische Schätzung der Oberfläche aus der deflektometrischen Messung. Hier könnten Ansätze von Liu [Liu15] genutzt werden, um eine eindeutige Rekonstruktion aus einer Messung zu erhalten oder Regularisierungspunkte zu bestimmen. Weiterhin ist zur Planung eine Prädiktion der Oberfläche mittels Teilwissen erforderlich, auf deren Basis eine sinnvolle Planung erfolgen kann.



# A Anhang

## A.1 Anhang 1

Dieser Abschnitt stellt eine Sammlung von Formeln für die Umformung von Normalverteilungen zusammen. Seien im Folgenden  $\underline{x}$  und  $\underline{y}$  stets normalverteilte Zufallsvektoren.

### A.1.1 Marginalisierung

Sei die Verbundverteilung der Zufallsvektoren  $\underline{x}$  und  $\underline{y}$  durch

$$\begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \underline{\mu}_x \\ \underline{\mu}_y \end{pmatrix}, \begin{pmatrix} \underline{\Sigma}_{xx} & \underline{\Sigma}_{xy} \\ \underline{\Sigma}_{yx} & \underline{\Sigma}_{yy} \end{pmatrix} \right), \quad (\text{A.1})$$

gegeben, dann berechnet sich die Marginalwahrscheinlichkeitsdichtefunktion des Zufallsvektors  $\underline{x}$  durch

$$\underline{x} \sim \mathcal{N}(\underline{\mu}_x, \underline{\Sigma}_{xx}) . \quad (\text{A.2})$$

## A.1.2 Bedingte Dichte

Sei die Verbundverteilung der Zufallsvektoren  $\underline{x}$  und  $\underline{y}$  durch

$$\begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{pmatrix} \right) \quad (\text{A.3})$$

gegeben, dann berechnet sich die bedingte Dichte des Zufallsvektors  $\underline{x}$  gegeben  $\underline{y}$  durch

$$\underline{x} | \underline{y} \sim \mathcal{N}(\mu_{x|y}, \Sigma_{x|y}) \quad (\text{A.4})$$

mit

$$\mu_{x|y} = \mu_x + \Sigma_{xy} \Sigma_{yy}^{-1} (y - \mu_y) \quad (\text{A.5})$$

$$\Sigma_{x|y} = \Sigma_{xx} - \Sigma_{xy} \Sigma_{yy}^{-1} \Sigma_{yx} . \quad (\text{A.6})$$

## A.1.3 Verbundverteilung

Ist die Verteilung des Zufallsvektors

$$\underline{x} \sim \mathcal{N}(\mu_x, \Sigma_{xx}) \quad (\text{A.7})$$

und die bedingte Verteilung

$$\underline{y} | \underline{x} \sim \mathcal{N}(A\underline{x} + \underline{n}, \Sigma_{y|x}) \quad (\text{A.8})$$

gegeben, dann ergibt sich für die Verbundverteilung

$$\begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \boldsymbol{\mu}_x \\ A\boldsymbol{\mu}_y + \mathbf{n} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xx}A^T \\ A\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_y | x + A\boldsymbol{\Sigma}_{xx}A^T \end{pmatrix} \right). \quad (\text{A.9})$$

### A.1.4 Inferenz

Bei der Inferenz ist die Dichte des Zufallsvektors

$$\underline{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx}) \quad (\text{A.10})$$

und die bedingte Dichte

$$\underline{y} | \underline{x} \sim \mathcal{N}(A\underline{x} + \mathbf{n}, \boldsymbol{\Sigma}_y | x) \quad (\text{A.11})$$

gegeben. Mittels der Ergebnisse in den Unterabschnitten A.1.3 und A.1.2 lässt sich die bedingte Dichte  $\underline{x} | \underline{y}$  herleiten. Es gilt

$$\underline{x} | \underline{y} \sim \mathcal{N}(\boldsymbol{\mu}_{x|y}, \boldsymbol{\Sigma}_{x|y}) \quad (\text{A.12})$$

mit

$$\boldsymbol{\mu}_{x|y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xx}A^T \mathbf{K}^{-1} (\mathbf{y} - A\boldsymbol{\mu}_x - \mathbf{n}) \quad (\text{A.13})$$

$$\boldsymbol{\Sigma}_{x|y} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xx}A^T \mathbf{K}^{-1} A\boldsymbol{\Sigma}_{xx} \quad (\text{A.14})$$

$$\mathbf{K} = \boldsymbol{\Sigma}_y | x + A\boldsymbol{\Sigma}_{xx}A^T. \quad (\text{A.15})$$

## A.2 Anhang 2

In diesem Abschnitt werden die in Unterabschnitt 4.3.6 hergeleiteten Gleichungen (4.56) umgeformt, so dass eine Matrix mit kleinerem Rang invertiert werden muss. Der Vollständigkeit halber sind die Gleichungen nochmals aufgelistet:

$$\mu_a = \mu_U + K_{UX}K^{-1}(q - \mu_X) \quad (\text{A.16})$$

$$\Sigma_a = K_{UU} - K_{UX}K^{-1}K_{XU} \quad (\text{A.17})$$

$$K = K_{XU}K_{UU}^{-1}K_{UX} + W. \quad (\text{A.18})$$

Unter Anwendung der Sherrison - Woodmore Identität

$$(ACB + D)^{-1} = D^{-1} - D^{-1}B(C^{-1} + BD^{-1}A)^{-1}AD^{-1} \quad (\text{A.19})$$

wird zunächst die Abbildungsmatrix im zweiten Summanden der Gleichung (A.16) umgeformt. Anwendung der Identität (A.19) liefert

$$K_{UX}K^{-1} \quad (\text{A.20})$$

$$= K_{UX} \left( K_{XU}K_{UU}^{-1}K_{UX} + W \right)^{-1} \quad (\text{A.21})$$

$$= K_{UX} \left( W^{-1} - W^{-1}K_{XU} \right) \quad (\text{A.22})$$

$$\left( K_{UU} + K_{UX}W^{-1}K_{XU} \right)^{-1} K_{UX}W^{-1} \\ = K_{UX}W^{-1} - K_{UX}W^{-1}K_{XU} \cdot \quad (\text{A.23})$$

$$\left( K_{UU} + K_{UX}W^{-1}K_{XU} \right)^{-1} K_{UX}W^{-1}.$$

Ausklammern von  $\mathbf{K}_{UX}\mathbf{W}^{-1}$ , des Terms  $(\mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU})^{-1}$  und weitere Vereinfachung der Faktoren ergibt:

$$= \left( \mathbf{I} - \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right)^{-1} \right) \mathbf{K}_{UX}\mathbf{W}^{-1} \quad (\text{A.24})$$

$$= \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} - \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right) \cdot \quad (\text{A.25})$$

$$\begin{aligned} & \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right)^{-1} \mathbf{K}_{UX}\mathbf{W}^{-1} \\ & = \mathbf{K}_{UU} \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right)^{-1} \mathbf{K}_{UX}\mathbf{W}^{-1}. \end{aligned} \quad (\text{A.26})$$

Für die Kovarianzmatrix (A.17) ergibt sich durch Anwendung obiger Gleichung

$$\mathbf{K}_{UU} - \mathbf{K}_{UX} \left( \mathbf{K}_{XU}\mathbf{K}_{UU}^{-1}\mathbf{K}_{UX} + \mathbf{W} \right)^{-1} \mathbf{K}_{XU} \quad (\text{A.27})$$

$$= \mathbf{K}_{UU} - \mathbf{K}_{UX} \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right)^{-1} \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU}. \quad (\text{A.28})$$

Ausklammern von  $\mathbf{K}_{UU}$ , des Terms  $(\mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU})^{-1}$  und weitere Vereinfachung der Faktoren ergibt

$$= \mathbf{K}_{UU} \left( \mathbf{I} - \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right)^{-1} \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right) \quad (\text{A.29})$$

$$= \mathbf{K}_{UU} \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right)^{-1} \cdot \quad (\text{A.30})$$

$$\begin{aligned} & \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} - \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right) \\ & = \mathbf{K}_{UU} \left( \mathbf{K}_{UU} + \mathbf{K}_{UX}\mathbf{W}^{-1}\mathbf{K}_{XU} \right)^{-1} \mathbf{K}_{UU} \end{aligned} \quad (\text{A.31})$$

und zusammenfassend erhält man schließlich für den Erwartungswert und die Kovarianzmatrix

$$\boldsymbol{\mu}_a = \boldsymbol{\mu}_U + \mathbf{K}_{UU} \mathbf{V}^{-1} \mathbf{d} \quad (\text{A.32})$$

$$\boldsymbol{\Sigma}_a = \mathbf{K}_{UU} \mathbf{V}^{-1} \mathbf{K}_{UU} \quad (\text{A.33})$$

$$\mathbf{V} = \mathbf{K}_{UU} + \mathbf{K}_{UX} \mathbf{W}^{-1} \mathbf{K}_{XU} \quad (\text{A.34})$$

$$\mathbf{d} = \mathbf{K}_{UX} \mathbf{W}^{-1} (\mathbf{q} - \boldsymbol{\mu}_X) . \quad (\text{A.35})$$



# Abbildungsverzeichnis

1.1	Abhängigkeit des Messbereichs in der Deflektometrie von der Oberfläche. (a), (c), (e) Darstellung der Szene mit unterschiedlichen Oberflächen Kugel, Ausschnitt des Inneren einer Kugel und einer gewellten Oberfläche. (b), (d), (f) zugehöriges von der Kamera beobachtetes Bild. . . . .	3
2.1	Schematische Darstellung der direkten Abhängigkeiten zwischen Zuständen, Beobachtungen und Aktionen in einem POMDP.	16
2.2	Direkte Abhängigkeiten von Aktionen $a_k$ , Beobachtung $z_{k+1}$ und der Oberfläche $S$ bei der Oberflächeninspektion. . . . .	21
3.1	Unschärfescheibe auf dem Kamerasensor. . . . .	39
3.2	Unschärfescheibe auf der Oberfläche . . . . .	40
3.3	Bildschirmmodell . . . . .	43
3.4	Schematische Darstellung der Lage eines Sensors relativ zu einer Oberfläche, dargestellt sind die Koordinatensysteme des Bildschirms, der Kamera und das lokale Koordinatensystem am Sichtpunkt $\chi$ . . . . .	46
3.5	Schematische Darstellung zur Bestimmung des Normalenvektors aus einer deflektometrischen Messung . . . . .	50
3.6	Mathematisches Modell der Bildentstehung durch den EMVA 1288 Standard [EMV10]. . . . .	58
3.7	(a) Schätzung des Phasenrauschens. (b) Schnittlinie bei $y = 450$ und Vergleich der Varianzen geschätzt durch die Methode von Fischer und einer Schätzung mittels einer Stichprobe. . . . .	61

---

3.8	Ergebnis einer Schätzung mittels Gaußmischverteilung. Die Wellenlängen der Muster für 3, 7 und 17 Wiederholungen betragen jeweils 640, circa 274 und circa 112 Pixel. . . . .	63
3.9	Dargestellt sind beispielhaft (a) der simulierte Lichtpfad und (b) die Standardabweichung der gemessenen Normalenvektoren aus der Simulation für eine konvexe Oberfläche. . . . .	68
3.10	Dargestellt sind beispielhaft (a) der Sichtwinkel der Kamera auf die Oberfläche und (b) die Größe der Unschärfescheibe auf der Oberfläche aus der Simulation für eine konvexe Oberfläche. . .	69
4.1	Schematische Darstellung der Unsicherheitsreduktion eines Gauß-Prozesses durch Beobachtungen. Die Messpunkte sind dargestellt mit einem $1\sigma$ -Streuungsbalken. Die Unsicherheit der Schätzung ist durch den hellblauen Bereich abgebildet, der die $1\sigma$ -Umgebung der Marginaldichten von dem Funktionswert $f(x)$ an jeder Stelle darstellt. . . . .	76
4.2	Rekonstruierter Ausschnitt einer Kugeloberfläche anhand eines Gaußprozesses mittels Höhenwerten und Neigungen. Farblich dargestellt sind (a) der absolute Fehler und (b) die Varianz der Schätzung in logarithmischer Skala. . . . .	92
4.3	Absoluter Fehler der Schätzung mittels Gauß-Prozess und ohne Referenzoberfläche in logarithmischer Skala nach zehn Iterationen.	95
4.4	Mittlerer absoluter und maximaler Fehler der Schätzung mittels Gauß-Prozess ohne Referenzoberfläche in Abhängigkeit des Iterationsschritts. . . . .	96
4.5	Rekonstruierter Ausschnitt einer Kugeloberfläche anhand eines Gauß-Prozesses mit Punkten und Gradienten ohne Referenzoberfläche. Zu sehen ist (a) der Erwartungswert der Oberflächenschätzung mit nur Höhendaten und (b) nach einer Iteration.	97
4.6	Darstellung des Approximationsfehlers $e_{\mathcal{D}}$ in Abhängigkeit des Gitterabstandes $T$ für Defekte mit $L_d = 10, 20, \dots, 60$ . . . . .	100

---

4.7	Diskretisierung einer Oberfläche mit Gauß-Delle im Zentrum anhand der beiden vorgestellten Kriterien. . . . .	105
5.1	(a) Varianzverteilung nach 11 Messungen. Große unzusammenhängende Bereiche auf der Oberfläche haben eine niedrige Varianz. Die Wahrscheinlichkeit, dass die Pertubationsfunktion eine geeignete Pertubation durchführt, wird mit steigender Anzahl der Messungen niedriger. (b) Zufällige Menge von $\chi$ , erzeugt durch einen Sampler, der unabhängig von der aktuellen Sensorkonfiguration proportional zur Varianz zieht. . . . .	119
5.2	Stichproben der besten relativen Varianz und die dafür benötigte Anzahl an Schritten, berechnet aus mehreren Optimierungsläufen jeweils mit der zufälligen Pertubation der Richtung, der Pertubation per Regeln mit homogenen und optimierten Gewichten. . . . .	122
5.3	Aufwand der Optimierung: (a) zeigt die Anzahl der Optimierungen für die einzelnen Regionen und (b) das korrespondierende Histogramm über die Anzahl der Optimierungen. . . . .	129
6.1	Ebene Testoberfläche für die simulationsbasierte Evaluation der Planungsalgorithmen . . . . .	137
6.2	Testoberflächen für die simulationsbasierte Evaluation der Planungsalgorithmen . . . . .	138
6.3	Durchschnittlicher Fehler für die Planung mittels Gauß-Prozess Modell bei Variation der Gütemaße. . . . .	141
6.4	Verlauf der relativen Varianz der mittels Greedy-Planung und Gauß-Prozess Oberflächenmodell gefundenen Pläne für die Testoberflächen. . . . .	143
6.5	Verlauf der relativen Varianz der mittels Sweep-Planung und Gauß-Prozess Oberflächenmodell gefundenen Pläne für die Testoberflächen. . . . .	143

6.6	Verlauf der relativen Varianz der mittels Greedy-Planung und diskretem Modell gefundenen Pläne für die Testoberflächen. .	146
6.7	Verlauf der relativen Varianz der mittels Sweep-Planung und diskretem Modell gefundenen Pläne für die Testoberflächen. .	146
6.8	Schematische Darstellung der Defekte bei variierender Ausdehnung (von 0.03 mm bis 1 mm in 0.03 mm Schritten) und konstanter maximaler Neigung ( $20^\circ$ ). Von oben nach unten: Wahre Neigungsabweichung des Defekts in logarithmischer Skala, geschätzte Neigungsabweichung in logarithmischer Skala, Anwendung eines Schwellenwertfilters mit jeweils $0.1^\circ$ , $0.2^\circ$ und $0.3^\circ$ . .	148
6.9	Schematische Darstellung der Defekte bei variierender maximaler Neigungsabweichung (von $0.116^\circ$ bis $1.4^\circ$ in $0.116^\circ$ Schritten) und konstanter Ausdehnung von 3 mm. Von oben nach unten: Wahre Neigungsabweichung des Defekts in logarithmischer Skala, geschätzte Neigungsabweichung in logarithmischer Skala, Anwendung eines Schwellenwertfilters mit jeweils $0.1^\circ$ , $0.2^\circ$ und $0.3^\circ$ . . . . .	148
6.10	(a) Unzureichend beobachteter Oberflächenanteil und (b) Berechnungszeit der approximativ berechneten Pläne für das diskrete Modell. . . . .	152
6.11	(a) Maximale Varianz und (b) Berechnungszeit der approximativ berechneten Pläne für den Gauß-Prozess. . . . .	153
6.12	Das zu untersuchende Karosserieteil ist eine Motorhaube für den Renault Twingo. . . . .	155
6.13	Messaufbau bestehend aus dem Industrieroboter KUKA KR 60 HA, kompaktem deflektometrischen Sensorkopf und Testoberfläche. . . . .	155
6.14	Nahaufnahme des verwendeten kompakten deflektometrischen Sensorkopfs bestehend aus Kamera und Bildschirm. Angezeigt wird ein Hilfsmuster zur Kalibrierung von Schirm-zu-Kamera. .	156

---

6.15	Darstellung der relevanten Koordinatensysteme und zu bestimmenden Transformationen zwischen diesen für die Kalibrierung des deflektometrischen Messsystems und Versuchsaufbaus. . .	158
6.16	Varianz nach der Messung mit dem zufälligen Startplan des Sweep-Verfahrens für einen Planlänge von 40. . . . .	166
6.17	Verlauf der durchschnittlichen relativen Varianz für Planlängen 30, 35, 39, 45, 50 und 55 initiiert durch die Gridsuche/Sweep-Planung und der Greedy-Planung . . . . .	167
6.18	(a) Messbereiche des vom Sweep-Verfahren berechneten optimalen Plans. Zur besseren Erkennung wurden die Konturen geglättet und in unterschiedlichen Farben dargestellt. (b) Messbereiche der real ausgeführten Messung. . . . .	168
6.19	Defektpositionen auf der Oberfläche . . . . .	170
6.20	Schnitt entlang der Zeilen durch die deflektometrische Registrierung an den jeweiligen Defektpositionen. . . . .	171
6.21	Defekte $D_D$ und $D_E$ sind in den Streifenbildern erkennbar. . .	172
6.22	Defekte $D_A$ , $D_C$ sind in den Streifenbildern erkennbar. . . .	173



# Tabellenverzeichnis

5.1	Statistische Merkmale der besten relativen Varianz und die dafür benötigte Anzahl an Schritten, berechnet aus mehreren Optimierungsläufen jeweils mit der zufälligen Perturbation der Richtung, der Perturbation per Regeln mit homogenen und optimierten Gewichten. . . . .	121
6.1	Zusammenfassung der Ergebnisse der Greedy-Planung und der Sweep-Planung mit Gauß-Prozess Modell für die jeweiligen Testoberflächen. Aufgelistet sind das Planungsverfahren (PV), die Oberfläche, die optimale gefundene Planlänge $n^*$ , die erreichte maximale Standardabweichung $\sigma_{\max}$ sowie die totale Zeit für die Planung mit Suche im sequentiellen $t_{\text{seq}}$ und parallelen $t_{\text{par}}$ Fall. Zur Berechnung der Pläne wurde ein Rechner mit 16 GB RAM und zwei Intel Xeon E5645 Prozessoren mit einem Prozessortakt von 2.4 Ghz und je 6 Kernen verwendet. .	144
6.2	Trefferquote (T) und Genauigkeit (G) für die Inspektion der unterschiedlichen Testoberflächen in Abhängigkeit des Schwellenwertes. . . . .	145

6.3 Zusammenfassung der Ergebnisse der Greedy- und Sweep-Planung mit dem diskreten Modell für die jeweiligen Testoberflächen. Aufgelistet sind das Planungsverfahren (PV), die Oberfläche, die optimale gefundene Planlänge  $n^*$ , die erreichte maximale Standardabweichung  $\sigma_{\max}$  sowie die totale Zeit für die Planung mit Suche im sequentiellen  $t_{\text{seq}}$  und parallelen  $t_{\text{par}}$  Fall. Zur Berechnung der Pläne wurde ein Rechner mit 16 GB RAM und zwei Intel Xeon E5645 Prozessoren mit einem Prozessortakt von 2.4 Ghz und je 6 Kernen verwendet. . . . . 147

6.4 Trefferquote (T) und Genauigkeit (G) für die Inspektion der unterschiedlichen Testoberflächen mit den Defektarten Ausdehnung und Neigung in Abhängigkeit des Schwellenwertes. . . . 150

6.5 Ergebnisse der Gridsuche/Sweep-Planung. Die als optimal bestimmte Planlänge beträgt 40. . . . . 167



# Literaturverzeichnis

- [Ara10] ARAYA, Mauricio; BUFFET, Olivier; THOMAS, Vincent und CHARPILLET, François: A POMDP Extension with Belief-dependent Rewards, in: J. D. Lafferty; C. K. I. Williams; J. Shawe-Taylor; R. S. Zemel und A. Culotta (Herausgeber) *Advances in Neural Information Processing Systems 23*, Curran Associates, Inc. (2010), S. 64–72
- [Bal08] BALZER, Jonathan: *Regularisierung des Deflektometrieproblems - Grundlagen und Anwendung*, Dissertation, Universität Karlsruhe (TH) (2008)
- [Bal14] BALZER, Jonathan; FELIZ, Daniel Acevedo; SOATTO, Stefano; HÖFER, Sebastian; HADWIGER, Markus und BEYERER, Jürgen: Cavlectometry: Towards Holistic Reconstruction of Large Mirror Objects., in: *3DV*, IEEE Computer Society, S. 448–455
- [Ber17] BERTSEKAS, Dimitri P.: *Dynamic Programming and Optimal Control*, Athena Scientific, 4th Aufl. (2017)
- [Bey15] BEYERER, Jürgen; LEÓN, Fernando Puente und FRESE, Christian: *Machine Vision: Automated Visual Inspection Theory, Practice and Applications*, Springer Publishing Company, Incorporated, 1st Aufl. (2015)

- [Bos92] BOSER, Bernhard E.; GUYON, Isabelle M. und VAPNIK, Vladimir N.: A Training Algorithm for Optimal Margin Classifiers, in: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, ACM, New York, NY, USA, S. 144–152
- [Box65] BOX, M. J.: A New Method of Constrained Optimization and a Comparison With Other Methods. *The Computer Journal* (1965), Bd. 8(1): S. 42–52
- [Bri00] BRIGGS, A. J.; SCHARSTEIN, D.; BRAZIUNAS, D.; DIMA, C. und WALL, P.: Mobile robot navigation using self-similar landmarks, in: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Bd. 2, S. 1428–1434
- [Che08] CHEN, Shengyong; LI, Y. F.; ZHANG, Jianwei und WANG, Wanliang: *Active Sensor Planning for Multiview Vision Tasks*, Springer Publishing Company, 1st Aufl. (2008)
- [Dan00] DANNER, T. und KAVRAKI, L. E.: Randomized planning for short inspection paths, in: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Bd. 2, S. 971–976
- [Dra11] DRAGIEV, S.; TOUSSAINT, M. und GIENGER, M.: Gaussian process implicit surfaces for shape estimation and grasping, in: *2011 IEEE International Conference on Robotics and Automation*, S. 2845–2850
- [Dun09a] DUNN, Enrique; BERG, Jur van den und FRAHM, Jan-Michael: Developing visual sensing strategies through next best view planning, in: *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, IROS'09, IEEE Press,

---

Piscataway, NJ, USA, S. 4001–4008

- [Dun09b] DUNN, Enrique und FRAHM, Jan-Michael: Next Best View Planning for Active Model Improvement, in: *British Machine Vision Conference, BMVC 2009 - Proceedings*, British Machine Vision Association, BMVA, S. 1–11
- [EMV10] EMVA Standard 1288 - Standard for Characterization of Image Sensors and Cameras, Release 3.0 (2010)
- [EPB82] E. P. BOX, George: Choice of Response Surface Design and Alphabetic Optimality. *Utilitas Mathematica (Canada)* (1982), Bd. 21B: S. 11–55
- [Fas05] FASSI, Irene und LEGNANI, Giovanni: Hand to sensor calibration: A geometrical interpretation of the matrix equation  $AX=XB$ . *Journal of Robotic Systems* (2005), Bd. 22(9): S. 497–506, URL <http://dx.doi.org/10.1002/rob.20082>
- [Fis12] FISCHER, Marc; PETZ, Marcus und TUTSCH, Rainer: Vorhersage des Phasenrauschens in optischen Messsystemen mit strukturierter Beleuchtung. *tm - Technisches Messen Heft* (2012), Bd. 79(10): S. 451–458
- [For12] FORSYTH, David A. und PONCE, Jean: *Computer Vision - A Modern Approach, Second Edition.*, Pitman (2012)
- [Gen11] GENG, JASON: Structured-light 3D surface imaging: a tutorial. *Adv. Opt. Photon.* (2011), Bd. 3(2): S. 128–160
- [Gui68] GUIN, J. A.: Modification of the Complex Method of Constrained Optimization. *The Computer Journal* (1968), Bd. 10(4): S. 416–417
- [Hau11] HAUSCHILD, Mark und PELIKAN, Martin: An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation* (2011), Bd. 1(3): S. 111–128

- [Hei97] HEIKKILA, J. und SILVEN, O.: A four-step camera calibration procedure with implicit image correction, in: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, S. 1106–1112
- [HL88] HO-LE, K.: Finite element mesh generation methods: a review and classification. *Computer-Aided Design* (1988), Bd. 20(1): S. 27–38
- [Höf13] HÖFER, Sebastian; ROSCHANI, Masoud und WERLING, Stefan: Pattern coding strategies for deflectometric measurement systems, in: *Proceedings of SPIE Volume 8791, Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection*, München, Deutschland, S. 8791 – 8791 – 11
- [Höf16] HÖFER, Sebastian und BEYERER, Jürgen: Scannende Infrarotdeflektometrie für die Inspektion diffus spiegelnder Oberflächen, in: Fernando Puente León und Michael Heizmann (Herausgeber) *tm - Technisches Messen*, Bd. 83, S. 374–385
- [Höf17] HÖFER, Sebastian: *Untersuchung diffus spiegelnder Oberflächen mittels Infrarotdeflektometrie*, Dissertation, Karlsruher Institut für Technologie (KIT) (2017)
- [Hol12] HOLLINGER, G. A.; ENLOT, B.; HOVER, F.; MITRA, U. und SUKHATME, G. S.: Uncertainty-driven view planning for underwater inspection, in: *2012 IEEE International Conference on Robotics and Automation*, S. 4884–4891
- [Hor07] HORBACH, Jan: *Verfahren zur optischen 3D-Vermessung spiegelnder Oberflächen*, Dissertation, Universität Karlsruhe (TH) (2007)
- [Hub09] HUBER, Marco: *Probabilistic Framework for Sensor Management*, Dissertation, Universität Karlsruhe (TH) (2009)

- [Its14] Itseez: *The OpenCV Reference Manual*, 2.4.9.0 Aufl. (2014)
- [Its15] ITSEEZ: Open Source Computer Vision Library, <https://github.com/itseez/opencv> (2015)
- [Kae98] KAEHLING, Leslie Pack; LITTMAN, Michael L. und CASSANDRA, Anthony R.: Planning and acting in partially observable stochastic domains. *Artificial Intelligence* (1998), Bd. 101(1): S. 99–134
- [Kam04] KAMMEL, Sören: *Deflektometrische Untersuchung spiegelnd reflektierender Freiformflächen*, Dissertation, Universität Karlsruhe (TH), Universitätsverlag Karlsruhe (2004)
- [Kir83] KIRKPATRICK, S.; GELATT, C. D. und VECCHI, M. P.: Optimization by Simulated Annealing. *Science* (1983), Bd. 220(4598): S. 671–680
- [Kus06] KUSS, Malte und RASMUSSEN, Carl E.: Assessing Approximations for Gaussian Process Classification, in: Y. Weiss; B. Schölkopf und J. C. Platt (Herausgeber) *Advances in Neural Information Processing Systems 18*, MIT Press (2006), S. 699–706
- [Lap92] LAPORTE, Gilbert: The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* (1992), Bd. 59(2): S. 231–247, URL <https://ideas.repec.org/a/eee/ejores/v59y1992i2p231-247.html>
- [Liu11] LIU, S. und COOPER, D. B.: A complete statistical inverse ray tracing approach to multi-view stereo, in: *CVPR 2011*, S. 913–920
- [Liu15] LIU, Miaomiao; HARTLEY, Richard und SALZMANN, Mathieu: Mirror Surface Reconstruction from a Single Image. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (2015), Bd. 37(4): S. 760–773

- [Lob13] LOBACHEV, Oleg; SCHMIDT, Martin und GUTHE, Michael: Optimizing Multiple Camera Positions for the Deflectometric Measurement of Multiple Varying Targets. *Journal of WSCG* (2013), Bd. 21: S. 145–152
- [Mur00] MURPHY, Kevin P.: A Survey of POMDP Solution Techniques, Techn. Ber., Michigan Tech (2000)
- [Neg12] NEGARA, Christian und ROSCHANI, Masoud: Effiziente probabilistische B-Spline-Oberflächenrekonstruktion durch Verwendung eines Informationsfilters, in: Fernando Puente León und Michael Heizmann (Herausgeber) *Forum Bildverarbeitung 2012*, Karlsruhe: KIT Scientific Publishing, Aachen, S. 131–142
- [Nel65] NELDER, J. A. und MEAD, R.: A Simplex Method for Function Minimization. *The Computer Journal* (1965), Bd. 7(4): S. 308–313
- [Pak13] PAK, Alexey: Reconstruction of specular surfaces via probabilistic voxel carving. *Proc. SPIE* (2013), Bd. 8791: S. 8791 – 8791 – 8
- [Pak14] PAK, Alexey: Stability of absolute depth reconstruction from deflectometric measurement data, in: *Proc. SPIE 9203, Interferometry XVII: Techniques and Analysis*, Bd. 9203, San Diego, California, USA, S. 9203 – 9203 – 7
- [Pak16] PAK, Alexey: The concept and implementation of smooth generic camera calibration. *Proc. SPIE* (2016), Bd. 9960: S. 9960 – 9960 – 10
- [Par94] PARK, F. C. und MARTIN, B. J.: Robot sensor calibration: solving  $AX=XB$  on the Euclidean group. *IEEE Transactions on Robotics and Automation* (1994), Bd. 10(5): S. 717–721
- [Per97] PERARD, Denis und BEYERER, Jürgen: Three-dimensional measurement of specular free-form surface with a structured-lighting

- reflection technique, in: *Proc. SPIE 3204, Three-Dimensional Imaging and Laser-based Systems for Metrology and Inspection III*, Bd. 3204, International Society for Optics and Photonics, S. 74–81
- [Pet06] PETZ, M.: *Rasterreflexions-Photogrammetrie - Ein neues Verfahren zur geometrischen Messung spiegelnder Oberflächen*, Dissertation, Technische Universität Braunschweig, Shaker Verlag, Aachen (2006)
- [Pet12] PETERSEN, K. B. und PEDERSEN, M. S.: *The Matrix Cookbook* (2012)
- [QC05] QUIÑONERO-CANDELA, Joaquin und RASMUSSEN, Carl Edward: A Unifying View of Sparse Approximate Gaussian Process Regression. *J. Mach. Learn. Res.* (2005), Bd. 6: S. 1939–1959
- [Rap11] RAPP, Holger H.: *Reconstruction of Specular Reflective Surfaces using Auto-Calibrating Deflectometry*, Dissertation, Karlsruher Institut für Technologie (KIT) (2011)
- [Ras05] RASMUSSEN, Carl E. und WILLIAMS, Christopher K. I.: *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*, The MIT Press (2005)
- [Rat95] RATHJEN, C.: Statistical properties of phase-shift algorithms. *Journal of the Optical Society of America A* (1995), Bd. 12(9): S. 1997–2008
- [Rod10] RODRIGUES, Rui; BARRETO, João P. und NUNES, Urbano: Camera Pose Estimation Using Images of Planar Mirror Reflections, in: Kostas Daniilidis; Petros Maragos und Nikos Paragios (Herausgeber) *Computer Vision – ECCV 2010*, Springer Berlin Heidelberg, Berlin, Heidelberg, S. 382–395

- [Ros12] ROSCHANI, Masoud und BEYERER, Jürgen: Planungsbasierte Oberflächeninspektion in der Deflektometrie bei gegebener Referenzfläche mittels Greedy-Optimierung, in: Prof. Dr.-Ing. Robert Schmitt (Herausgeber) *Tagungsband des XXVI. Messtechnisches Symposiums*, Shaker, Aachen, S. 131–142
- [Ros13] ROSCHANI, Masoud und BEYERER, Jürgen: Planungsbasierte Oberflächeninspektion in der Deflektometrie mittels Greedy-Optimierung. *tm Technisches Messen* (2013), Bd. 80(6): S. 189–195
- [Ros14] ROSCHANI, Masoud: Optimizing Deflectometric Measurements for Visibility, in: Jürgen Beyerer und Alexey Pak (Herausgeber) *Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*, Nr. IES-2013-09 in *Karlsruher Schriften zur Anthropomatik*, KIT Scientific Publishing, S. 101–111
- [Rue05] RUE, Havard und HELD, Leonhard: *Gaussian Markov Random Fields: Theory And Applications (Monographs on Statistics and Applied Probability)*, Chapman & Hall/CRC (2005)
- [Sal04] SALVI, Joaquim; PAGÈS, Jordi und BATLLE, Joan: Pattern Codification Strategies in Structured Light Systems. *PATTERN RECOGNITION* (2004), Bd. 37: S. 827–849
- [Sär11] SÄRKKÄ, Simo: Linear operators and stochastic partial differential equations in Gaussian process regression, in: *Proceedings of the 21st international conference on Artificial neural networks - Volume Part II*, ICANN'11, Springer-Verlag, Berlin, Heidelberg, S. 151–158
- [Sch99] SCHÖLKOPF, Bernhard; SMOLA, Alexander und MÜLLER, Klaus-Robert: Kernel principal component analysis, in: *ADVANCES IN*



- 
- KERNEL METHODS - SUPPORT VECTOR LEARNING*, MIT Press, S. 327–352
- [Sch01] SCHÖLKOPF, Bernhard und SMOLA, Alexander J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, USA (2001)
- [Sco03] SCOTT, William R.; ROTH, Gerhard und RIVEST, Jean-François: View Planning for Automated Three-dimensional Object Reconstruction and Inspection. *ACM Comput. Surv.* (2003), Bd. 35(1): S. 64–96
- [See03] SEEGER, Matthias; WILLIAMS, Christopher K. I. und LAWRENCE, Neil D.: Fast Forward Selection to Speed Up Sparse Gaussian Process Regression, in: *IN WORKSHOP ON AI AND STATISTICS 9*
- [Sha49] SHANNON, C. E.: Communication in the Presence of Noise. *Proceedings of the IRE* (1949), Bd. 37(1): S. 10–21
- [Shi89] SHIU, Y. C. und AHMAD, S.: Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form  $AX=XB$ . *IEEE Transactions on Robotics and Automation* (1989), Bd. 5(1): S. 16–29
- [Shr09] SHRIHARI, Vasudevan; FABIO, Ramos; ERIC, Nettleton und HUGH, Durrant-Whyte: Gaussian process modeling of large-scale terrain. *Journal of Field Robotics* (2009), Bd. 26(10): S. 812–840
- [Sma73] SMALLWOOD, Richard D. und SONDIK, Edward J.: The Optimal Control of Partially Observable Markov Processes over a Finite Horizon. *Operations Research* (1973), Bd. 21(5): S. 1071–1088
- [Smi10] SMITH, Mike; POSNER, Ingmar und NEWMAN, Paul: Generating Implicit Surfaces from Lidar Data, in: *Towards Autonomous Robotic Systems*, Plymouth, UK, 08

- [Sne06] SNELSON, Edward und GHARAMANI, Zoubin: Sparse Gaussian Processes using Pseudo-inputs, in: *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, MIT press, S. 1257–1264
- [Son78] SONDIK, Edward J.: The Optimal Control of Partially Observable Markov Processes over the Infinite Horizon: Discounted Costs. *Operations Research* (1978), Bd. 26(2): S. 282–304
- [ST04] SHAWE-TAYLOR, John und CRISTIANINI, Nello: *Kernel Methods for Pattern Analysis*, Cambridge University Press, New York, NY, USA (2004)
- [Str06] STROBL, K. H. und HIRZINGER, G.: Optimal Hand-Eye Calibration, in: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, S. 4647–4653
- [Sur97] SURREL, Yves: Additive noise effect in digital phase detection. *Applied Optics* (1997), Bd. 36(1): S. 271–276
- [Tar95] TARABANIS, K. A.; ALLEN, P. K. und TSAI, R. Y.: A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation* (1995), Bd. 11(1): S. 86–104
- [Tsa89] TSAI, R. Y. und LENZ, R. K.: A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation* (1989), Bd. 5(3): S. 345–358
- [Wen07] WENHARDT, Stefan; DEUTSCH, Benjamin; ANGELOPOULOU, Elli und NIEMANN, Heinrich: Active Visual Object Reconstruction using D-, E-, and T-Optimal Next Best Views. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* (2007), Bd. 0: S. 1–7

- [Wer07] WERLING, Stefan; BALZER, Jonathan und BEYERER, Jürgen: A New Approach for Specular Surface Reconstruction Using Deflectometric Methods, in: *37. Jahrestagung der Gesellschaft für Informatik e.V. (GI): INFORMATIK 2007 - Informatik trifft Logistik*, Bd. 1, S. 44–48
- [Wer11] WERLING, Stefan Bruno: *Deflektometrie zur automatischen Sichtprüfung und Rekonstruktion spiegelnder Oberflächen*, Dissertation, Karlsruher Institut für Technologie (KIT) (2011)
- [Wha90] WHAITE, P. und FERRIE, F. P.: From uncertainty to visual exploration, in: *[1990] Proceedings Third International Conference on Computer Vision*, S. 690–697
- [Whi85] WHITE, Douglas J.: Real Applications of Markov Decision Processes. *Interfaces* (1985), Bd. 15(6): S. 73–83
- [Whi88] WHITE, D. J.: Further Real Applications of Markov Decision Processes. *Interfaces* (1988), Bd. 18(5): S. 55–61
- [Whi93] WHITE, D. J.: A SURVEY OF APPLICATIONS OF MARKOV DECISION-PROCESSES. *Journal of the Operational Research Society* (1993), Bd. 44(11): S. 1073–1096
- [Wil01] WILLIAMS, Christopher und SEEGER, Matthias: Using the Nyström Method to Speed Up Kernel Machines, in: *Advances in Neural Information Processing Systems 13*, MIT Press, S. 682–688
- [Wil07] WILLIAMS, Oliver und FITZGIBBON, Andrew: Gaussian Process Implicit Surfaces, in: *Gaussian Processes In Practice*
- [Zha99] ZHANG, RUO; TSAI, Ping-Sing; CRYER, J. E. und SHAH, M.: Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1999), Bd. 21(8): S. 690–706

- [Zha00] ZHANG, Zhengyou: A Flexible New Technique for Camera Calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* (2000), Bd. 22(11): S. 1330–1334
- [Zie14] ZIEBARTH, Mathias; HEIZMANN, Michael und BEYERER, Jürgen: Sichtbarkeit von Dellen und Beulen auf spiegelnden Oberflächen, in: Fernando Puente León und Michael Heizmann (Herausgeber) *Forum Bildverarbeitung*, KIT Scientific Publishing, Regensburg, S. 153–165
- [Zie15] ZIEBARTH, Mathias: Defect perception thresholds on specular surfaces, in: Jürgen Beyerer und Alexey Pak (Herausgeber) *Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory*, Karlsruher Schriften zur Anthropomatik, KIT Scientific Publishing, S. 123–132





**SCHRIFTENREIHE AUTOMATISCHE SICHTPRÜFUNG UND BILDVERARBEITUNG  
(ISSN 1866-5934)**

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

---

- Band 1     **JONATHAN BALZER**  
Regularisierung des Deflektometrieproblems Grundlagen  
und Anwendung. 2008  
ISBN 978-3-86644-230-6
- Band 2     **IOANA GHEȚA**  
Fusion multivariater Bildserien am Beispiel eines Kamera-Arrays. 2011  
ISBN 978-3-86644-684-7
- Band 3     **STEFAN BRUNO WERLING**  
Deflektometrie zur automatischen Sichtprüfung  
und Rekonstruktion spiegelnder Oberflächen. 2011  
ISBN 978-3-86644-687-8
- Band 4     **JAN WASSENBERG**  
Efficient Algorithms for Large-Scale Image Analysis. 2012  
ISBN 978-3-86644-786-8
- Band 5     **MARTIN GRAFMÜLLER**  
Verfahrensfortschritte in der robusten Echtzeiterkennung  
von Schriftzeichen. 2013  
ISBN 978-3-86644-979-4
- Band 6     **JÜRGEN BRAUER**  
Human Pose Estimation with Implicit Shape Models. 2014  
ISBN 978-3-7315-0184-8
- Band 7     **MARKUS MÜLLER**  
Szeneninterpretation unter Verwendung multimodaler Sensorik  
und Salienzmaßen. 2017  
ISBN 978-3-7315-0240-1
- Band 8     **ROBIN GRUNA**  
Beleuchtungsverfahren zur problemspezifischen Bildgewinnung  
für die automatische Sichtprüfung. 2015  
ISBN 978-3-7315-0313-2
- Band 9     **THOMAS STEPHAN**  
Beitrag zur Unterwasserbildrestauration. 2017  
ISBN 978-3-7315-0579-2

- Band 10 **JAN-PHILIP JARVIS**  
A Contribution to Active Infrared Laser Spectroscopy  
for Remote Substance Detection. 2017  
ISBN 978-3-7315-0725-3
- Band 11 **MIRO TAPHANEL**  
Chromatisch konfokale Triangulation – Hochgeschwindigkeits 3D-Sensorik  
auf Basis der Wellenlängenschätzung mit optimierten Filtern. 2018  
ISBN 978-3-7315-0646-1
- Band 12 **SEBASTIAN HÖFER**  
Untersuchung diffus spiegelnder Oberflächen mittels  
Infrarotdeflektometrie. 2017  
ISBN 978-3-7315-0711-6
- Band 13 **MATTHIAS RICHTER**  
Über lernende optische Inspektion am Beispiel der  
Schüttgutsortierung. 2018  
ISBN 978-3-7315-0842-7
- Band 14 **MATHIAS ZIEBARTH**  
Wahrnehmungsgrenzen kleiner Verformungen auf  
spiegelnden Oberflächen. 2019  
ISBN 978-3-7315-0890-8
- Band 15 **JOHANNES MEYER**  
Light Field Methods for the Visual Inspection of Transparent Objects. 2019  
ISBN 978-3-7315-0912-7
- Band 16 **MASOUD ROSCHANI**  
Probabilistische Planungsverfahren für die  
deflektometrische Oberflächeninspektion. 2019  
ISBN 978-3-7315-0907-3





Bei der Inspektion von spiegelnden Oberflächen wird die Deflektometrie eingesetzt. Dabei handelt es sich um ein optisches Messverfahren, das sensitiv bezüglich der Neigung der Oberfläche ist. Mittels eines robotergeführten Sensorkopfes können auch große oder komplex geformte Oberflächen mittels mehrerer Messungen inspiziert werden.

Diese Arbeit behandelt die automatisierte Bestimmung der Anzahl der benötigten Messungen sowie die jeweiligen Parameter des Sensorkopfes, wie die Position und Orientierung des Sensors mit dem Ziel eine vollständige Beobachtung der Prüfoberfläche unter Berücksichtigung der Ortsauflösung und Messunsicherheit zu erlangen.

Die Formulierung des Problems erfolgt im Rahmen der probabilistischen sequentiellen Entscheidungstheorie. Damit lassen sich sowohl die Messunsicherheiten als auch das Unwissen über die Oberfläche einheitlich beschreiben. Es werden zwei probabilistische Oberflächenmodelle sowie zwei Planungsverfahren vorgestellt und ihre Eignung für die deflektometrische Oberflächeninspektion untersucht.

