



OTTO VON GUERICKE  
UNIVERSITÄT  
MAGDEBURG

EIT

FAKULTÄT FÜR  
ELEKTROTECHNIK UND  
INFORMATIONSTECHNIK

# Spline-Methoden für die modellbasierte 3D-Oberflächeninspektion

**Dissertation**

zur Erlangung des akademischen Grades

**Doktoringenieur  
(Dr.-Ing.)**

von Dipl.-Ing. Nikolaus Sebastian Freiherr von Enzberg  
geb. am 06. Februar 1984 in Stuttgart

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik  
der Otto-von-Guericke-Universität Magdeburg

**Gutachter:**

apl. Prof. Dr.-Ing. habil. Ayoub Al-Hamadi  
Prof. Dr. rer. nat. Andreas Wendemuth  
Prof. Dr. rer. nat. Gunther Notni

Promotionskolloquium am 18. Dezember 2018



## Zusammenfassung

Die 3D-Oberflächenforminspektion ist insbesondere für komplex geformte und verformbare Bauteile eine anspruchsvolle Aufgabe. Dabei wird die Oberfläche auf kleinste lokale Verformungen wie z. B. Beulen und Dellen untersucht, die im mittleren Mikrometerbereich, und somit unterhalb von Bauteiltoleranzen und -verformungen, aber oberhalb der Oberflächenrauheit liegen.

Die vorliegende Arbeit beschreibt ein automatisches, modellbasiertes Oberflächeninspektionssystem, das auf einem photogrammetrischen Messsystem mit Musterprojektion aufbaut. Klassische Inspektionssysteme sind stark ingenieurmäßig motiviert und bedienen sich Methoden aus der Messtechnik und der digitalen Signalverarbeitung. Der Schwerpunkt dieser Arbeit liegt im Einsatz von Methoden aus der Computergrafik, vornehmlich B-Spline-Oberflächenmodellen und maschinellen Lernverfahren, insbesondere autoassoziative künstliche neuronale Netze (*Assoziativspeicher*).

Entlang der gesamten Verarbeitungskette werden neue Verfahren zur Verbesserung des Inspektionssystems vorgeschlagen: Ein hierarchischer Assoziativspeicher sowie ein spline-basierter Assoziativspeicher als Oberflächenmodell, ein spline-basiertes Registrierungsverfahren sowie eine Multi-Schwellen-Segmentierung.

## Abstract

3D surface inspection of manufactured surfaces is a challenging task, especially for complexly shaped, non-rigid parts. Its purpose is the detection of local surface deformation in the medium micro-meter range, e. g. dents or bumps. The surface defects lie within part tolerances and deformation, but well above surface roughness.

This thesis presents an automated model-based surface inspection system based on a photogrammetric measurement system using pattern projection. Conventional inspection systems are mostly motivated by electrical engineering disciplines as metrology and digital signal processing. This work focuses on the application of techniques borrowed from computer graphics, namely B-spline surface models, as well as machine learning, particularly auto-associative artificial neural networks (*associative memory*).

Novel methods for improvement of the inspection system are proposed alongside the entire processing chain: A hierarchical associative memory as well as a spline-based associative memory as a surface model, a spline-based registration approach as well as a multi-threshold segmentation algorithm.



---

# Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielstellung . . . . .	2
1.3	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Oberflächeninspektion</b>	<b>7</b>
2.1	Definition und Abgrenzung . . . . .	7
2.2	Oberflächeninspektion im Karosseriebau . . . . .	10
2.3	Stand der Technik . . . . .	13
2.3.1	Manuelle Verfahren . . . . .	14
2.3.2	Automatische optische Verfahren . . . . .	17
2.3.3	Nicht-optische Methoden . . . . .	20
<b>3</b>	<b>Ein System zur Automatisierten Oberflächeninspektion</b>	<b>23</b>
3.1	Architektur des Systems . . . . .	23
3.2	Messwertaufnahme . . . . .	26
3.2.1	Photogrammetrische Messverfahren . . . . .	27
3.2.2	Bildkorrelation . . . . .	33
3.2.3	Methoden zur Punktsuche . . . . .	36
3.3	Mathematische Beschreibung von Oberflächen . . . . .	39
3.3.1	Diskrete Beschreibung von Oberflächen . . . . .	39
3.3.2	Analytische Beschreibung von Oberflächen . . . . .	43
3.3.3	Freiformkurven und -flächen . . . . .	48
3.3.4	B-Spline-Flächen . . . . .	54
3.4	Oberflächenmodell und Modellvergleich . . . . .	57
3.4.1	Assoziativspeicher als Oberflächenmodell . . . . .	58
3.4.2	Abstandsberechnung . . . . .	61
3.5	Auswertung von Tiefenkarten . . . . .	62
<b>4</b>	<b>Der Assoziativspeicher als Oberflächenmodell</b>	<b>67</b>
4.1	Mathematische Beschreibung . . . . .	67
4.1.1	Auto-Assoziative Neuronale Netze . . . . .	68

4.1.2	Hauptkomponentenanalyse . . . . .	69
4.1.3	Karhunen-Loève-Transformation . . . . .	71
4.2	Effizientes Training des Assoziativspeichers . . . . .	73
4.2.1	Duale Eigenwertzerlegung . . . . .	74
4.2.2	Ergebnisse . . . . .	76
4.3	Hierarchischer Ansatz für den Assoziativspeicher . . . . .	77
4.3.1	Multiskalen-Analyse . . . . .	78
4.3.2	Beschreibung des entwickelten Ansatzes . . . . .	79
4.3.3	Ergebnisse . . . . .	83
<b>5</b>	<b>Spline-Methoden für die Oberflächeninspektion</b>	<b>91</b>
5.1	B-Spline-Flächen für die Oberflächeninspektion . . . . .	91
5.1.1	Grundlagen der Flächenapproximation . . . . .	91
5.1.2	Flächenapproximation für 3D-Messwerte . . . . .	97
5.2	Spline-basierter Assoziativspeicher . . . . .	100
5.2.1	Orthogonale B-Spline-Basis . . . . .	102
5.2.2	Herleitung der KLT für B-Spline-Basen . . . . .	103
5.2.3	Ergebnisse . . . . .	105
5.3	Spline-basierte Registrierung von Oberflächen . . . . .	107
5.3.1	Problemanalyse . . . . .	109
5.3.2	Beschreibung des entwickelten Ansatzes . . . . .	112
5.3.3	Ergebnisse . . . . .	121
<b>6</b>	<b>Weiterverarbeitung von Defektkarten</b>	<b>125</b>
6.1	Mehr-Schwellen-Segmentierung . . . . .	126
6.2	Merkmale und Klassifikatoren . . . . .	127
6.3	Ergebnisse . . . . .	130
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>133</b>
	<b>Abbildungsverzeichnis</b>	<b>140</b>
	<b>Tabellenverzeichnis</b>	<b>141</b>
	<b>Veröffentlichungen</b>	<b>143</b>
	<b>Literaturverzeichnis</b>	<b>145</b>

---

# Symbolverzeichnis

---

$(x, y, z)$	3D-Weltkoordinaten
$(u, v)$	2D-Bildkoordinaten
$f_{cam}$	Kameramodell
$\mathbf{G}(u, v)$	2D-Bildmatrix (Grauwerte)
$\varphi(u, v)$	Phasenbild
$\mathbf{R}$	Rotationsmatrix
$\mathbf{t}$	Translationsvektor
$c$	Kamerakonstante
$\mathbf{x}$	3D-Punkt
$\mathbf{p}$	3D-Punkt, speziell: Kontrollpunkt
$\mathbf{n}$	Oberflächennormale
$\mathbf{z}$	Tiefenkarte
Index $m$	3D-Punkt oder Tiefenkarte aus einer Messung
Index $s$	3D-Punkt oder Tiefenkarte aus einem Oberflächenmodell
Index $c$	3D-Punkt aus einem Kurvenmodell
$\mathbf{m}$	Maskierungsmatrix
$i, j, k, l$	Punktiteratoren
$\ell$	Iterationstiefe eines Algorithmus'
$n, m$	Dimensionsweise Punktanzahl bzw. Iterationsgrenzen
$N$	Dimensionalität von Daten (= Anzahl Punkte je Messung)
$M$	Anzahl der Messungen (= Punktwolken) in einem Datensatz
$\mathbf{X} = [\mathbf{x}_i]$	Datensatz, Matrix aus Messungen
$\mathbf{W} = [\mathbf{w}_i]$	Matrix der Basisvektoren eines linearen Modells
$\mathbf{C}$	Kovarianzmatrix
$a, b, c, d$	Bereichsgrenzen
$u, v$	Parametrische Oberflächen oder Kurvenkoordinaten
$\mathbf{x}_s(u, v)$	Spline-Fläche
$b^\kappa(u), b^\gamma(v)$	B-Spline Basis $\kappa$ -ten bzw. $\gamma$ -ten Grades
$d$	Punktabstand





---

# Kapitel 1

## Einleitung

---

### 1.1 Motivation

Die stetig fortschreitende Automatisierung in der industriellen Produktion hat in den letzten Jahren eine entscheidende Wende genommen. Lag der Schwerpunkt bisher vor allem in der Automatisierung physischer Arbeiten, können durch neue Technologien zunehmend auch Arbeiten automatisiert werden, die höhere kognitive Leistungen erfordern. Dies geht hin bis zur autonomen Entscheidungsfindung, die einer menschlichen Leistung qualitativ entspricht oder diese übersteigt. Neben der Entlastung bei monotonen, ermüdenden Arbeiten kann somit gleichzeitig die Fehlerquote gesenkt werden.

Ermöglicht wurde diese Entwicklung durch die Konvergenz verschiedener technischer Entwicklungen: Moderne Sensorik ermöglicht eine sehr schnelle, hochaufgelöste Aufnahme physikalischer Daten, die die menschlichen Fähigkeiten weit übersteigen. In noch viel stärkerem Maße ging die Entwicklung der Rechentechnik voran. Dies betrifft neben der reinen Rechenleistung auch Methoden zur hochparallelen Verarbeitung sowie stark gestiegene Speicherkapazitäten und -zugriffsgeschwindigkeiten. Dies erlaubt sowohl die Verarbeitung der immer größer werdenden Datenmengen in Echtzeit, als auch die Anwendung immer komplexerer Algorithmen.

Gemeinsam mit den neuen Möglichkeiten zur Aufnahme großer Datenmengen und der gesteigerten Rechenkapazität haben sich auch die Methoden zur Datenverarbeitung gewandelt. An die Stelle von stark spezialisierten Algorithmen für eingeschränkte Problemfelder sind allgemeingültige Verfahren getreten, die ausgehend von einer großen Menge an Beispieldaten Zusammenhänge selbständig erlernen. Diese Methoden werden unter dem Begriff *Maschinelles Lernen* zusammengefasst. Man spricht - im Gegensatz zu rein analytischen Modellen - auch von *datengetriebener Modellbildung*. Auf Grund der Allgemeingültigkeit dieser Methoden gibt es Überschneidungen sowie parallele und wechselseitige Entwicklungen in verschiedensten Forschungsbereichen und Fachrichtungen, wie z. B. Signalverarbeitung (*Mustererkennung*, engl. Pattern Recognition), Robotik (*Maschinelles Sehen*, engl. Computer Vision), Informatik (*Data-Mining*, *Data Analytics*) oder Statistik

(*Schätzverfahren*) [7]. Das Maschinelle Lernen wird heute auch als einer der wichtigsten Bausteine der Künstlichen Intelligenz gesehen [98].

Aufgrund ihrer stark interdisziplinären Implikationen sind Entwicklungen in der Datenverarbeitung von einer starken Dynamik geprägt und bieten ein enormes Innovationspotential zur Augmentation vorhandener technischer Systeme.

Die vorliegende Arbeit steht an der Schnittstelle zwischen Ingenieurwissenschaften und Informatik. Es betrifft das lange bestehende technische Problem der automatisierten 3D-Oberflächeninspektion, wie sie in der industriellen Produktion, z. B. bei der Qualitätskontrolle von Automobil-Karosserieteilen gefragt wird. Existierende optische 3D-Messsysteme basierend auf Musterprojektion ermöglichen eine hochpräzise Vermessung von Oberflächen, die quantitativ weit über der menschlichen Wahrnehmung liegen. Gleichzeitig kann ein geschulter Auditor aber qualitative Aussagen zur Oberflächengüte treffen, deren *explizite* Modellierung für die maschinelle Beurteilung bislang kaum möglich war.

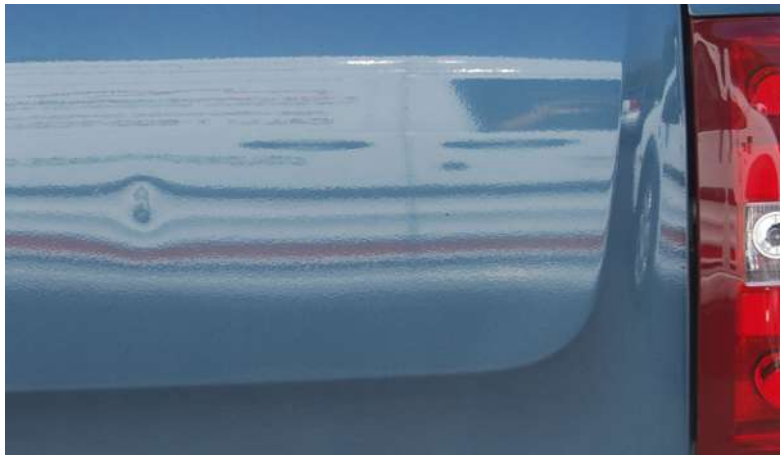
Mögliche Ansätze können aus verschiedenen Teilbereichen der Informatik entnommen werden. Durch die Anwendung von Verfahren aus dem Maschinellen Lernen können „weiche“ menschliche Entscheidungen *implizit* modelliert werden. Dies ermöglicht nicht nur die Lösung des komplexen technischen Problems der Erkennung von Oberflächendefekten, sondern auch die Quantifizierbarkeit dieser bislang menschlichen Entscheidung.

Die Verwendung von spline-basierten Modellen hat sich aufgrund ihrer einfachen, aber dennoch flexiblen Repräsentation von Oberflächen in der Computergrafik und im computergestützten Design (CAD) bewährt. Die Anwendung für Inspektionsanwendungen birgt daher viele Vorteile, die bislang aber noch nicht umfassend untersucht wurden.

Die Beiträge der vorliegenden Dissertationsschrift ergeben sich daher aus der Kombination von Methoden des Maschinellen Lernens, Spline-Methoden der Computergrafik sowie Wissen über die zugrundeliegende 3D-Messtechnik.

## 1.2 Zielstellung

Im Fokus dieser Arbeit liegt die Anwendung der Oberflächeninspektion in der Automobilindustrie zur Bewertung der Oberflächeneigenschaften von Karosserieteilen. Eine automatische Inspektion wird momentan nach Rohbau und Lackierung durchgeführt, da 3D-Oberflächenfehler erst auf glänzenden Oberflächen deutlich sichtbar sind. Das Aussehen von reflektierten Mustern ist abhängig von der Krümmung sowie der Entfernung des Musters und des Beobachters zur Oberfläche [22, 138]. Bereits durch Beulen oder Dellen mit wenigen Mikrometern Höhe werden so sichtbare Störungen in den Reflexionsmustern hervorgerufen (siehe Abbildung 1.1). Folglich werden Beulen aufgrund ihrer geringen Höhe nicht direkt wahrgenommen, sondern erst die von ihnen ausgehende Verzerrung der Reflexionsmuster nach Lackierung der Oberfläche. Für hochpreisige Produkte z. B. aus



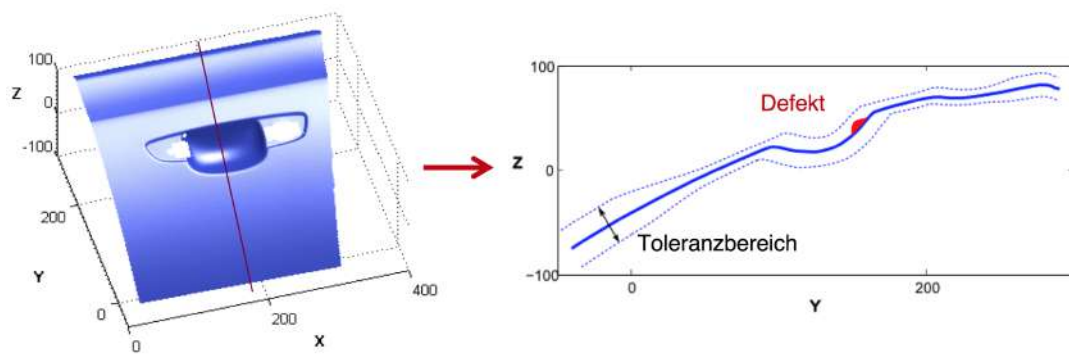
**Abbildung 1.1:** Reflexionsmuster einer Delle an einer Kfz-Hecktür

der Automobilbranche existieren sehr hohe Qualitätsansprüche der Kunden; der erste optische Eindruck beeinflusst die Kaufentscheidung deutlich.

Der Aufwand für die Nacharbeitung von Oberflächenfehlern steigt exponentiell mit den Verarbeitungsschritten (siehe auch Abbildung 2.7). Wünschenswert ist eine Fehlerdetektion noch vor Lackierung und Rohbau, möglichst direkt nach der Blechumformung im Presswerk. Die Bewertung der Oberfläche ist in diesem Zustand besonders herausfordernd, da das Bauteil nicht montiert und somit in seiner Form variabel ist. Weiterhin liegen zulässige Toleranzen der Oberflächenform im Millimeterbereich und somit einige Größenordnungen über den zu detektierenden Fehlerhöhen (siehe Abbildung 1.2). Ohne Lackierung ist außerdem die optische Bewertung von Reflexionsmustern nicht möglich, welche wesentlich zum negativen Eindruck bei der Bewertung des äußeren Erscheinungsbildes führen.

Daher findet an dieser Stelle im Fertigungsprozess von Automobilen aktuell eine manuelle Inspektion durch geschulte Auditoren anhand von Stichproben statt. Eine 100%-Qualitätskontrolle ist somit nicht möglich. Ziel der manuellen Prüfung ist vielmehr die Prozesskontrolle der Pressenstraße. Diese ist wichtig, da von einer Verschlechterung der Presswerkzeuge oder Verschmutzungen am Stempel Auswirkungen auf die gesamte Serienproduktion ausgehen.

Neben der fehlenden 100%-Kontrolle ist die manuelle Inspektion sehr personal- und arbeitsintensiv. Die Erkennung von Fehlern erfordert ein hohes Maß an Erfahrung und somit einen hohen Schulungsaufwand und hohe laufende Personalkosten. Die Erkennungsrate hat starke, nicht konstante Abhängigkeiten wie z. B. Tagesform des Auditors, Taktzeit, Erfahrung des Personals. Die Erkennung selbst ist subjektiv und stellt eine monotone, ermüdende Arbeit dar. Eine automatische Oberflächeninspektion ist also unumgänglich, nicht nur um Mitarbeiter zu entlasten, sondern um die gewünschten Qualitätsvorgaben überhaupt vollständig erfüllen zu können.



**Abbildung 1.2:** Links: 3D-Messung einer Türgriffmulde, rechts: Darstellung eines Querschnittes mit Defekt innerhalb des Toleranzbereiches

Das zugrundeliegende Ziel dieser Arbeit ist daher die *Entwicklung eines automatisierten Systems zur Oberflächenforminspektion*. Dazu werden folgende wesentliche Beiträge geleistet<sup>1</sup>, die den Stand der Technik erweitern:

- Systematische Beschreibung eines Oberflächeninspektionssystems in seiner Gesamtheit (Kapitel 3),
- Entwicklung eines hierarchischen Ansatzes für den Assoziativspeicher (Kapitel 4),
- Untersuchungen zur Anwendbarkeit von B-Spline-Oberflächen und zugehöriger Methoden in der Oberflächeninspektion (Kapitel 5),
- Entwicklung eines spline-basierten Assoziativspeichers für die Modellierung variabler Oberflächenformen (Kapitel 5),
- Entwicklung eines spline-basierten Registrierungsverfahrens für die hochpräzise Verrechnung mehrerer Teilmessungen (Kapitel 5),
- Entwicklung eines Multi-Schwellen-Verfahrens für die Segmentierung von Oberflächendefekten aus Distanzkarten (Kapitel 6),
- Untersuchungen zur Verwendung eines Klassifikationssystems für die robuste Einschätzung von Oberflächendefekten (Kapitel 6).

---

<sup>1</sup>Die Ergebnisse wurden in wissenschaftlichen Publikationen in Fachkonferenzen und -zeitschriften präsentiert [45, 46, 47, 48, 49, 50, 51, 52, 93, 111].

## 1.3 Aufbau der Arbeit

Die Arbeit ist in 7 Kapitel gegliedert.

Nach der Einleitung wird in *Kapitel 2* der Begriff „Oberflächeninspektion“ definiert und von benachbarten Problemen abgegrenzt. Weiterhin wird ein Stand der Technik von Verfahren zur Lösung des Gesamtproblems gegeben.

In *Kapitel 3* wird ein automatisiertes Oberflächeninspektionssystem in seiner Gesamtheit beschrieben. Dazu wird ein systematischer Überblick über die Architektur eines solchen Systems, die verschiedenen Teilprobleme und ihre Zusammenhänge gegeben. Daraus werden grundsätzliche Lösungsansätze und Möglichkeiten zur Verbesserung abgeleitet.

Die folgenden *Kapitel 4, 5 und 6* sind der Kern der Arbeit und enthalten die wesentlichen wissenschaftlichen Beiträge und die Beschreibung der entwickelten Verfahren. Diese Kapitel enthalten jeweils eine Beschreibung der verwendeten mathematischen und technischen Grundlagen. Anschließend werden jeweils die neu entwickelten Verfahren beschrieben und anhand von Beispielen untersucht. In *Kapitel 4* werden Verbesserungen des *Assoziativspeichers* beschrieben, dabei handelt es sich um ein adaptives Oberflächenmodell. In *Kapitel 5* werden verschiedene Entwicklungen auf Grundlage von B-Spline-Oberflächenbeschreibungen beschrieben, unter anderem die Erweiterung des Assoziativspeichers auf B-Splines sowie ein Registrierungsverfahren basierend auf der B-Spline-Darstellung von Oberflächen. Die Segmentierung, Merkmalsextraktion und Klassifikation von Oberflächendefekten ist Inhalt von *Kapitel 6*.

Abschließend erfolgt in *Kapitel 7* eine Zusammenfassung der Arbeiten sowie ein Ausblick.



---

## Kapitel 2

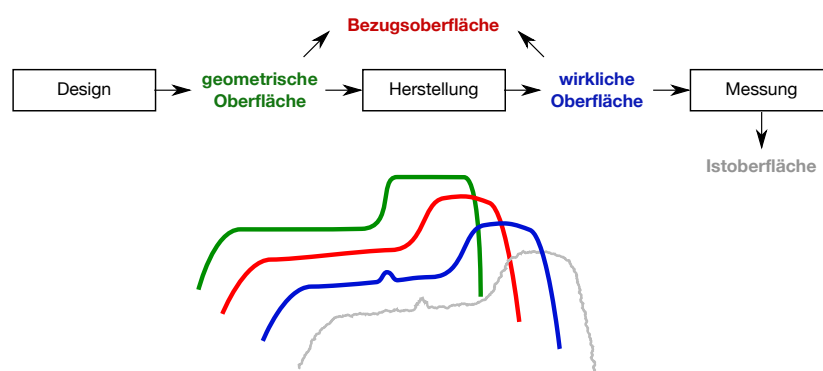
# Oberflächeninspektion

---

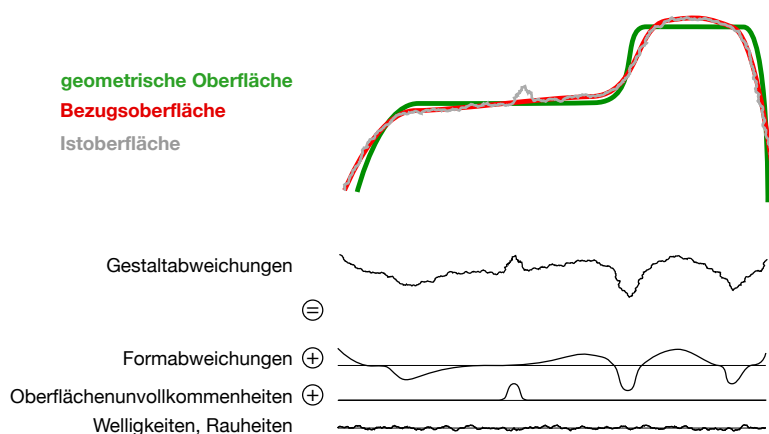
Ziel dieses Kapitels ist die Definition des Begriffes „Oberflächeninspektion“ und die Abgrenzung von benachbarten Problemen in Abschnitt 2.1. Die Bedeutung der Inspektion entlang des Produktentstehungsprozesses für die Karosserie wird in Abschnitt 2.2 dargestellt. Weiterhin wird in Abschnitt 2.3 ein Stand der Technik von Verfahren zur Oberflächeninspektion gegeben, der manuelle, automatische sowie weitere, nicht-optische Verfahren umfasst.

### 2.1 Definition und Abgrenzung

Bei den verschiedenen Prüfaufgaben im Bereich Oberflächenprüfung wird lediglich die sichtbare Hülle eines Objektes geprüft. Beschädigungen oder Defekte im Volumen werden dabei nicht betrachtet. Zur eindeutigen Beschreibung der verschiedenen Aufgaben im Bereich Oberflächenprüfung wurden in DIN 4760 [39] und ISO 8785 [40] verschiedene Oberflächenbegriffe definiert, die in Abbildung 2.1 zusammengefasst sind.



**Abbildung 2.1:** Zusammenfassung der Oberflächenbegriffe nach DIN 4760 und ISO 8785 anhand eines Oberflächenquerschnittes



**Abbildung 2.2:** Arten von Gestaltabweichungen einer Istoberfläche

Die *wirkliche Oberfläche* ist die Oberfläche, die den zu prüfenden Gegenstand von dem ihn umgebenden Medium trennt. Das Resultat ihrer messtechnischen Erfassung ist die *Istoberfläche*, sie ist aufgrund von zufälligen und systematischen Fehlern abhängig von Messverfahren und -bedingungen. Die durch Zeichnungen oder technische Unterlagen vorgegebene ideale Oberfläche wird als *geometrische Oberfläche* bezeichnet. Etwas komplexer gestaltet sich die Definition der *Bezugsfläche* nach ISO 8785 [40]. Sie besitzt die „Form einer geometrischen Oberfläche“, ihr Verlauf ist aber durch die wirkliche Oberfläche definiert und enthält möglicherweise Formabweichungen im Vergleich zur geometrischen Oberfläche. Auf diese Definition wird später genauer eingegangen.

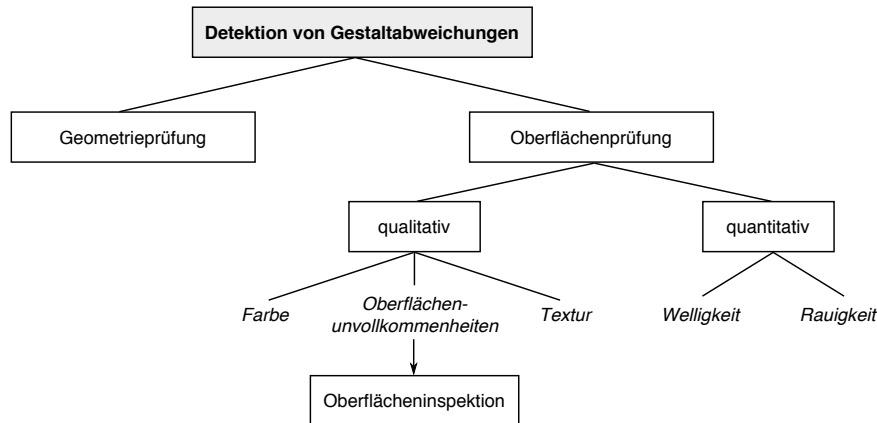
Zunächst lässt sich durch Bestimmung der Abweichungen aus der Istoberfläche und der geometrischen Oberfläche die *Gestaltabweichung* bestimmen, genauer gesagt eine Überlagerung von einzelnen Gestaltabweichungen verschiedener Ordnungen. Diese sind in der Abbildung 2.2 dargestellt, bezogen auf die Beispieloberfläche aus Abbildung 2.1. Die Einteilung in die Ordnungen ergibt sich aus der Ausdehnung der Gestaltabweichung (bzw. bei Abweichungen mit sich wiederholender Form: aus der Periodenlänge der Abweichung) entlang der Oberfläche. Interessant sind dabei vor allem die folgenden drei Arten:

- **Formabweichungen** (Gestaltabweichungen erster Ordnung), wie z. B. Geradheits-, Ebenheits-, Rundheitsabweichung;
- **Oberflächenunvollkommenheiten**, wie z. B. Dellen, Beulen, Kratzer, Riefen;
- **Welligkeiten und Rauheiten** (Gestaltabweichungen höherer Ordnung), z. B. durch Gefügestruktur oder Gitteraufbau des Werkstoffes bedingt.

Daraus leiten sich drei verschiedene Prüfaufgaben mit stark unterschiedlichen Anforderungen ab, die im Allgemeinen verschiedene Lösungsansätze erfordern. Die Bestimmung von Form- und Maßhaltigkeit ist Aufgabe der *Geometrieprüfung*, das Messen von Rauheiten und Gestaltabweichungen höherer Ordnung wird als *quantitative Oberflächenprüfung* bezeichnet, und die Oberflächeninspektion oder präziser *Oberflächenforminspektion* ist die



Untersuchung von Oberflächen auf lokale, qualitative Oberflächenfehler [69]. Die verschiedenen Prüfaufgaben sind in Abbildung 2.3 zusammengefasst. Die verwendeten Verfahren und Messmethoden für diese Aufgaben überschneiden sich teilweise, der Schwerpunkt dieser Arbeit liegt jedoch in der Oberflächenforminspektion.

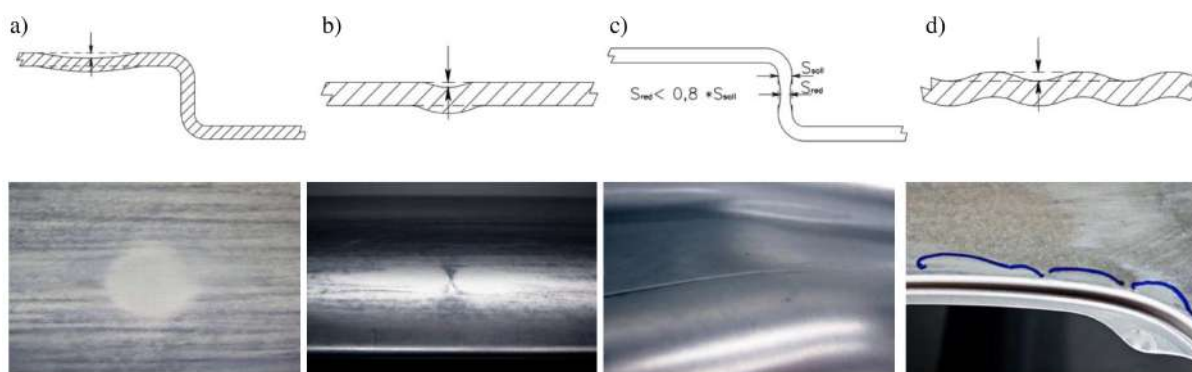


**Abbildung 2.3:** Überblick der verschiedenen Prüfaufgaben und Einordnung der Oberflächeninspektion (Abbildung nach [69])

Hierfür werden zunächst die Abweichungen der Istoberfläche von der Bezugsfläche bestimmt, daraus ergeben sich die *Oberflächenunvollkommenheiten*. Als *Oberflächenfehler oder -defekte* werden anschließend die Unvollkommenheiten bezeichnet, aufgrund derer bestimmte Anforderungen an die Oberfläche nicht erfüllt werden. Die Definition von Oberflächendefekten ergibt sich also aus den Anforderungen, die quantifiziert sein können (z. B. durch eine maximale Defekthöhe oder -ausdehnung), in der Praxis aber oft nur qualitativ gegeben sind (z. B. Minderung des visuellen Oberflächeneindrucks).

Neben der weichen Definition von Oberflächendefekten ist ein weiteres wesentliches Problem die Bestimmung der Bezugsfläche. In DIN EN ISO 8785 ist damit eine „mittlere Oberfläche“ gemeint, die sich „nach der Methode der kleinsten Abweichungsquadrate aus der Istoberfläche ergibt, und durch die höchste Spitze der wirklichen Oberfläche verläuft“. Die Bezugsfläche enthält keine Oberflächenunvollkommenheiten, sie wird an diesen Stellen über die benachbarten Bereiche, ohne den Einfluss von Formabweichungen, eingeschätzt [40]. Diese Definition ist für die Oberflächeninspektion nicht nützlich, da sie Vorwissen über die Lage der Oberflächenunvollkommenheiten erfordert, die gefunden werden sollen. Im Abschnitt 2.3 werden verschiedene Lösungen aus dem Stand der Technik beschrieben.

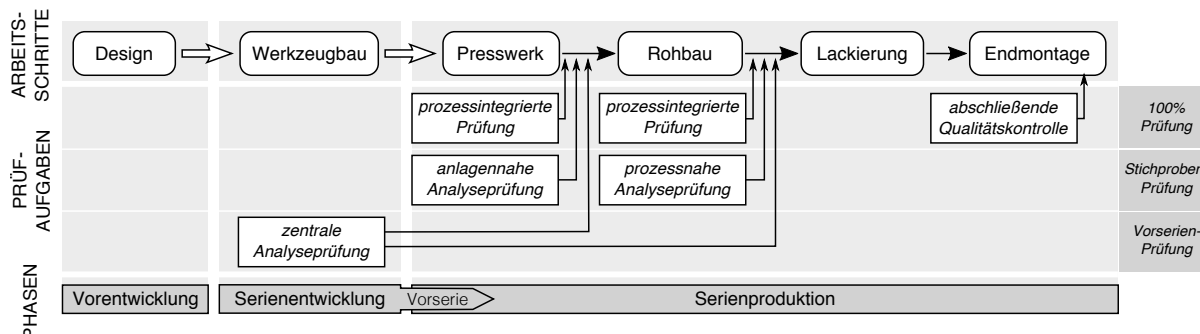
Einige typische Beispiele für Oberflächenunvollkommenheiten nach DIN EN ISO 8785 sind in Abbildung 2.4 dargestellt. Kratzer und andere Fehler, die vor allem die Oberflächentextur verändern, sind nach der weiteren Oberflächenbehandlung (z. B. Lackierung) nicht sichtbar. Interessant sind vor allem 3D-Fehler wie Beulen, Dellen oder Welligkeiten, die nach Lackierung deutlich zum Vorschein treten. Das Verhältnis von Ausdehnung auf der Oberfläche und Höhe zur Bezugsfläche wird mit bis zu etwa 1000:1 angegeben



**Abbildung 2.4:** Beispiele für Oberflächendefekte. Oben: schematische Darstellung im Querschnitt, unten: Fotografie von abgezogenen oder markierten Oberflächen. **a)** Einfallstellen, z. B. Beulen und Dellen, **b)** Druckstellen, Pickel und Pickelnester, **c)** Anzugskanten, Risse und Einschnürungen, **d)** Welligkeiten. Quelle: INB Vision

(Gestaltabweichungen zweiter Ordnung), d.h. eine Delle mit einer Höhe von  $20\ \mu\text{m}$  über der Fläche kann mehrere Zentimeter entlang der Fläche ausgedehnt sein. Dies erschwert die automatisierte Detektion zusätzlich.

## 2.2 Oberflächeninspektion im Karosseriebau



**Abbildung 2.5:** Prüfaufgaben der Oberflächeninspektion in der Produktentstehung

Um die Bedeutung der Oberflächeninspektion im Produktionsprozess besser einzuschätzen, wird hier und im folgenden der Karosseriebau in der Automobilfertigung betrachtet. Für eine detailliertere Darstellung sei auf Özkul [101] oder Braess et al [24] verwiesen, an denen sich die folgenden Ausführungen orientieren. Die verschiedenen Prüfaufgaben und ihre Zusammenhänge mit den Arbeitsschritten und Phasen der Produktentstehung sind in Abbildung 2.5 dargestellt.

Die Karosserie bezeichnet den äußeren Aufbau eines Kraftfahrzeuges und ist somit eine seiner wesentlichen Baugruppen. Die Karosserie setzt sich aus Außenhautteilen zusammen, die über Strukturteile miteinander verbunden werden.

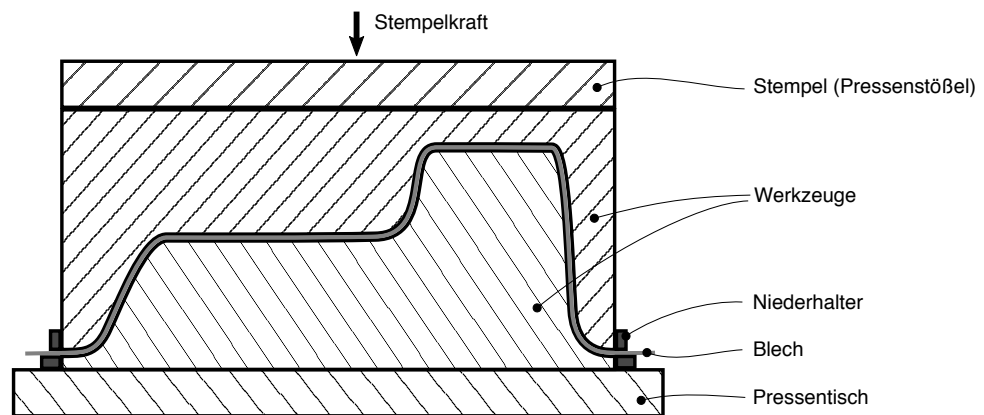


Abbildung 2.6: Mechanisches Tiefziehen eines Blechformteils

**Phasen der Produktentstehung.** Der Produktentstehungsprozess (PEP) für Karosserieaußenteile erstreckt sich von einer Vorentwicklungsphase über die Serienentwicklungsphase bis zur Serienproduktion [132].

Zunächst werden die Form der Außenhaut und Anforderungen an ihre Eigenschaften im Designprozess festgelegt. Dies findet im Wesentlichen in der *Vorentwicklung* statt, kann sich aber bis zum sogenannten „Design-Freeze“, bei dem die Teileigenschaften final definiert werden, in die Serienentwicklung fortsetzen. Die Entwicklung der Außenhautteile ist eng verbunden mit der Entwicklung des zugehörigen Produktionssystems und der Definition der zugehörigen Prozesse. Dies umfasst die Festlegung der notwendigen Bearbeitungsschritte, sowie die Gestaltung und die Fertigung der dafür zu verwendenden Werkzeuge.

Der Werkzeugbau ist wesentlicher Teil der *Serienentwicklung*. Dabei werden bereits erste Karosserieteile gefertigt (siehe auch der folgende Abschnitt *Presswerk*). Am Ende dieser Phase liegen sogenannte Grenzmuster der Karosserieteile im Werk vor, die als Referenz für erlaubte Grenzwerte der Form, der Materialeigenschaften sowie weiterer Merkmale gelten. Diese Grenzmuster definieren somit einen Qualitätsstandard für die Oberfläche und dienen als wesentliche Information für die Detektion von Oberflächendefekten.

In der *Serienproduktion* werden die produzierten Teile anhand der in der Serienentwicklung definierten Norm überwacht. Es findet weiterhin eine Optimierung des Produktes und der Prozesse statt.

**Presswerk.** Der Karosseriebau besteht zunächst aus der Fertigung der Außenhautteile im Presswerk. Ausgehend vom geschnittenen Blechwerkstoff ist der wesentliche Umformungsschritt das Tiefziehen mit starren Werkzeugen. Die Umformung geschieht unter Druck in der Presse, in der die Werkzeughälften an einem festen Pressentisch und einem beweglichen Pressenstößel (Stempel) montiert sind [125], siehe auch Abbildung 2.6. Es erfolgt dann ein schrittweises Beschneiden, Lochen und Nachformen in mehreren Schritten, bis die gewünschte Oberflächenform erreicht ist.

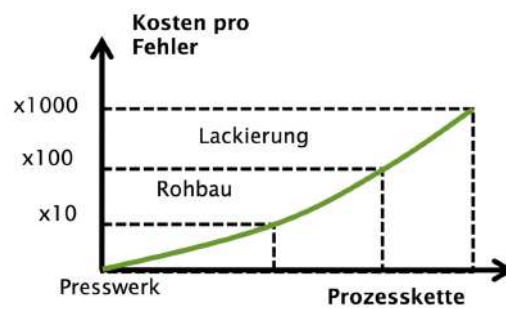
Im Presswerk können Oberflächendefekte durch systematische Verschlechterung der Prozessparameter, wie z. B. durch Abnutzung der Werkzeuge oder sporadisch, z. B. durch Einbringung von Verschmutzungen, entstehen. Das Tiefziehen führt jedoch auch prinzipbedingt zu schlecht vorhersehbaren Verformungen, da dabei eine inhomogene innere Spannungsverteilung im Blechteil hervorgerufen wird. Nach dem Tiefziehen oder Schneiden können so ungewollte Deformationen entstehen. [20, 33] Daher finden drei wesentliche Prüfungen der Oberfläche statt, die verschiedene Ziele verfolgen und mit unterschiedlicher Häufigkeit durchgeführt werden:

- Ziel der *prozessintegrierten Prüfung* ist das Auffinden sporadisch auftretender lokaler Formabweichungen. Dazu findet für alle Teile eine grobe Sichtprüfung statt, bei einer üblichen Taktung von 5 Sekunden pro Bauteil. Eine genauere Sichtprüfung wird stichprobenhaft alle 10 Minuten durchgeführt.
- Ziel der *anlagennahen Analyseprüfung* ist die Prozesskontrolle, also das Auffinden systematischer lokaler Formabweichungen, die auf eine Verschlechterung des Fertigungsprozesses bzw. der Pressenanlage hinweisen. Anhand von Stichproben wird im Stundentakt eine ausführliche manuelle Überprüfung der Oberflächenmerkmale (siehe Abschnitt 2.3.1) und ein Vergleich mit dem Musterteil durch geschulte Auditoren vorgenommen.
- Die *zentrale Analyseprüfung* erfolgt einmalig vor dem Serienlauf oder bei Änderung des Fertigungsprozesses. Das Ziel dieser Prüfung ist somit die Freigabe und Gewährleistung einer stabilen Serienproduktion. Sie erfolgt halbautomatisch mittels optischer und taktiler Sensoren durch einen Messtechniker sowie manuell durch einen Auditor. Dabei werden auch Grenzmuster gesammelt und eingelagert (Erstbemusterung).

**Rohbau.** Bei der anschließenden Montage werden die Außenhautteile gemeinsam mit den Strukturteilen verbunden. Dies geschieht durch Umformprozesse (z. B. Bördeln) oder Fügeprozesse (z. B. Kleben oder Schweißen). Als Ergebnis liegt die unlackierte fertige Karosserie, der sogenannte *body-in-white*, vor.

Im Vergleich zum unmontierten Teil unterliegt die Oberflächenform nun deutlich geringeren Toleranzen; die Form der Außenhaut ist durch die Montage an definierten Punkten nicht mehr von der Auflage abhängig. Typischerweise werden durch die Umform- und Fügeprozesse jedoch Einfallstellen und Welligkeiten eingebracht; weitere Oberflächendefekte können jedoch auch durch Transport und Lagerung entstehen. Die Strategie zur Prüfung der Oberflächen ist ähnlich wie im Presswerk dreistufig:

- Die *prozessintegrierte Prüfung* findet bei sämtlichen Unterzusammenbauten sowie bei sämtlichen fertigen Karosserien statt. Bei einer üblichen Taktzeit von 50 Sekunden kann auch hier nur eine grobe visuelle Kontrolle vorgenommen werden, die einer stark subjektiven Entscheidung des Auditors unterliegt.



**Abbildung 2.7:** Relative Kosten für Nacharbeiten in verschiedenen Stadien der Produktion

- Ziel der *prozessnahen Analyseprüfung* ist es, den Prozess der Montage, aber auch zugehörige Teilprozesse wie den Transport, auf eine systematische Verschlechterung hin zu überwachen. Es erfolgt eine ausführliche manuelle Kontrolle im Wochenrhythmus.
- Die *zentrale Analyseprüfung* wird, ähnlich wie im Presswerk, während der Serienentwicklung durchgeführt und gewährleistet eine stabile Serienproduktion.

Werden Oberflächendefekte gefunden, können sie durch Nacharbeiten entfernt werden. Die Kosten steigen jedoch im Laufe des Prozessfortschrittes exponentiell an, da gegebenenfalls die gesamte Karosserie aus dem Prozess ausgeschleust werden muss (siehe Abbildung 2.7). Daher ist die frühzeitige Detektion von Fehlern, möglichst nach dem Umformprozess im Presswerk, erstrebenswert.

**Endmontage und Lackierung.** Die montierte Karosserie wird anschließend lackiert und bei der Endmontage mit den weiteren Baugruppen wie Antrieb, Fahrwerk, Ausstattung, Elektrik verbunden. Es findet eine abschließende Prüfung statt. Diese erfolgt manuell durch einen Auditor im Lichttunnel oder automatisiert mittels deflektometrischer Systeme (siehe Abschnitt 2.3.2). In diesem Stadium können die Kosten für Nacharbeiten jedoch das 1000-fache des Ausschusses im Presswerk erreichen.

## 2.3 Stand der Technik

Die folgenden Abschnitte sollen einen Überblick der Methoden zur Detektion von Oberflächenunvollkommenheiten geben. Weit verbreitet ist nach wie vor die manuelle Inspektion, die in Abschnitt 2.3.1 dargestellt wird. In Abschnitt 2.3.2 werden automatische Inspektionsverfahren basierend auf optischer Messtechnik beschrieben. Abschließend werden in Abschnitt 2.3.3 weitere für die Oberflächeninspektion angewandte Sensorprinzipien benannt.

Die Verwendung automatischer, optischer Inspektionsverfahren ist heute weit verbreitet. Entsprechend existieren eine ganze Reihe von Verfahren. Echte 3D-Verfahren digitalisieren die Oberflächenform z. B. in Form einer 3D-Punktwolke. Sie werden heute bevorzugt für die Oberflächeninspektion genutzt, da sie eine schnelle, großflächige und hochauflösende Auswertung erlauben und auch bei matten Oberflächen genutzt werden können. Die aufgenommenen Daten können geometrisch gedeutet und einfach visualisiert werden. Die zugehörige Sensorik wird bei der Beschreibung des Gesamtsystems im folgenden Kapitel genauer beschrieben (siehe Abschnitt 3.2).

In dieser Arbeit sollen vor allem *modellbasierte 3D-Verfahren* betrachtet werden. Diese nutzen *a priori* Wissen über die Oberflächenform und erlauben somit die Auswertung beliebig komplex geformter Oberflächen. Insbesondere datengetriebene Modelle sind flexibel für verschiedenste Oberflächen einsetzbar und bieten damit die größten Potentiale für den Einsatz in der industriellen Oberflächeninspektion.

### 2.3.1 Manuelle Verfahren

Wie bereits bei der Beschreibung der Prüfaufgaben im Karosseriebau ersichtlich, ist die manuelle Inspektion zur schnellen Bewertung unlackierter und nicht-montierter Blechteile in der Praxis weit verbreitet. Grob kann die optisch-zerstörungsfreie, die optisch-zerstörende und die taktile manuelle Inspektion unterschieden werden.

**Optisch-zerstörungsfreie manuelle Inspektion.** Diese Methoden zeichnen sich durch eine schnelle Erkennung von Oberflächendefekten aus und können somit bei der prozessintegrierten Prüfung (100%-Kontrolle) angewendet werden.

Bei der einfachen Sichtkontrolle wird die Oberflächentextur unter Berücksichtigung der Beleuchtung durch den Auditor auf Unregelmäßigkeiten begutachtet. Diese Bewertung orientiert sich an den Grenzmustern. Die Entscheidung ist stark subjektiv und vor allem von der Schulung und der Konzentration des Auditors abhängig. Kleine Fehler können nicht erkannt werden. [101]

Auf reflektierenden Oberflächen können Beleuchtungsmuster genutzt werden, was zu einer höheren Sensitivität der Fehlererkennung führt. Typischerweise erfolgt die Prüfung im Lichttunnel, in dem geradlinige, parallele Leuchtmuster um das Prüfteil angebracht sind (siehe Abbildung 2.8). [102]

Eine noch höhere Sensitivität bis auf wenige Mikrometer erreicht man durch die optische Prüfung mit einem Haarlineal [80]. Dieses wird an die Oberfläche angelegt. Mit Gegenlicht werden dann Lichtspalte erkannt, die auf Oberflächendefekte hinweisen. Diese Prüfmethode ist deutlich langsamer und wird daher nur bei Stichprobenprüfungen (anlagen- oder prozessnah) eingesetzt.



**Abbildung 2.8:** Optisch-zerstörungsfreie manuelle Inspektion in einem Lichttunnel. Quelle: Volkswagen AG



**Abbildung 2.9:** Optisch-zerstörende manuelle Inspektion mit einem Abziehstein. Quelle: Audi AG

**Optisch-zerstörende manuelle Inspektion.** Bei nicht-reflektierenden Oberflächen können durch eine rein optische Prüfung kleinere Fehler nicht detektiert werden, da sie nur bei sehr spezifischer Beleuchtung zu einer sichtbaren Änderung der Oberflächentextur führen. Zur Sichtbarmachung kleiner Defekte werden daher zerstörende Verfahren eingesetzt, bei der die Oberfläche mittels Abziehstein, Schleifpad oder Schmirgelpapier leicht abgeschliffen wird [17]. Das resultierende Schliffmuster kann dann optisch begutachtet werden; in konkaven Bereichen sind keine oder schwächere Schleifriefen zu erkennen, in konvexen Bereichen sind stärkere Schleifriefen zu sehen. Die Inspektion mit dem Abziehstein ist in Abbildung 2.9 dargestellt; typische Schliffmuster an Oberflächendefekten sind in Abbildung 2.4 zu sehen.

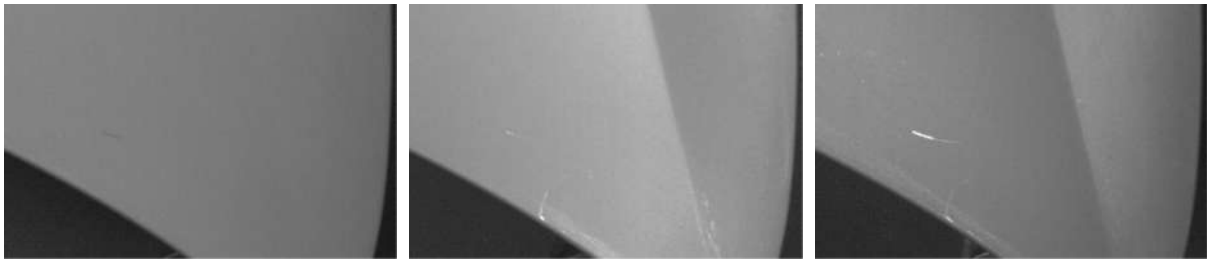
Die optisch-zerstörende manuelle Inspektion wird vornehmlich anlagen- oder prozessnah

an Stichprobenteilen durchgeführt.

**Taktile manuelle Inspektion.** Alternativ zu zerstörenden Verfahren können nicht-reflektierende Oberflächen auch taktil inspiziert werden. Hierzu führt ein Auditor seine Hand mit einem Fühlhandschuh über die Oberfläche. Unebenheiten von unter 30 µm können dadurch von einem erfahrenen Auditor ertastet werden. [101]

Da diese Methode sehr zeitaufwändig sein kann, wird sie vor allem lokal an Stellen angewendet, die entweder typisch für Prozessfehler in der Serie sind oder an denen zuvor bereits eine optische Kontrolle auffällig war. Die Inspektion mit Fühlhandschuh findet vor allem bei der zentralen Analyseprüfung und vereinzelt prozessintegriert im Rohbau statt.





**Abbildung 2.10:** Erkennung von Kratzern aus der beobachteten Oberflächentextur eines Blechteiles bei verschiedenen Beleuchtungsrichtungen

### 2.3.2 Automatische optische Verfahren

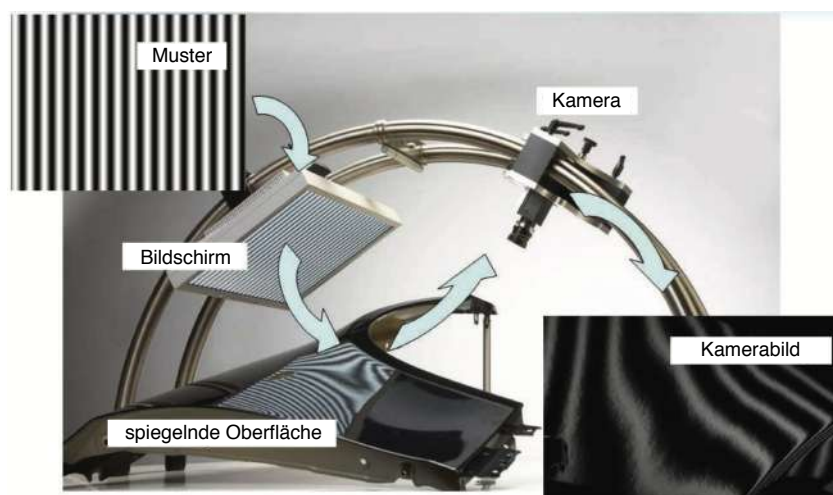
Für die automatisierte Inspektion bieten sich Verfahren auf Grundlage optischer Messtechnik mit anschließender Bildverarbeitung an. Diese arbeiten berührungslos, sind flexibel einsetzbar und gewährleisten eine gute Erweiterbarkeit und Anpassung, insbesondere hinsichtlich der Datenverarbeitung. Es existieren eine Reihe verschiedener Ansätze zur Datenauswertung optischer Sensoren. Diese lassen sich grob aufteilen in Ansätze der *Texturauswertung*, *Deflektometrie* sowie *modellfreie* und *modellbasierte 3D-Datenauswertung*.

Einen Überblick über optische Inspektionsmethoden liefert Newman [96], zu diesem frühen Zeitpunkt noch mit einem starken Fokus auf 2D-Bildverarbeitung. In dem Überblick von Malamas [91] werden verschiedene Anwendungsbeispiele für spezialisierte industrielle Inspektionssysteme zusammengefasst. Neogi [95] liefert einen Überblick über Inspektionssysteme für Stahloberflächen. Durch die schnelle Entwicklung in den letzten 20-30 Jahren sind aktuellere Überblicke wesentlich spezialisierter; diese werden in den folgenden Abschnitten berücksichtigt.

**Texturauswertung.** Die ersten Systeme zur Oberflächeninspektion basierten vor allem auf klassischen industriellen 2D-Bildverarbeitungssystemen. Dabei wird das Oberflächenabbild unter geeigneter Beleuchtung aufgenommen und die beobachtete Textur ausgewertet.

Xie [141] liefert einen Überblick zur texturbasierten Auswertung von Oberflächenfehlern. Darin zeigt sich, dass die Texturauswertung vor allem bei Defekten mit geringer Tiefenausprägung (z. B. Kratzer) sinnvoll ist. Es werden Texturmodelle wie die Co-Occurrence-Matrix verwendet (siehe [4]) oder geeignete Merkmale definiert und für die Auswertung genutzt (siehe [99]).

Für Defekte mit geometrischer Ausprägung (z. B. Dellen oder Beulen) kann durch gezielte Beleuchtung aus verschiedenen Richtungen indirekt auf Tiefeninformation geschlossen werden (angewandt z. B. in [73] oder [57]). Dies ist jedoch nur bei konvex geformten Oberflächen mit geringer Geometrievariation sinnvoll möglich und erfordert eine gezielte



**Abbildung 2.11:** Prinzipieller Aufbau des reflectCONTROL Deflektometriesensors der Firma micro epsilon (nach [88])

Einrichtung der Beleuchtung (z. B. schräg zur Oberfläche, siehe Abbildung 2.10). Nickolay [97] hat bereits 1992 ein System entwickelt, das Texturinformationen aus zwei Kamerawinkeln auswertet. Dies kann als Vorläufer von echten 3D-Auswertungen betrachtet werden.

**Deflektometrie.** Deflektometrische Verfahren werden auf spiegelnden oder teilweise spiegelnden Oberflächen angewendet. Dazu wird ein Bildschirm mit Streifenmuster über der Oberfläche angeordnet. Eine digitale Kamera nimmt dann das an einer Oberfläche reflektierte Muster auf, und fokussiert dabei auf das Muster (siehe Abbildung 2.11). Es wird also nicht die Oberfläche selbst, sondern ihre reflektierende Eigenschaft als Teil des optischen Systems beobachtet [72]. Die Oberflächengeometrie wird so indirekt über ihre Krümmung gemessen [127], deflektometrische Verfahren sind somit sehr tiefensensitiv und können bei spiegelnden Oberflächen bis zu 1 nm auflösen [22].

Im Karosseriebau sind diese Verfahren erst nach Lackierung sinnvoll nutzbar. Es existieren eine Reihe kommerzieller Prüfsysteme (z. B. [88]). Dabei können auch vorhandene, für die manuelle Inspektion genutzte Lichttunnel genutzt werden [8, 128], die Tiefenauflösung liegt dabei jedoch knapp unter 1 mm. Komplex geformte Bauteile erschweren die Auswertung deutlich und erfordern möglicherweise, wie 3D-Verfahren auch, modellbasierte Ansätze [79]. Nicht-spiegelnde Blechteile im Presswerk können möglicherweise durch Nutzung von Infrarotmustern und Wärmekameras dennoch deflektometrisch ausgewertet werden [121]. Die Sensitivität ist dabei trotz des höheren technischen Aufwandes für den deflektometrischen Aufbau vergleichbar zu 3D-Verfahren mit Musterprojektion. Einen Vergleich zwischen Deflektometrie und photogrammetrischen Verfahren macht Hung [142].

**Modellfreie 3D-Datenauswertung.** Photogrammetrische Messsysteme erlauben die berührungslöse, schnelle, hochpräzise, flächige Erfassung von diffus reflektierenden Oberflächen (siehe Abschnitt 3.2). Die resultierenden Messdaten (z. B. Kamerabilder oder 3D-Oberflächendaten) können direkt und intuitiv gedeutet werden. Es existieren eine Reihe von Verfahren, die 3D-Punktwolken geeignet weiterverarbeiten. Grundannahme modellfreier Methoden ist, dass Oberflächendefekte im Vergleich zur Oberflächenform eine deutlich kleinere örtliche Ausdehnung besitzen. Durch geeignete lokale Filterung der Messdaten kann daher eine defektfreie Bezugsoberfläche ermittelt werden, anhand derer Oberflächenfehler detektiert werden können.

Geeignete Filtermethoden wurden aus verschiedensten Disziplinen entlehnt und ihre Anwendung für die Oberflächeninspektion untersucht:

- **Signalorientierte Bildverarbeitung:** In [36] wurden die Bezugsoberfläche, Defekte sowie Rauschen und Kratzer durch Transformation und Filterung der Punktwolke im Ortsfrequenzraum separiert. Es wurde auch die Transformation mit Wavelet-Methoden untersucht [112].
- **Analytische Geometrie:** Polynomflächen (siehe Abschnitt 3.3.3) können als Flächenapproximatoren und somit als Bezugsoberfläche für die Inspektion verwendet werden [43], [42].
- **Computer-aided Design:** Unter Flächenrückführung (*Reverse Engineering*) versteht man die Bestimmung eines NURBS oder Spline-Modells aus einer gemessenen Oberfläche. Die Methoden können auch für die Ermittlung einer Referenzoberfläche für die Inspektion genutzt werden [26].
- **Computergrafik:** Methoden des *Mesh Processings* bzw. *Geometry Processing* können zur Filterung verwendet werden. In [105] wurde dazu die Oberfläche als Dreiecksnetz (siehe Abschnitt 3.3.1) betrachtet.
- **Statistik:** In [21] wird in lokalen Oberflächenbereichen die mittlere Oberflächennormale sowie ihre Standardabweichung ermittelt. Eine hohe Standardabweichung weist dabei auf Merkmale, also auch auf Oberflächendefekte, hin.
- **Mechanik:** Der Abziehstein wurde in Abschnitt 2.3.1 als mechanisches, zerstörendes Werkzeug für die manuelle Oberflächeninspektion beschrieben. Chung [29, 30] beschreibt, wie die Wirkung eines Abziehsteines auf die Oberfläche an der gemessenen Punktwolke simuliert und ausgewertet werden kann.

Modellfreie Verfahren sind für spezifische Anwendungen sinnvoll, insbesondere bei sehr flachen, merkmalsfreien Bauteilen wie z. B. gewalzten Blechen [94]. Alle Verfahren haben gemeinsam, dass sie bei komplexer Oberflächenform sehr schwierig parametrisierbar oder überhaupt nicht sinnvoll einsetzbar sind.

**Modellbasierte 3D-Datenauswertung.** Modellbasierte Verfahren bringen Wissen über die Oberfläche ein, um so eine bessere Darstellung der Bezugsfläche zu gewährleisten. Dies ist insbesondere bei komplex geformten Bauteilen mit starken Krümmungsbereichen notwendig.

Li [81] gibt einen umfassenden Überblick zu verschiedenen Aspekten der Oberflächeninspektion, ausgehend von Messmethoden über Oberflächendarstellungen bis hin zu Modellvergleichen. Avenhaim [2] fasst ebenfalls eine Reihe von Verfahren zusammen, mit dem Fokus auf der Inspektion verformbarer Oberflächen.

Es zeigt sich, dass ein Großteil existierender modellbasierter Verfahren auf dem CAD-Modell des Bauteils beruhen (z. B. [18, 82]). Diese geometrische Oberfläche weicht, insbesondere bei verformbaren Oberflächen bzw. Teilen mit hohen Toleranzen, deutlich von einer geeigneten Bezugsfläche ab. Toleranzbereiche sind oft in den CAD-Daten enthalten und können bei der Verformung berücksichtigt werden [61]. Denkena [37] schlägt eine Filterung im Ortsfrequenzbereich nach dem Modellvergleich vor.

Um die Verformung flexibler Oberflächen im Modell zu berücksichtigen, sind physikalische, FEM-basierte Verformungsmodelle üblich [3, 115, 134]. Diese ist sehr rechenintensiv und daher für die prozessnahe Inspektion nicht praktikabel. Es wurden daher verschiedene Approximationen der FEM-Verformung untersucht [5, 70]. Die erreichten Genauigkeiten sind mit knapp unter 0.25 mm jedoch nicht hinreichend für die Oberflächeninspektion, außerdem erfordern diese Algorithmen eine vergleichsweise komplexe Parametrisierung.

Datengetriebene Modelle nutzen reale Messungen als Trainingsdaten, um die geometrische Form der Oberfläche, aber insbesondere auch mögliche Verformungen und Toleranzbereiche, zu lernen. Der *Assoziativspeicher* nach Lilienblum [85, 86] ist ein autoassoziatives künstliches neuronales Netz und wurde bereits an konkreten Beispielen erprobt. Er wird bzw. wurde in dieser Arbeit und den zugehörigen wissenschaftlichen Publikationen [45, 46, 47, 51, 52] weiterentwickelt.

### 2.3.3 Nicht-optische Methoden

Im folgenden werden Sensormodalitäten beschrieben, die neben der optischen Messung für die Oberflächeninspektion denkbar sind. Nicht betrachtet werden zerstörende Prüfverfahren, die für eine 100%-Kontrolle nicht in Frage kommen.

- **Taktile Messverfahren** arbeiten mechanisch berührend und bieten sehr hohe Messgenauigkeiten bis zu  $0.01 \mu\text{m}$  [133]. Sie werden daher vor allem für die Offline-Referenzmessung eingesetzt, so z. B. Koordinatenmessgeräte für die absolute geometrische Vermessung oder Tastschnittgeräte für die Rauheitsmessung. Auf Grund der punkt- oder linienweisen Abtastung und entsprechend hoher Messdauer sind taktile Verfahren nicht online-fähig. Durch die Berührung erfolgt außerdem eine Krafteinwirkung auf die Oberfläche.

- **Elektromagnetische Verfahren** lassen sich bei elektrisch leitenden Werkstoffen anwenden. Beim Wirbelstromverfahren [126] wird durch ein magnetisches Wechselfeld eine Wechselfeldspannung im Bauteil induziert. Die resultierenden Wirbelströme erzeugen wiederum ein magnetisches Wechselfeld, das detektiert wird. Es lassen sich nur sehr einfache Inhomogenitäten der Oberfläche erkennen (z. B. Risse ab einer Tiefe von  $200\ \mu\text{m}$ ), dafür lassen sich jedoch auch Materialeigenschaften sowie Defekte im Inneren des Bauteils messen.
- **Thermografische Verfahren** basieren auf der Erkennung von Änderungen in der Wärmeausbreitung durch Oberflächendefekte. Nach gezielter Erwärmung des Bauteils treten an Oberflächendefekten Temperaturunregelmäßigkeiten auf. Die Wärmeausbreitung wird mittels Wärmebildkamera gemessen und ausgewertet [12]. Auf diese Weise können auch verdeckte Risse und Volumendefekte detektiert werden. Es können Defekte mit einer Ausbreitung von bis zu  $200\ \mu\text{m}$  lokalisiert werden.



---

## Kapitel 3

# Ein System zur Automatisierten Oberflächeninspektion

---

In diesem Kapitel soll ein System zur automatisierten Oberflächeninspektion im Überblick dargestellt werden. Es handelt sich dabei um ein *modellbasiertes System*, bei dem *a priori*-Wissen über die Oberflächenform in einem geeigneten Oberflächenmodell vorliegt. Dies erlaubt die Defekterkennung auch bei komplexer Oberflächenform.

Es werden zunächst die wesentlichen Teilaspekte des Systems in Abschnitt 3.1 im Überblick vorgestellt. Dies umfasst Grundlagen über Verfahren zur Messwertaufnahme (Abschnitt 3.2), Verfahren zur Oberflächenbeschreibung (Abschnitt 3.3), Verfahren zur Oberflächenmodellierung, insbesondere den Assoziativspeicher (Abschnitt 3.4), sowie Verfahren zur Auswertung von Tiefenkarten (Abschnitt 3.5). Neben der Darstellung der Funktionsweise dieser Verfahren sollen Herausforderungen abgeleitet werden, aus denen sich Potentiale zur Erforschung und Weiterentwicklung ergeben. Diese sind Grundlage für die weiteren Untersuchungen in der vorliegenden Arbeit.

### 3.1 Architektur des Systems

**Die 2D-Bildverarbeitungskette.** Für den Aufbau von Mustererkennungssystemen in der industriellen Anwendung hat sich die klassische Bildverarbeitungskette etabliert [68]. Da sie auch dem Aufbau des Defekterkennungssystems zugrunde liegt, soll sie hier anhand von Abbildung 3.1 kurz beschrieben werden. Ausgehend von der *Bildaufnahme* wird eine Szene optisch auf ein zweidimensionales Bild abgebildet und anschließend digitalisiert. Ergebnis ist eine Matrixrepräsentation, bei der jedes Bildelement (Pixel) durch eine eindeutige, diskrete 2D-Bildkoordinate beschrieben wird und ihm ein oder mehrere diskrete Zahlenwerte zugeordnet sind (z. B. ein skalarer Ganzzahlwert von 0...255 für 8-Bit Grauwerte oder ein dreidimensionaler Vektor für die Farbkanäle Rot, Grün, Blau). Durch eine geeignete *Vorverarbeitung* kann das Bild hinsichtlich der zu beobachtenden Merkmale optimiert werden. Übliche Operationen sind z. B. die Anwendung linearer oder nichtlinearer Hoch- und Tiefpassfilter oder Punktoperatoren für die Korrektur der Sensorkennlinie.

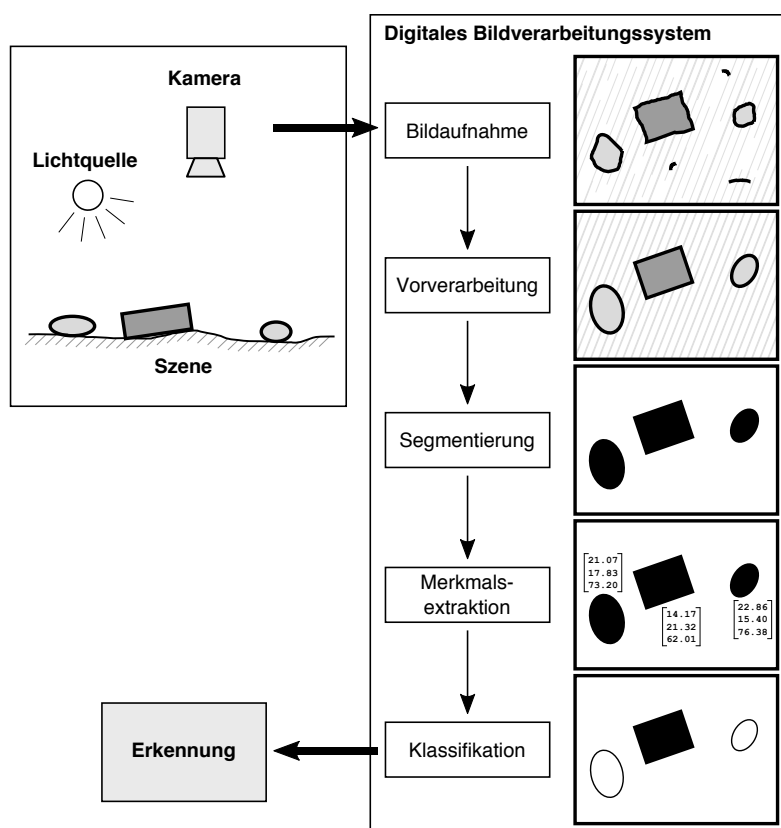
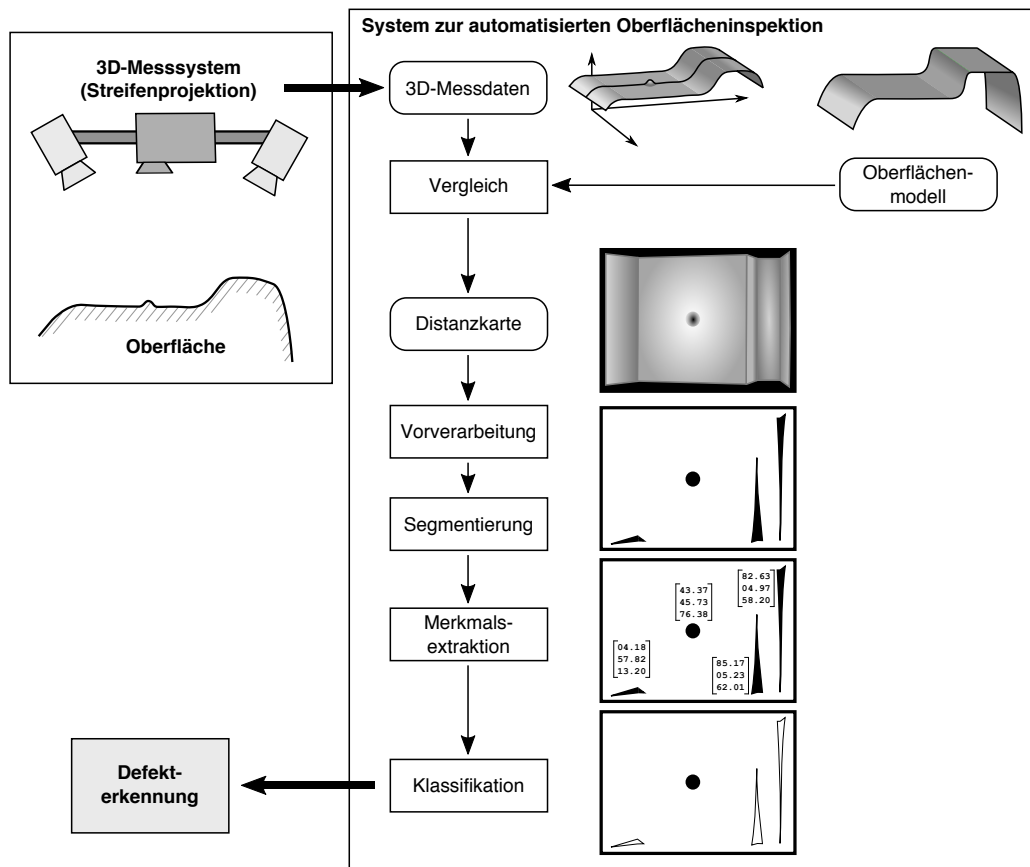


Abbildung 3.1: Schematische Darstellung einer klassischen 2D-Bildverarbeitungskette

Es folgt die eigentliche Bildanalyse. Ziel der *Segmentierung* ist zunächst die Lokalisierung relevanter Bildbereiche, die zum Beispiel ein gewisses Muster oder ein Objekt enthalten. Für diese Bereiche wird eine *Merkmalsextraktion* vorgenommen – Ziel ist es, mit einer Menge quantifizierbarer Merkmale eine charakteristische, eindeutige Beschreibung der Bereiche zu erreichen. Die resultierenden Zahlenwerte werden für jeden segmentierten Bereich in Merkmalsvektoren zusammengefasst, mit denen eine *Klassifikation* möglich ist. Die Bereiche können so entsprechend ihrer Eigenschaften in zwei oder mehrere Klassen eingeordnet werden, womit die Erkennung eines Objektes oder eines Musters durch das System geleistet wird.

**Verarbeitungskette für 3D-Defekterkennung.** Überträgt man den Systemaufbau der klassischen Bildverarbeitungskette auf ein System für die 3D-Datenverarbeitung, so tritt an die Stelle der Bildaufnahme eine geeignete 3D-Messwertaufnahme (Abschnitt 3.2). Dabei wird eine Menge von Oberflächenpunkten einer dreidimensionalen Szene vermessen und im Allgemeinen als 3D-Punktwolke repräsentiert. Diese liegt oft als geordnete Punktwolke vor (z. B. als Tiefenkarte); eine Überführung in andere Oberflächendarstellungen ist möglich (Abschnitt 3.3). Anschließend können die 3D-Daten durch geeignete Vorverarbeitung optimiert werden, z. B. durch Eliminierung von Ausreißern oder Filtern von Messrauschen.





**Abbildung 3.2:** Schematische Darstellung der 3D-Datenverarbeitungskette für ein System zur Defekt-erkennung

Eine wesentliche Abweichung von der klassischen Bildverarbeitungskette ergibt sich durch das Einbringen von Vorwissen über die Oberflächenform (Abschnitt 3.4). Dazu werden die 3D-Daten mit einem Oberflächenmodell verglichen, das als Bezugsfläche im Sinne von ISO 8785 dient (vgl. Abschnitt 2.1). Die Herausforderung dabei ist die Überführung der Messung in das Modellkoordinatensystem, oder umgekehrt die Überführung des Oberflächenmodells in das Messkoordinatensystem, sowie die Berücksichtigung von Fertigungstoleranzen, Messfehlern und Verformung von nicht-starren Oberflächen.

Das Ergebnis des Modellvergleichs ist eine Distanzkarte, mit der die Abstände zwischen Messung und Modell entlang der Oberfläche beschrieben wird. Diese ist ähnlich wie eine Tiefenkarte als 2D-Matrix mit reellen Werten gegeben oder kann in diese Form überführt werden. Obwohl die Distanzkarte 3D-Daten repräsentiert, ist so eine weitere Verarbeitung mit Methoden der 2D-Bildverarbeitung möglich (Abschnitt 3.5). Das Ziel ist das Finden von Bereichen mit großer Modellabweichung (d.h. zusammenhängende Bereiche mit hohen Distanzwerten) sowie die Entscheidung, ob der Bereich tatsächlich aufgrund eines Defektes zustande kommt.

Dazu wird zunächst eine Vorverarbeitung der Distanzkarte durchgeführt, die – ähnlich wie in der 2D-Bildverarbeitung – Operationen wie lineare Hoch- oder Tiefpassfilter beinhaltet. Dies vereinfacht die Weiterverarbeitung und Defektdetektion. Aus der vorverarbeiteten Distanzkarte (die auch als Defektkarte oder engl. *defect map* bezeichnet wird) können nun Bereiche mit hohen Abweichungen segmentiert werden. Für diese Bereiche werden geeignete Merkmale extrahiert.

Mit dem resultierenden Merkmalsvektor kann über ein geeignetes Klassifikationsverfahren die Entscheidung „Defekt“ oder „kein Defekt“ getroffen werden, um echte Defekte von Modell- oder Messfehlern zu unterscheiden. Es ermöglicht auch die Klassifizierung verschiedener Arten und Ausprägungen von Defekten, die jedoch nicht Teil dieser Arbeit ist (siehe dazu z. B. [41, 42, 44]).

Die gesamte Verarbeitungskette zur modellbasierten Defektdetektion aus 3D-Messwerten ist in Abbildung 3.2 dargestellt. Die einzelnen Komponenten und zugrundeliegende Verfahren werden in den folgenden Abschnitten diskutiert.

## 3.2 Messwertaufnahme

Für die Oberflächeninspektion ist eine möglichst dichte flächige Messung bei hoher Tiefenauflösung notwendig, die außerdem keinen Einfluss auf den Produktionsprozess nimmt und robust in der rauen Produktionsumgebung arbeitet. Zusammengefasst lassen sich die folgenden Anforderungen an die Messtechnik ableiten:

### 1) Großes Messfeld

- Mit einem Messvorgang muss eine Oberfläche von mindestens  $300\text{ mm} \times 300\text{ mm}$  erfasst werden.

### 2) Hohe laterale Auflösung

- Der Abstand des Messpunktrasters muss  $1\text{ mm} \times 1\text{ mm}$  oder kleiner sein.

### 3) Hohe Tiefenauflösung

- Die Auflösung entlang der Messrichtung muss  $10\text{ }\mu\text{m}$  oder geringer sein.

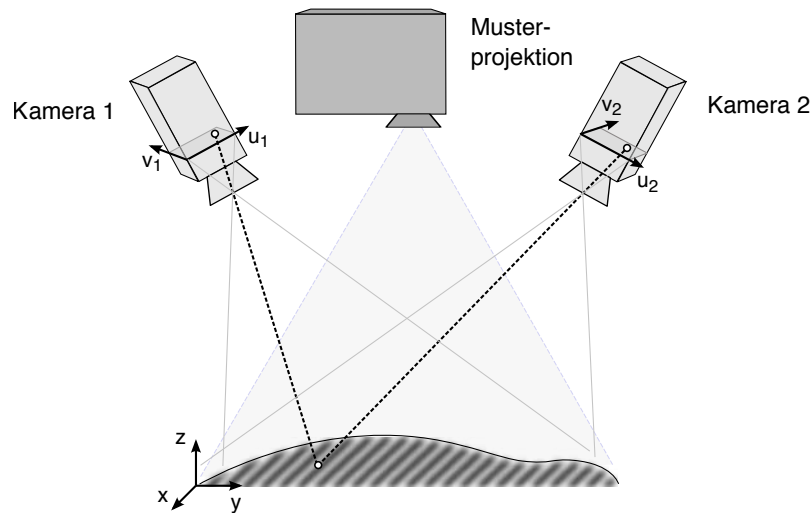
### 4) Berührungslose Messung

### 5) Geringe Gesamtmessdauer

- Ein Messvorgang muss in unter 1 s abgeschlossen sein.

### 6) Robustheit der Messung

- Die Messung muss robust gegenüber Umgebungsbedingungen sein, wie z. B. Schwingungen während der Messung, Umgebungslicht, Verdreckung.



**Abbildung 3.3:** Prinzip der Triangulation mit zwei Messkameras und Musterprojektion

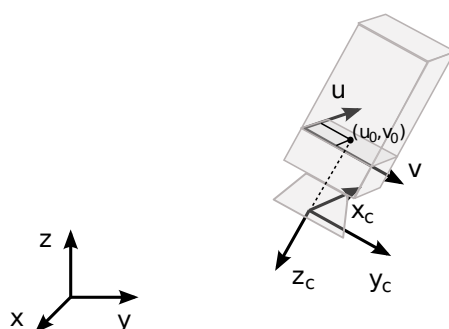
- Die Messung muss robust gegenüber ungünstigen Oberflächeneigenschaften sein, wie z. B. Oberflächentextur, glänzende, dunkle oder transparente Oberflächen.

Dabei kommen optische 3D-Messverfahren in Frage (empfohlen sei der Überblick in [32] oder [19] sowie [120, 123, 144]). Insbesondere optische Messsysteme mit Musterprojektion erfüllen diese Anforderungen. Die Funktionsweise wird in Lehrbüchern (z. B. [89], [140]) ausführlich beschrieben und wird im vorliegenden Kapitel kompakt zusammengefasst. Es soll dabei insbesondere auf Eigenschaften und Verfahren eingegangen werden, die im Zusammenhang mit den speziellen Anforderungen der Oberflächeninspektion stehen. Dazu gehören Erwägungen zur lateralen Auflösung und der Tiefenauflösung, deren Zusammenhang mit Bildkorrelationsverfahren sowie die Betrachtung verschiedener Methoden zur Suche korrelierender Bildpunkten.

### 3.2.1 Photogrammetrische Messverfahren

Prinzip photogrammetrischer Messverfahren ist die Ermittlung von 3D-Informationen aus zwei oder mehr 2D-Abbildungen einer zu vermessenden Szene. Dies waren ursprünglich Fotografien, aus denen zum Zwecke der Fernerkundung die räumliche Lage von sichtbaren und identifizierbaren Objekten bestimmt wurde. In der heute üblichen Anwendung werden digitale Kameras verwendet. Die zugeschlossenen Objektpunkte werden in diesem Fall mittels automatischer Bildverarbeitung in den digitalen Bildern gefunden.

Abbildung 3.3 zeigt das Grundprinzip der *Triangulation* mit einer zusätzlichen Streifenlichtprojektion zur Lösung des Korrespondenzproblems. Im Idealfall bilden ein Objektpunkt und die korrespondierenden Bildpunkte in zwei Kameras ein Dreieck; die Koordinaten des Objektpunktes lassen sich aus den bekannten Kamerapositionen und den zu den



**Abbildung 3.4:** Welt-  $(x, y, z)$ , Kamera-  $(x_c, y_c, z_c)$  und Bildkoordinatensystem  $(u, v)$  zur Beschreibung der Kalibrierung

Bildpunkten gehörigen Sehstrahlen ermitteln. In der Praxis stehen Strahlen meist windschief zueinander, sie ergeben also keinen eindeutigen Schnittpunkt. Der Objektpunkt ergibt sich dann aus dem Punkt mit dem geringsten Strahlabstand.

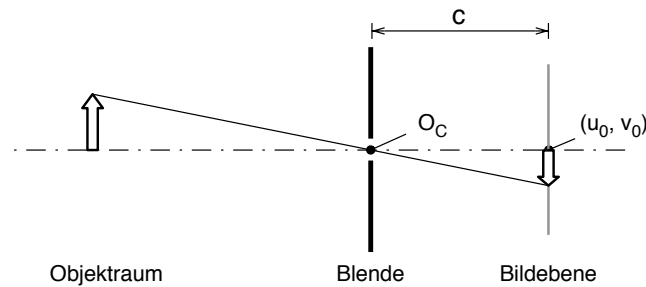
Die Photogrammetrie stellt dafür die folgenden Konzepte zur Verfügung: Das *Kameramodell* beschreibt die Abbildung eines Lichtstrahles aus dem Objektraum in den Bildraum der Kamera mathematisch. Durch Rückprojektion lassen sich daraus für jeden Bildpunkt die zugehörigen Sehstrahlen rekonstruieren. Die Parameter des Kameramodells müssen durch *Kamerakalibrierung* ermittelt werden, dies umfasst die Abbildungseigenschaften der verwendeten Kameras (innere Orientierung) sowie die zu den Bildern zugehörige Ausrichtung und Position im Raum (äußere Orientierung).

Die Bestimmung korrespondierender Objektpunkte in den Bildern kann z. B. manuell erfolgen, geschieht heute aber vorwiegend durch *digitale Bildkorrelation* (siehe Abschnitt 3.2.2). Zur Vereinfachung dieser Bildverarbeitungsaufgabe bzw. zur Erhöhung erreichbarer Genauigkeiten können Lichtmuster auf die Oberfläche projiziert werden. Für die effiziente Suche zusammengehöriger 2D-Bildpunkte und 3D-Objektpunkte existieren verschiedene Algorithmen (3.2.3).

**Koordinatensysteme.** Die mathematische Formulierung der photogrammetrischen Messung erfolgt in drei Koordinatensystemen, die in Abbildung 3.4 zusammenfassend dargestellt sind. Das *Weltkoordinatensystem*  $(x, y, z)$  ist ein 3D-Koordinatensystem, das eine eindeutige Positionierung im Messraum erlaubt. Der Ursprung und die Orientierung können beliebig aber gleichbleibend im 3D-Raum festgelegt werden. Üblicherweise erfolgt die Festlegung anhand des Kalibrierkörpers; es ist zweckmäßig den Ursprung innerhalb des Messfeldes zu wählen und die positive  $z$ -Richtung zum Sensor hin zeigend zu definieren.

Das *Kamerakoordinatensystem*  $(x_c, y_c, z_c)$  definiert die Kameraposition und Orientierung (Blickrichtung) relativ zum 3D-Raum. Dabei liegt der Ursprung  $O_c$  im Bildhauptpunkt, mit der  $Z$ -Achse entlang der Kamerablickrichtung orientiert.

Die *Bildkoordinaten*  $(u, v)$  werden in Pixeln angegeben, dabei befindet sich der Ursprung üblicherweise links oben im Bild. In der Praxis treten dabei verschiedene Konventionen



**Abbildung 3.5:** Zentralprojektive Abbildung mit Projektionszentrum  $O_C$ , Kammerkonstante  $c$  und Bildhauptpunkt  $u_H, v_H$

auf: Die Zählung der Pixel kann bei 0 oder 1 beginnen und der Koordinatenursprung kann in der Mitte oder am Rand des ersten Pixels liegen.

**Kameramodell.** Üblicherweise wird ein Lochkameramodell angenommen, das auf der Zentralprojektion durch einen Bildhauptpunkt basiert (siehe Abbildung 3.5). Dies ist eine sinnvolle Näherung, wenn der Bildhauptpunkt hinreichend nah beim Brennpunkt des Objektivs liegt (Fokussierung auf unendlich) und die Blende hinreichend klein ist (große Blendenzahl).

Es existieren alternative Kameramodelle, wie zum Beispiel implizite Kamerakalibrierungen [135, 136], bei denen statt des physikalischen Modells ein rein empirisches Modell verwendet wird. In einigen Fällen ist dies sogar notwendig, zum Beispiel bei der Verwendung telezentrischer Objektive. Hier findet eine parallelperspektivische Abbildung statt, die durch das Lochkameramodell nur unzureichend dargestellt werden kann.

Das gewählte Kameramodell basiert auf [129]. Mittels der äußeren Orientierung, der Rotationsmatrix  $\mathbf{R}$  und dem Translationsvektor  $\mathbf{t}$ , erfolgt die Transformation vom Welt- in das Kamerakoordinatensystem:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \mathbf{R} \cdot \left[ \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \mathbf{t} \right] \quad (3.1)$$

Die Abbildung von 3D-Kamerakoordinaten in 2D-Bildkoordinaten erfolgt über den Strahlensatz, indem die Objektweite zur Bildweite ins Verhältnis gesetzt wird. Die Bildweite entspricht der Kammerkonstante  $c$  im Lochkameramodell, wobei hier zusätzlich auch ein Skalierungsfaktor für den Übergang in Pixel beinhaltet sein kann.

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix} = \frac{c}{z_c} \cdot \begin{pmatrix} x_c \\ y_c \end{pmatrix} \quad (3.2)$$

Entsprechend der Definition des Bildkoordinatensystems wird der Ursprung aus der Mitte des Bildes an den Rand verschoben, hierzu dient der Bildhauptpunkt  $(u_H, v_H)$ . Weiterhin kann mit dem Skalierungsparameter  $s_y$  eine ungleichmäßige Skalierung der  $u$ - und  $v$ -Achse ausgeglichen werden.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} x_p \\ s_y y_p \end{pmatrix} + \begin{pmatrix} u_H \\ v_H \end{pmatrix} \quad (3.3)$$

Damit ist ein minimales Lochkameramodell definiert. Zusätzlich können radial-symmetrische, radial-asymmetrische und tangentiale Linsenverzeichnungen durch Potenzreihenentwicklungen berücksichtigt werden, für die radial-symmetrische Verzeichnung z. B. durch

$$\begin{pmatrix} \tilde{x}_p \\ \tilde{y}_p \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} \left( 1 + a_1(r^2 - r_0^2) + a_2(r^4 - r_0^4) + \dots \right) \quad (3.4)$$

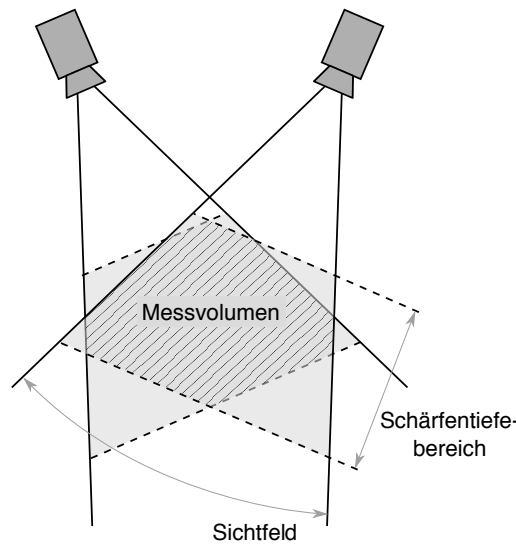
Die Beschreibung der Kameraverzeichnung ist Hauptunterscheidungsmerkmal verschiedener Kameramodelle [31, 56, 66, 129, 143].

**Kamerakalibrierung.** Das Kameramodell kann vereinfacht als 2D→3D-Abbildung mit der inneren und äußeren Orientierung als Parameter dargestellt werden:

$$\begin{pmatrix} u \\ v \end{pmatrix} = f_{cam}(x, y, z; \mathbf{R}, \mathbf{t}, c, h_x, h_y, s_y, \dots) \quad (3.5)$$

Durch bekannte Paare von Bildkoordinaten  $(u_i, v_i)$  und zugehörige Objektkoordinaten  $(x_i, y_i, z_i)$  ergibt sich ein nichtlineares Gleichungssystem, aus dem sich die unbekanntenen Kameraparameter durch ein geeignetes Lösungsverfahren ermitteln lassen. Dieser Vorgang wird als Kalibrierung bezeichnet; meist wird der zugehörige Parametersatz ebenfalls Kamerakalibrierung genannt. Kalibrierungsverfahren sind zusammenfassend z. B. in [116, 147] beschrieben.

**Messbereich.** Im Folgenden soll erörtert werden, wie sich die messtechnischen Größen *Messbereich*, *Auflösung* und *Genauigkeit* aus dem Prinzip der Triangulation ergeben. Das Messvolumen ist im Wesentlichen durch das Sichtfeld der Kameras (*field of view*), den Schärfentiefebereich (*depth of field*) und die Ausrichtung der Messkameras begrenzt (siehe Abbildung 3.6). Das Sichtfeld einer Kamera erhöht sich bei Wahl eines Objektivs mit größerer Brennweite oder eines größeren Kamerasensors. Bei gleichbleibender Sensorauflösung verliert man dadurch jedoch an absoluter Auflösung im 3D-Raum. Der Schärfentiefebereich hängt ab von der Fokussierung (Verhältnis von Bildweite zu Brennweite)



**Abbildung 3.6:** Messvolumen bei Verwendung zweier Kameras

sowie der Blende. Eine größere Blendenzahl bedeutet einen höheren Schärfentiefebereich, aber gleichzeitig auch eine weniger starke Belichtung. Die Kameras sollten so zueinander ausgerichtet sein, dass sie sich ein möglichst großes Sichtfeld und einen großen Schärfentiefebereich teilen.

Prinzipiell kann der Messbereich auch durch das Hinzufügen zusätzlicher Kameras erweitert werden. Durch nicht-ideale Kamerakalibrierung passen überlappende Punktwolken, die in den gemeinsamen Sichtfeldern mehrerer Kameras entstehen, nicht perfekt zusammen. Daher ist eine Strategie zur Behandlung der einzelnen Teilbereiche der Messung notwendig, wie z. B. eine Korrektur des Koordinatensystems mittels *Iterative Closest Point Algorithmus* (ICP) [15, 53].

**Auflösung.** Man unterscheidet zwischen Tiefenauflösung sowie lateraler Auflösung der 3D-Punktwolke. In Anlehnung an Messprinzipien wie z. B. Ultraschall oder Time-of-Flight-(ToF-)Kameras wird damit der kleinste unterscheidbare Messwert entlang der Messrichtung (axial) bzw. quer zur Messrichtung (lateral) bezeichnet. Ist die  $z$ -Achse des Messkoordinatensystems entlang der Kamerablickrichtung ausgerichtet, entspricht die Tiefenauflösung der Auflösung entlang der  $z$ -Achse. Die laterale Auflösung bezeichnet dann den Abstand der Messpunkte in der  $x$ - $y$ -Ebene. Es bietet sich an, die Oberfläche annähernd parallel zu dieser Ebene auszurichten, sodass die laterale Auflösung bei gering gekrümmten Flächen in etwa dem Punktabstand entlang der Oberfläche entspricht. Genau genommen sind die laterale und Tiefenauflösung aber nicht konstant im Messvolumen und werden typischerweise zu den Randbereichen der Kamerasichtfelder schlechter.

Die Zusammenhänge zwischen den verschiedenen Eigenschaften des Messsystems und der lateralen sowie Tiefenauflösung sind komplex. Die *laterale Auflösung* hängt im Wesentlichen ab vom Bildmaßstab und der Pixelgröße der Kameras. Daraus ergibt sich wiederum

eine Abhängigkeit zur verwendeten Brennweite sowie zum Bild- und Objektstand der Kameraoptik bzw. zur Größe und Bildauflösung des Kamerasensors. Weitere Zusammenhänge bestehen zur optischen Auflösung der Kameras (und somit zur Abbildungsschärfe, Tiefenschärfebereich, Blendenöffnung), zur Ausrichtung der Kameras und zur Auflösung des projizierten Musters (und somit zur Auflösung des Projektors, Unschärfe des Projektors, Abbildungsmaßstab des Projektors, Reflexionseigenschaften der Oberfläche). Die *Tiefenauflösung* hängt im Wesentlichen ab von der Basisbreite (Abstand) und der Ausrichtung (Winkel) der Kameras zueinander und der erreichbaren Genauigkeit bei der Bildkorrelation. Diese steht wiederum im Zusammenhang mit Bildmaßstab und Pixelgröße der Kamera. Weitere Zusammenhänge bestehen zur Strukturgröße und Anzahl der verwendeten Muster (z. B. Gray-Code oder Phasen-Schiebe-Verfahren), zu verwendetem Korrelationsverfahren, Auflösung, Kontrast, Unschärfe und Abbildungsmaßstab des Projektors, Farbtiefe der Kameras sowie Reflexionseigenschaften der Oberfläche.

Aus den wechselseitigen Abhängigkeiten ergibt sich eine Unschärfe-Relation zwischen Tiefenauflösung und lateraler Auflösung: Die Tiefenauflösung und laterale Auflösung eines photogrammetrischen Messsystems können bei hinreichend glatter Messoberfläche reziprok gegeneinander getauscht werden. In direkter Konsequenz können die Tiefenwerte benachbarter Punkte gemittelt werden, um eine genauere Tiefeninformation bei schlechterer lateraler Positionierung zu erhalten. Ähnliche Überlegungen können auch für die Genauigkeit angestellt werden.

**Genauigkeit.** Die Genauigkeit eines Messsystems ergibt sich aus der Auflösung des Systems, dem systematischen Messfehler sowie zufälligen Messfehlern. Auflösungslimitierende Systemeigenschaften wurden bereits genannt; sie betreffen Kamerasensoren und -optik, Projektoraufklärung und -optik, Musterprojektionsverfahren und Anordnung der Komponenten und können bei der Auslegung des Systems berücksichtigt werden. Weitere systematische Messfehler resultieren vor allem aus der ungenauen Bestimmung der Kamerakalibrierung, Unzulänglichkeiten des Kameramodells sowie anderer äußerer Einflüsse (z. B. Tageslicht oder externe Lichtquellen, Reflexionseigenschaft der Messoberfläche, Vibration während der Messung).

Zufällige Fehler sind bedingt durch Unzulänglichkeiten der Kamerasensorik (additives Rauschen; nichtlineare, nichtkonstante Sensorkennlinien) und des Projektors (Helligkeitsschwankungen). Diese führen zu einer ungenauen Bestimmung der Bildkoordinaten in der Bildkorrelation und pflanzen sich somit zu ungenauen 3D-Messwerten fort. Zufällige äußere Einflüsse (Instabilität des Messaufbaus, Verunreinigungen und Reflexionen auf der Oberfläche, Übertragungsfehler, Bewegung des Messobjektes) beeinflussen ebenfalls die Genauigkeit der 3D-Messwerte.

Zur Charakterisierung der Genauigkeit von flächenhaft messenden 3D-Messsystemen haben sich drei Kenngrößen etabliert [130]. Die Antastabweichung  $R$ , Kugelabstandsabweichung  $\Delta\ell$  sowie Ebenheitsabweichung  $R_E$ .



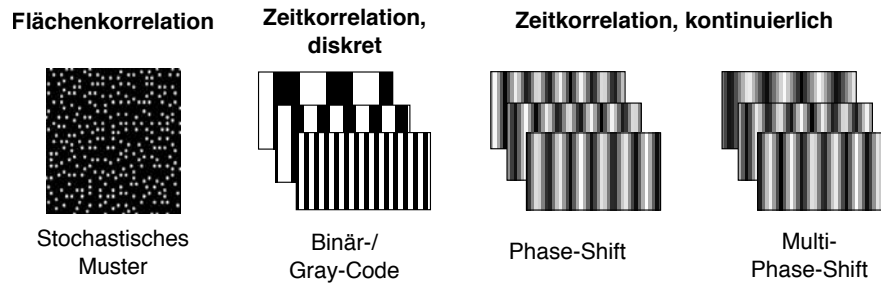


Abbildung 3.7: Beispiele für strukturierte Muster

### 3.2.2 Bildkorrelation

Um zwei korrespondierende Bildpunkte zu finden, wird ein Ähnlichkeitsmaß benötigt, mit dem die Übereinstimmung zweier Bildbereiche bestimmt werden kann. Für die photogrammetrische Anwendung existieren zwei grundlegende Strategien: Die Flächen- sowie die Zeitkorrelation. Im Folgenden werden Methoden aus diesen beiden Kategorien dargestellt. Weiterhin ist eine merkmalsbasierte Korrelation möglich, mit der jedoch keine für die Oberflächeinspektion notwendige flächenhafte Messung möglich ist.

Homogene Oberflächen ohne Texturierung sind merkmalsarm, es ist daher nur sehr schwierig möglich, korrespondierende Punkte auf solchen Oberflächen zu finden. Abhilfe schafft die Projektion von Mustern. Auch bei nicht-homogenen Oberflächen wird die eindeutige Zuordnung dadurch vereinfacht bzw. die erreichbare Genauigkeit erhöht. Zeitkorrelationsverfahren sind prinzipbedingt auf die Projektion einer Musterserie angewiesen. Eine Auswahl verschiedener Muster für die Flächen- und Zeitkorrelation ist in Abbildung 3.7 dargestellt, einen Vergleich verschiedener Musterprojektionen für die 3D-Vermessung findet man in [118], [117] und [144].

**Flächenkorrelation.** Die Flächenkorrelation geht von einem Bildausschnitt (Block) um den Bildpunkt  $(u_i^0, v_i^0)$  eines Referenzbildes  $G^0$  aus. In einer Suchumgebung werden im Vergleichsbild  $G^1$  Blöcke der selben Größe über ein Korrelationsmaß mit dem Referenzblock verglichen. Die korrespondierende Bildposition  $(u_i^1, v_i^1)$  ergibt sich dort, wo das Korrelationsmaß minimal bzw. maximal ist, also die größte Ähnlichkeit zwischen Referenz- und Vergleichsblock besteht.

Es handelt sich um die Anwendung des Blockmatching-Verfahrens. Verbreitete Korrelationsmaße sind z. B. die Summe absoluter Differenzen (SAD) oder die normierte Kreuzkorrelationsfunktion (nKKF):

$$\text{SAD}(u_i^1, v_i^1) = \sum_{u=-d_u}^{+d_u} \sum_{v=-d_v}^{+d_v} \left| G^1(u_i^1 + u, v_i^1 + v) - G^0(u_i^0 + u, v_i^0 + v) \right| \quad (3.6)$$

$$\begin{aligned}
 \text{nKKF}(u_i^1, v_i^1) = & \\
 & \frac{\sum_{u=-d_u}^{+d_u} \sum_{v=-d_v}^{+d_v} G^1(u_i^1 + u, v_i^1 + v) \cdot G^0(u_i^0 + u, v_i^0 + v)}{\sqrt{\left( \sum_{u=-d_u}^{+d_u} \sum_{v=-d_v}^{+d_v} [G^1(u_i^1 + u, v_i^1 + v)]^2 \right) \cdot \left( \sum_{u=-d_u}^{+d_u} \sum_{v=-d_v}^{+d_v} [G^0(u_i^0 + u, v_i^0 + v)]^2 \right)}} \quad (3.7)
 \end{aligned}$$

Alle Verfahren haben als gemeinsame Parameter die Blockgröße  $(2d_u + 1) \times (2d_v + 1)$  sowie den Suchbereich. Die Wahl der Blockgröße hat Auswirkungen auf verschiedene Eigenschaften des Suchalgorithmus:

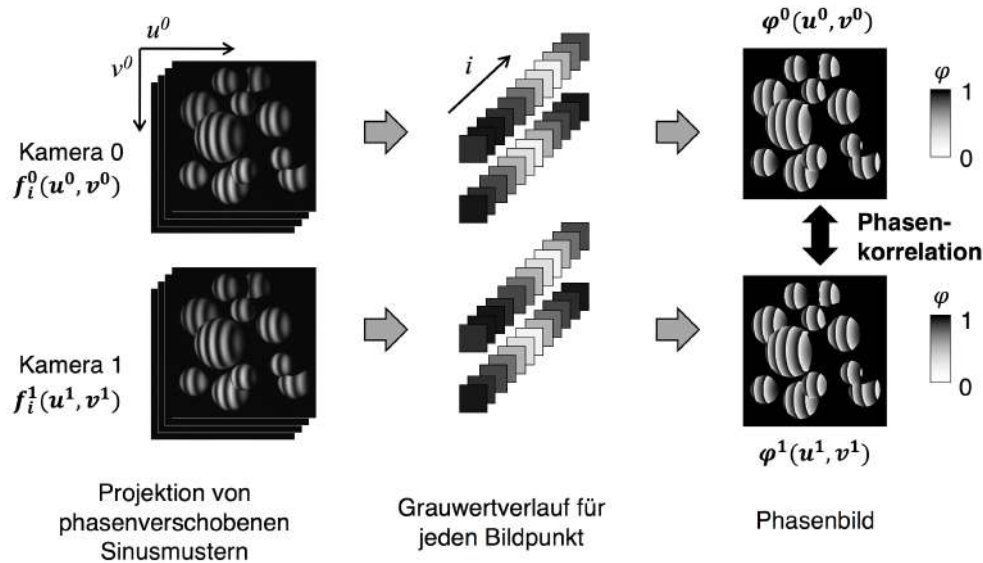
- **Rechenzeit.** Die Rechenzeit skaliert linear mit der Pixelzahl der Blöcke, kleine Blöcke sind zu bevorzugen. Eine Parallelisierung ist möglich.
- **Ortsauflösung.** Die zugehörigen Blöcke benachbarter Bildpunkte  $(u_i^0, v_i^0)$  und  $(u_{i+1}^0, v_{i+1}^0)$  überlappen sich mehr oder weniger, abhängig von der Blockgröße. Dementsprechend sind die Korrelationseigenschaften benachbarter Bildpunkte sehr ähnlich. Die effektive Ortsauflösung sinkt bei größeren Blöcken.
- **Robustheit.** Kleine Blöcke führen möglicherweise zu mehreren Minima bzw. Maxima der Korrelationsfunktion im Suchbereich. Blöcke sollten hinreichend groß sein, um Mehrdeutigkeiten zu vermeiden. Andererseits ändert sich die Abbildung des Objektes abhängig vom Kamera- und Beleuchtungswinkel und der Objektbeschaffenheit; eine Korrelation ist nur in lokalen Bereichen gewährleistet. Blöcke sollten entsprechend nicht zu groß gewählt werden.

Die Abhängigkeit der Abbildung vom Betrachtungswinkel stellt ein grundsätzliches Problem bei der Flächenkorrelation dar. Aufgrund starker Höhenänderungen, Verdeckungen, Glanzstellen, nicht-ortsfestem Speckle oder Reflexion auf der Oberfläche kann eine Korrelation unter Umständen nicht möglich sein.

Durch Projektion geeigneter Muster (z. B. stochastische Muster) auf die Oberfläche kann das Problem der Mehrdeutigkeit bei der Korrelationssuche verringert und mit kleinen Blöcken gearbeitet werden. Dadurch werden auch die genannten Probleme der Rechenzeit, Ortsauflösung und lokaler Abbildungsvariabilität abgeschwächt.

Für die Wahl des Suchbereiches ist es sinnvoll, Vorwissen über die Kameraorientierungen einzubeziehen. Dies führt zu einer starken Einschränkung des Suchbereiches und damit zu einer Minimierung der Rechenzeit (siehe Abschnitt 3.2.3).

**Zeitkorrelation und Phasen-Schiebe-Verfahren.** Nutzt man ein photogrammetrisches Messsystem mit Musterprojektion, kann eine Szene in kurzem zeitlichen Abstand mit unterschiedlichen projizierten Mustern aufgenommen werden. Setzt man voraus, dass sich die Szene zwischen den Aufnahmen nicht verändert, resultiert aus der Bildserie für



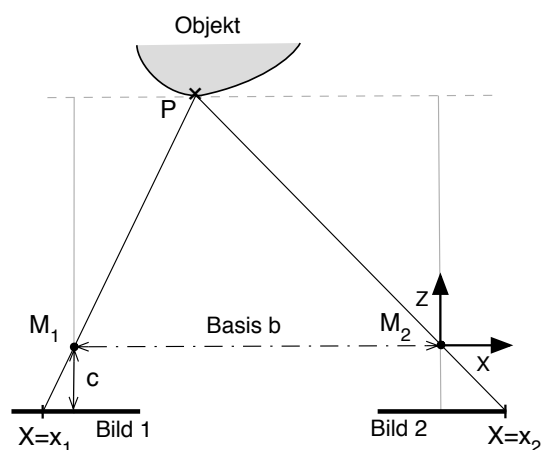
**Abbildung 3.8:** Musterprojektion beim Phasenshiftverfahren und resultierende Phase (in Anlehnung an [6])

Bildbereiche (z. B. bei Streifenprojektion entlang eines Streifens) eine individuelle zeitliche Folge von Grauwerten.

Die Zeitkorrelation nutzt dieselben Korrelationsmaße wie die Flächenkorrelation, nur dass sich die auszuwertenden Pixelblöcke aus der zeitlichen Bildfolge ergeben. Die örtliche Auflösung hängt daher nicht mehr von der Blockgröße ab, sondern von der Auflösung der projizierten Muster und dadurch indirekt von der Auflösung der Kameras, da das Abtasttheorem bei der Bildaufnahme nicht verletzt werden darf. Die örtliche Auflösung ist folglich deutlich höher als bei der Flächenkorrelation. Auch das Problem der unterschiedlichen perspektivischen Verzerrung zweier Kamerabilder kommt hier weniger stark zu tragen.

Das Phasen-Schiebe-Verfahren ist eine Variante der Zeitkorrelation, bei der Streifenmuster mit sinusförmigem Helligkeitsverlauf projiziert werden, die für jedes Bild gleichmäßig verschoben werden. Für jeden Bildpunkt ergibt sich somit ebenfalls zeitlich ein sinusförmiger Helligkeitsverlauf. Die Auswertung erfolgt über die Berechnung des Phasenwertes im Helligkeitsverlauf zu einem definierten Zeitpunkt (siehe Abbildung 3.8). Phasenwerte lassen sich lokal sehr gut bis in den Subpixelbereich interpolieren, da die Phase quer zu den projizierten Streifen kontinuierlich steigt. Weiterhin findet eine Mittelung des Phasenwertes über alle Aufnahmen statt, womit zufällige Fehler (z. B. durch Kamerarauschen, Fehler bei der Projektion, ungleichmäßige Reflexionseigenschaften und Textur der Oberfläche) entsprechend der Anzahl aufgenommener Bilder verringert werden.

Die Phasenkorrelation in den Bildern kann daher mit sehr hoher Ortsauflösung erfolgen, woraus eine hohe laterale 3D-Auflösung und Tiefenaufklärung resultiert. Dies erlaubt eine



**Abbildung 3.9:** Triangulation eines Objektpunktes aus zwei Kameras im Stereonormalfall (Darstellung im Querschnitt)

hochpräzise und flächenhafte Vermessung von Oberflächen, wie sie für die Oberflächeninspektion benötigt wird.

Die Korrelation ist mehrdeutig entsprechend der Periodenlänge der Streifen. Unter gewissen Bedingungen (hinreichend große Streifenbreite, geringe Höhenunterschiede der Oberfläche) lassen sich durch Phase-Unwrapping direkt eindeutige Werte bestimmen [25]. In den meisten Fällen muss die Mehrdeutigkeit durch Projektion weiterer, eindeutiger Muster (z. B. Binär- oder Graycode [119], Phasenshift mit anderer Periodenlänge [84]) oder durch hinreichend gute Näherungswerte für die Oberflächendaten aufgelöst werden (siehe Abschnitt 3.2.3).

### 3.2.3 Methoden zur Punktsuche

Die beschriebenen Korrelationsverfahren stellen Ähnlichkeitsmaße zum Vergleich zweier Bildbereiche bzw. Bildpunkte dar. Zusätzlich bedarf es einer Strategie, welche Bildpunkte überhaupt für den Vergleich in Betracht gezogen werden. Eine naive Suche würde ausgehend von dem Bildpunkt  $(u_i^1, v_i^1)$  im ersten Kamerabild einen Vergleich mit sämtlichen Bereichen des zweiten Kamerabildes durchführen. Offensichtlich ist es sinnvoll, den Suchbereich geeignet einzuschränken. Dazu können die Kameramodelle beider Kameras ausgenutzt werden.

**Stereonormalfall und Rektifizierung.** Der Stereonormalfall ist dadurch gekennzeichnet, dass beide Kameras mit parallelen Sichtachsen ausgerichtet sind. Weiterhin liegen die Kameraabbildungen in derselben Ebene und sind gleich orientiert. Der Zusammenhang ist im Querschnitt in Abbildung 3.9 skizziert.

Dies hat zur Folge, dass die Schnittebenen im Raum, die aus jeweils einer Zeile des ersten und zweiten Kamerabildes entstehen, zusammenfallen. Geht man also von einem Pixel der

ersten Kamera aus, so kann der korrelierende Punkt der zweiten Kamera in exakt einer Zeile gesucht werden (siehe Abbildung 3.10). Dies schränkt den Suchraum im Bild auf lediglich eine Bildzeile ein. Nachdem die erste Punktkorrespondenz in einer Zeile gefunden wurde, kann der Suchraum noch weiter auf die nachfolgenden Bildpunkte eingeschränkt werden.

Der Stereonormalfall ist nur näherungsweise mit sehr hohem Aufwand für die Justierung des Messaufbaus erreichbar [83]. Stattdessen kann aber aufgrund der bekannten Kamerakalibrierung eine *Rektifizierung* der Bilder vorgenommen werden. Dabei werden die Kamerabilder so transformiert, dass die nicht-orthogonale Ausrichtung sowie weitere Bildverzerrungen eliminiert werden. Der vereinfachten Korrespondenzsuche steht hier ein erhöhter einmaliger Aufwand für die Bildtransformation gegenüber.

**Bildbasierte Suche.** Durch die Rektifizierung werden Messwerte verfälscht. Bei der Transformation werden Bildbereiche gestaucht und gestreckt, sodass Bilddaten verloren gehen bzw. durch Interpolation Ungenauigkeiten eingebracht werden. Insbesondere beim Einsatz hochpräziser Verfahren wie z. B. dem Phasen-Schiebe-Verfahren sollte möglichst auf den ursprünglichen Bilddaten gearbeitet werden, um keine zusätzlichen systematischen Messfehler einzubringen.

Bei der bildbasierten Suche werden daher die Bilder nicht verändert. Das Wissen über die Abbildungsgeometrie und Verzeichnung wird stattdessen auf die Suchgerade angewendet; man nutzt die sog. *Epipolargeometrie* der Szene. Ein Punkt  $(u_i^1, v_i^1)$  im ersten Bild dient als Referenz und wird über das Kameramodell als Sehstrahl  $\mathbf{x}_c(u_i^1, v_i^1; t)$  in die Szene projiziert. Die Rückprojektion einzelner Punkte auf dem Strahl  $\mathbf{x}_c$  aus der Szene in das zweite Kamerabild ergibt die Epipolarcurve, entlang derer mögliche Kandidaten für korrespondierende Bildpunkte liegen (siehe Abbildung 3.11).

**Objektbasierte Suche.** Die bildbasierte Punktsuche geht von 2D-Punkten in einem Kamerabild aus. Die resultierenden 3D-Punkte liegen im Allgemeinen unregelmäßig im Weltkoordinatensystem. Ihre Verteilung auf der zu messenden Oberfläche hängt im Wesentlichen von der Ausrichtung der Referenzkamera zum Objekt ab. Oft ist es jedoch sinnvoll, eine gewisse Struktur der 3D-Punktewolke im Weltkoordinatensystem zu fordern. Üblich ist z. B. eine Tiefenkarte mit  $z$ -Werten auf einem äquidistanten  $x$ - $y$ -Raster (siehe 3.3.1).

Die objektbasierte Punktsuche geht daher von initialen 3D-Punkten im Weltkoordinatensystem (z. B.  $x$ - $y$ -Raster) aus. Weiterhin wird eine Suchrichtung (z. B. parallel zur  $z$ -Achse) definiert. Entlang dieser Suchgeraden können nun 3D-Punkte  $(x_i, y_i, z_i)$  in beide Kamerabilder rückprojiziert, und jeweils das Ähnlichkeitsmaß der zugehörigen Bildbereiche bestimmt werden (siehe Abbildung 3.12). Zwischen den 3D-Punkten mit der höchsten Ähnlichkeit der zugehörigen Bilddaten kann nun auf dieselbe Art und Weise verfeinert werden.

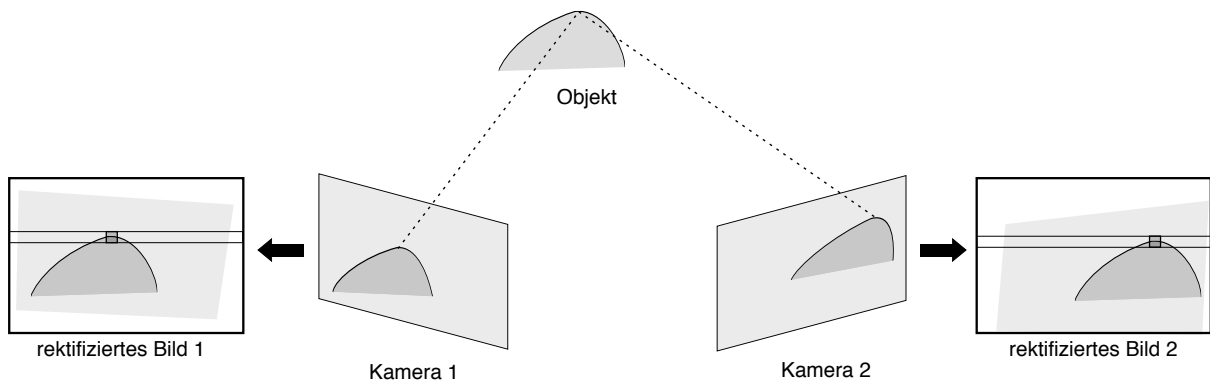


Abbildung 3.10: Rektifizierung von Kamerabildern zur Vereinfachung der Korrespondenzsuche

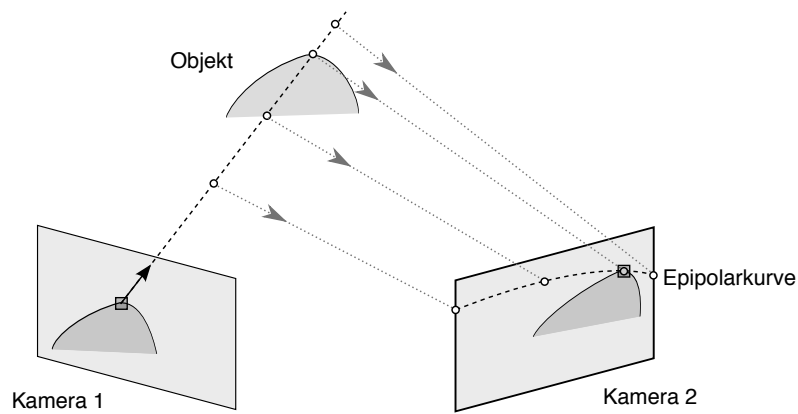


Abbildung 3.11: Nutzung der Epipolargeometrie für die bildbasierte Punktsuche in Kamerabild 2, ausgehend von Kamera 1

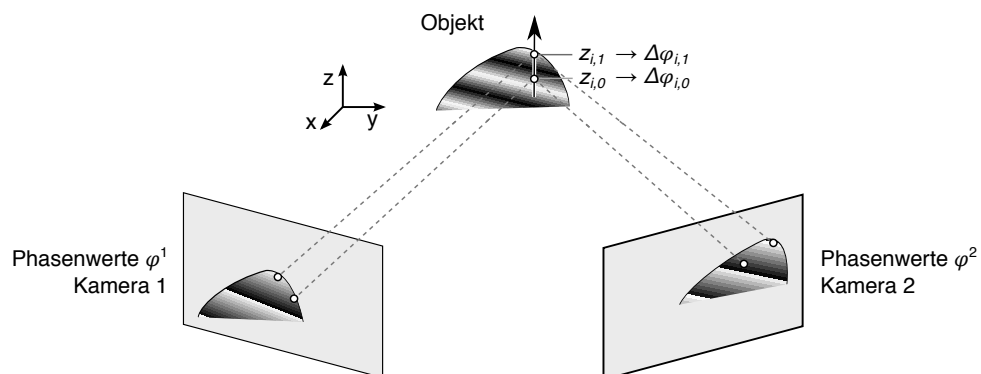


Abbildung 3.12: Objektbasierte Suche in beiden Kamerabildern für das Phase-Shift-Verfahren. Der korrespondierende Punkt weist eine minimale Phasendifferenz  $\|\Delta\varphi\|$  auf

Für das Phase-Shift-Verfahren ist die Ähnlichkeit zweier Bildpunkte über die Phasendifferenz  $\Delta\varphi = \varphi^1 - \varphi^2 \stackrel{!}{\approx} 0$  (d.h. möglichst gleiche Phasenwerte in beiden Bildern) definiert. So lässt sich den 3D-Punkten entlang der Suchgeraden Werte  $\Delta\varphi(x_i, y_i, z_i)$  zuordnen. Der gesuchte 3D-Punkt entspricht der Nullstelle dieser Funktion.

## 3.3 Mathematische Beschreibung von Oberflächen

Je nach Anwendungsfall variieren die Anforderungen an die Oberflächenbeschreibung. Daher kommen im Gesamtsystem abhängig vom Verarbeitungsschritt verschiedene Beschreibungsformen zum Einsatz. Diese werden in den folgenden Abschnitten beschrieben.

Gemessene Flächen werden durch eine *diskrete Oberflächenbeschreibung* repräsentiert (Abschnitt 3.3.1). Für die theoretische mathematische Behandlung werden im folgenden verschiedene *analytische Beschreibungen* eingeführt (Abschnitt 3.3.2). *Freiformflächen* sind eine Klasse von analytischen Flächenbeschreibungen, die für die Geometrische Modellierung im CAD eingesetzt werden (Abschnitt 3.3.3). Als verbreitete Variante werden B-Spline-Flächen in Abschnitt 3.3.4 gesondert betrachtet.

### 3.3.1 Diskrete Beschreibung von Oberflächen

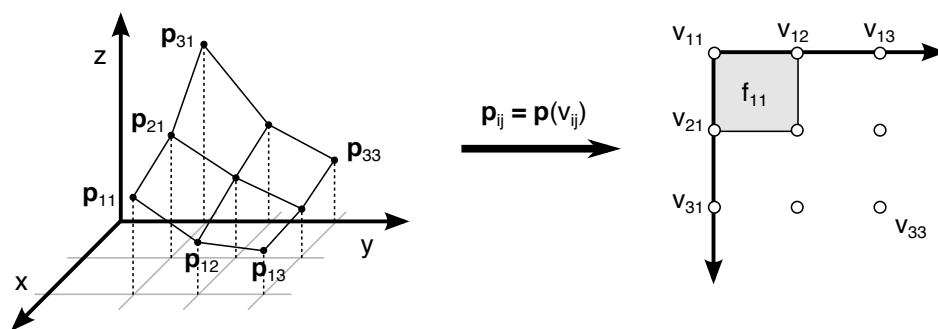
**Punktwolken und Netze.** Das Ergebnis einer 3D-Oberflächenmessung ist eine Menge  $\mathcal{P}$  von Punkten aus dem dreidimensionalen Raum:

$$\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid 1 \leq i \leq N\} \quad \text{mit} \quad \mathbf{p}_i = (x_i, y_i, z_i) \quad (3.8)$$

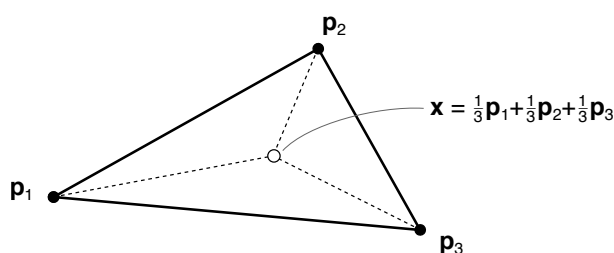
Sie liegen im Messkoordinatensystem vor und beschreiben diskrete Beobachtungen der beobachteten kontinuierlichen Oberfläche. Um die Oberfläche hinreichend gut zu beschreiben, sollte der euklidische Abstand der Punkte klein im Vergleich zur Ausprägung der interessanten Merkmale und der Krümmung der Oberfläche sein.

Im Allgemeinen ist eine Punktwolke ungeordnet, d.h. es liegt zwar die geometrische Information, aber keine topologische Information vor. Die Topologie beschreibt den Zusammenhang zwischen den Punkten  $\mathbf{p}_i$ . Diese kann nicht ohne Weiteres aus einer ungeordneten Punktwolke extrahiert werden, da die Punkte mit geringstem Abstand im Allgemeinen keine Nachbarpunkte sein müssen.

Eine geordnete Punktwolke liegt also durch die Definition von Punkten und ihrer Topologie vor. Eine verbreitete Variante ist das Dreiecks- oder Polygonnetz (*surface mesh*), das als 3-Tupel  $(\mathcal{P}, \mathcal{V}, \mathcal{F})$  beschrieben werden kann. Der Zusammenhang der einzelnen 3D-Punkte in  $\mathcal{P}$  wird durch einen zugrundeliegenden ungerichteten Graphen bestimmt,



**Abbildung 3.13:** Visualisierung von 3D-Punkten, Vertices und Polygonen eines Polygonnetzes mit Vierecksflächen



**Abbildung 3.14:** Beispiel für baryzentrische Koordinaten  $(\alpha_1, \alpha_2, \alpha_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$  und drei Punkten  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$

der durch Knoten  $\mathcal{V}$  (*vertices*) sowie Kanten, Dreiecke oder Polygone  $\mathcal{F}$  definiert ist. Die Vertices sind dabei lediglich eine Zuordnung zu den 3D-Punkten:

$$\mathcal{V} = \{v_i \mid 1 \leq i \leq N\} \quad \text{mit} \quad \mathbf{p}_i = \mathbf{p}(v_i), \quad (3.9)$$

deren Verbindung untereinander durch die Menge  $\mathcal{F}$  definiert wird:

$$\mathcal{F} = \{f_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V} \mid 1 \leq i \leq F\} \quad (\text{z. B. Dreiecksflächen}) \quad (3.10)$$

Genau genommen ist ein Polygonnetz – im Gegensatz zur geordneten Punktwolke – eine stückweise stetige Oberflächenbeschreibung, da jeder Oberflächenpunkt durch Interpolation der angrenzenden Vertices ausgewertet werden kann. Für ein Dreieck  $f_x = (v_1, v_2, v_3)$  eines Dreiecksnetzes ergibt sich ein Oberflächenpunkt  $\mathbf{x}$  als baryzentrische Kombination der zugehörigen Dreieckspunkte (siehe auch Abbildung 3.14):

$$\begin{aligned} \mathbf{x}_{tri} &= \alpha_1 \mathbf{p}_1 + \alpha_2 \mathbf{p}_2 + \alpha_3 \mathbf{p}_3 & (3.11) \\ \text{mit} \quad &\alpha_1 + \alpha_2 + \alpha_3 = 1 \\ \text{und} \quad &\mathbf{p}_1 = \mathbf{p}(v_1), \quad \mathbf{p}_2 = \mathbf{p}(v_2), \quad \mathbf{p}_3 = \mathbf{p}(v_3) \end{aligned}$$



Ist eine ungeordnete Punktwolke  $\mathcal{P}$  gegeben, so kann ein mögliches Netz durch Triangulierung (z. B. Delaunay-Triangulierung) ermittelt werden. Zur Visualisierung eines Netzes wird die 3D-Oberfläche mittels Zentralprojektion auf ein 2D-Bild abgebildet und mittels Shading die zugehörigen Farben im Bild ermittelt. Dazu müssen die Abbildungseigenschaften einer virtuellen Kamera sowie von Lichtquellen definiert werden.

Das verbreitetste Dateiformat zum Datenaustausch von Dreiecksnetzen sind STL-Dateien (*surface tessellation language*); sie enthalten für jedes Dreieck die 3D-Koordinaten der zugehörigen Knoten sowie die Oberflächennormale des Dreiecks. Für weiterführende Ausführungen zu Dreiecksnetzen und deren Verarbeitung sei z. B. auf [23] verwiesen.

**Tiefenkarten.** Als Sonderfall von Polygonflächen können die Vertices in Form von Viereckflächen verbunden, und die  $x$ - $y$ -Komponenten der zugehörigen 3D-Punkte als Gitter definiert werden (siehe auch Abbildung 3.13):

$$\mathcal{P} = \{\mathbf{p}_{i,j} \mid 1 \leq i \leq N, 1 \leq j \leq M\} \quad (3.12)$$

$$\text{mit } \mathbf{p}_{i,j} = (x_i, y_j, z_{i,j})$$

$$\mathcal{V} = \{v_{i,j} \mid 1 \leq i \leq N, 1 \leq j \leq M\} \quad (3.13)$$

$$\text{mit } \mathbf{p}(v_{i,j}) = \mathbf{p}_{i,j}$$

$$\mathcal{F} = \{f_{k,l} \mid 1 \leq k \leq N-1, 1 \leq l \leq M-1\} \quad (3.14)$$

$$\text{mit } f_{k,l} = (v_{k,l}, v_{k+1,l}, v_{k+1,l+1}, v_{k,l+1})$$

Im Speziellen können die Gitterpunkte in  $x$ - $y$  in einem äquidistanten Raster gegeben sein, dann sind Koordinaten durch einen Startwert  $x_0, y_0$  und ein Inkrement  $d_x, d_y$  eindeutig definiert:

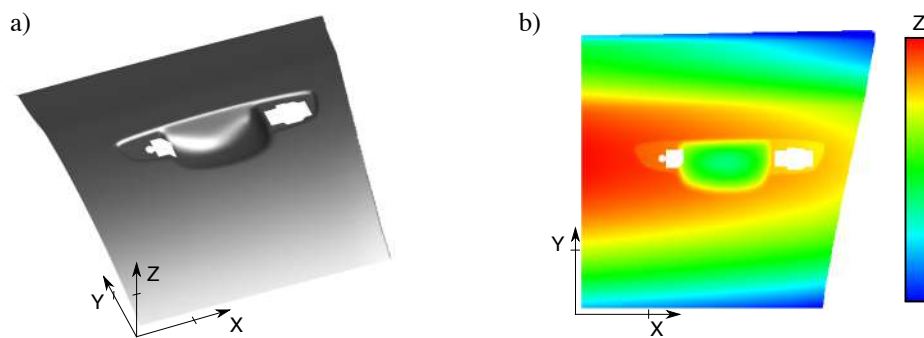
$$x_i = x_0 + (i-1)d_x \quad \forall 1 \leq i \leq N \quad (3.15)$$

$$y_j = y_0 + (j-1)d_y \quad \forall 1 \leq j \leq M \quad (3.16)$$

Die wesentliche Oberflächeninformation steckt nun in den Tiefenwerten  $z_{i,j}$ . Diese können in Form einer Matrix

$$\mathbf{z} = \{z_{i,j} \in \mathbb{R} \mid 1 \leq i \leq N, 1 \leq j \leq M\} \quad (3.17)$$

gespeichert werden, die als Tiefenkarte bezeichnet wird. Dabei werden wesentliche Einschränkungen an die Anordnung der 3D-Punkte gestellt und die Nachbarschaftsbeziehungen der Vertices stark vereinfacht. Die Tiefenkarte eignet sich daher insbesondere für Oberflächen, die im Vergleich zur lateralen Ausdehnung eine geringe Grundkrümmung bzw. keine großen Krümmungsbereiche besitzen und keine Überlappung entlang der  $z$ -Achse enthalten.



**Abbildung 3.15:** Darstellung der Tiefenkarte einer dreidimensional vermessenen Oberfläche „Türgriffmulde“ a) als gerenderte 2D-Abbildung, b) in Falschfarbendarstellung

Aus der Darstellung als Tiefenkarte ergeben sich jedoch wesentliche Vorteile:

- Die Speicherung der 3D-Daten ist vergleichsweise effizient.
- Die Tiefenkarte kann leicht interpretiert und visualisiert werden.
- Die Verarbeitung ist sehr einfach möglich mit Matrix- und Faltungsoperationen.
- Es lassen sich viele Methoden aus der 2D-Bildverarbeitung adaptieren.

Zusätzlich zum Rendering als 2D-Bild können Tiefenkarten auch in Falschfarbendarstellung visualisiert werden. Dazu wird ein kontinuierlicher Farbverlauf (z. B. blau-grün-rot) definiert und einem Bereich von Tiefenwerten zugeordnet. Es ergibt sich ein 2D-Bild, bei dem sich die Farbe jedes Pixel aus dem entsprechenden Wert der Tiefenkarte ergibt (siehe Abbildung 3.15).

**Maskierung von Tiefenkarten.** Bei der Beschreibung mittels Tiefenkarte wird zunächst davon ausgegangen, dass sämtliche Werte  $z_{i,j}$  im Gitter definiert sind. Dies ist an Randbereichen der physikalischen Oberfläche, z. B. an „Löchern“, Ausstanzungen und Kanten nicht der Fall (siehe auch Abbildung 3.15). An den zugehörigen Gitterpositionen  $(x_i, y_j)$  ist dann kein gültiger  $z_{i,j}$ -Wert definiert, d.h. das entsprechende Matrixelement muss geeignet markiert werden.

Dafür kann ein reservierter Nicht-Zahlenwert (NaN, *Not a Number*) verwendet werden. Viele Berechnungsverfahren scheitern aber an der Behandlung von NaN-Werten. Daher ist es sinnvoll, ungültige Positionen  $(i, j)$  der Matrix mit einem beliebigen Zahlenwert (z. B. 0) zu versehen und diese Positionen in einer binären Matrix

$$\mathbf{m} = \{m_{i,j} \in \{0, 1\} \mid 1 \leq i \leq N, 1 \leq j \leq M\} \quad (3.18)$$

zu vermerken. Die so markierten Gitterpositionen können dann bei Bedarf separat behandelt werden.

Die Maskierung  $\mathbf{m}$  stellt eine Projektion der Oberfläche auf die  $x$ -/ $y$ -Ebene dar, in der Darstellung kann man die „Umrisse“ des Objekts erkennen. Die Genauigkeit für die Lokalisierung der Umrisse und damit der Oberflächenränder ist allerdings von der Auflösung des zugrundeliegenden  $x$ - $y$ -Gitters abhängig. Diese ist üblicherweise deutlich schlechter als die Auflösung der  $z$ -Koordinate.

Durch das Messsystem bedingt kommt es an Randbereichen außerdem oft zu systematischen Fehlern, sodass eine Auswertung der Daten dort oft nicht sinnvoll ist. Daher können verschiedene Operatoren auch auf die Maskierung  $\mathbf{m}$  angewendet werden, z. B. eine morphologische Erosion zur Verkleinerung des maskierten Randbereiches (siehe Abschnitt 3.5).

**Distanz- und Defektkarten.** In dieser Arbeit wird zudem mit Distanz- und Defektkarten gearbeitet. Die Speicherung, Verarbeitung und Visualisierung mit Falschfarben erfolgt genau wie bei Tiefenkarten. Sie beschreiben allerdings nicht direkt eine Oberfläche, d. h. es liegt keine vergleichbare Polygonfläche zugrunde. Stattdessen beschreibt eine Distanzkarte die punktwisen Abstände zwischen zwei Oberflächen; sie ist also stets auf eine Referenzoberfläche bezogen (im Gegensatz zur Tiefenkarte, die auf ein äquidistantes  $x$ - $y$ -Raster bezogen ist). Defektkarten (*defect maps*) sind nichts anderes als weiterverarbeitete Distanzkarten, die möglicherweise Defekte enthalten und zur Defekterkennung genutzt werden.

Für Distanzkarten kann die Angabe einer Standardabweichung sinnvoll sein:

$$\sigma_z = \sqrt{\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (z_{i,j} - \bar{z})^2} \quad \text{mit} \quad \bar{z} = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M z_{i,j} \quad (3.19)$$

$\sigma_z$  gibt eine mittlere Abweichung von der Referenzfläche an; dies kann je nach Zustandekommen der Distanzkarte ein Maß für den mittleren Oberflächen- oder Modellfehler, für den Messfehler oder die Ähnlichkeit zweier Oberflächen sein.

Wichtige Methoden zur Weiterverarbeitung von Tiefen- und Distanzkarten werden in Abschnitt 3.5 dargestellt.

### 3.3.2 Analytische Beschreibung von Oberflächen

Die in Abschnitt 3.3.1 dargestellten Polygonflächen und Tiefenkarten sind zwar stückweise stetig, jedoch nicht stetig differenzierbar. Dies ist problematisch, da eine mindestens einfache Differenzierbarkeit für die Berechnung einer Oberflächennormale nötig ist. Diese wird von vielen Algorithmen für weiterführende Berechnungen vorausgesetzt.<sup>2</sup>

<sup>2</sup>Es existieren eine Reihe von Näherungsverfahren für die Normalenberechnung aus der Computergrafik. Eine „echte“ stetige Beschreibung ist für die Anwendung in der Oberflächeninspektion jedoch zu bevorzugen.

Man spricht von  $C^n$ -*Stetigkeit*, wenn die Oberfläche  $n$ -fach stetig differenzierbar ist. Eine  $C^\infty$ -stetige Oberfläche wird auch als *glatt* bezeichnet. Oftmals ist eine *hinreichend glatte* Oberfläche ausreichend, also eine  $C^n$ -stetige Oberfläche bei der  $n < \infty$  hinreichend groß ist. Die konkrete Wahl von  $n$  hängt vom jeweils vorliegenden Anwendungsfall ab.

Analytische Kurven und Flächen werden nach impliziter, expliziter und parametrischer Darstellung unterschieden [13, 34].

**Implizite und explizite Flächenbeschreibung.** Die funktionelle Darstellung einer der Koordinaten in Abhängigkeit der verbleibenden Koordinaten wird auch als *explizite Flächenbeschreibung* bezeichnet. Ein Beispiel sind Höhenfelder der Form

$$z = f(x, y). \quad (3.20)$$

Die explizite Flächenbeschreibung findet vor allem bei der Beschreibung geometrischer Formen wie Zylinder oder Kugel Anwendung. Sie sind in ihrer Beschreibungsfähigkeit im Vergleich zu impliziter und parametrischer Darstellung eingeschränkt. So kann mit der Darstellung (3.20) z. B. keine Fläche parallel zur  $z$ -Achse beschrieben werden. Folglich ist zwar eine Überführung der expliziten Darstellung in eine andere analytische Beschreibung stets möglich, andersherum ist dies nicht zwingend möglich.

Bei der *impliziten Flächenbeschreibung* werden die Koordinaten nicht in direkten (expliziten) Bezug zueinander gesetzt, sondern indirekt bzw. implizit in der Form

$$f(x, y, z) = 0. \quad (3.21)$$

Mit der impliziten Beschreibung lassen sich alle unbegrenzten Flächen im 3D-Raum darstellen. *Algebraische Flächen* werden über ein Polynom der Koordinaten  $x, y, z$  beschrieben:

$$f_n(x, y, z) = \sum_{0 \leq i+j+k \leq n} a_{ijk} x^i y^j z^k = 0. \quad (3.22)$$

Diese lassen sich z. B. für die einfache Flächenapproximation einsetzen.

Einige Auswertungen lassen sich in impliziter Form sehr effizient durchführen. Insbesondere kann für einen gegebenen Punkt durch Einsetzen sehr schnell ermittelt werden, ob er in der Fläche liegt. Dies kann bei geeigneter Formulierung der impliziten Beschreibung (z. B. *Signed Distance Function*) auf eine sehr effiziente Abstandsberechnung von Punkten verallgemeinert werden. Auf diesen Eigenschaften bauen *Level-Set-Methoden* zur numerischen Analyse von Flächen auf [100, 145].

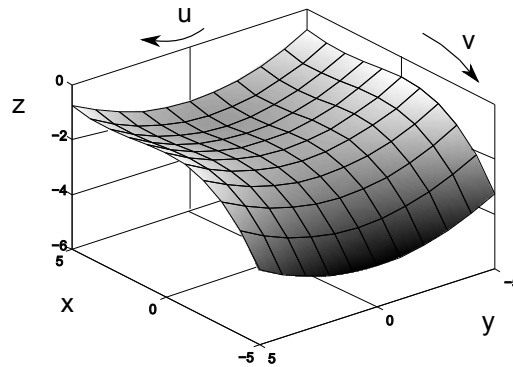


Abbildung 3.16: Oberfläche mit isoparametrischen Linien

**Parametrische Beschreibung.** Oft ist es sinnvoll, die einzelnen Koordinaten  $x, y, z$  einer Fläche in Abhängigkeit von Parametern  $u, v$  darzustellen:

$$\mathbf{x}_s = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} \quad (u, v) \in [a, b] \times [c, d] \subset \mathbb{R}^2 \quad (3.23)$$

Es handelt sich dabei also um eine Abbildung  $\mathbb{R}^2 \rightarrow \mathbb{R}^3$  aus einem 2D-Parameterraum in den 3D-Objektraum. Im Gegensatz zur impliziten und expliziten Beschreibung können die Parameter  $u, v$  durch die Intervalle  $[a, b]$  bzw.  $[c, d]$  ein- oder zweiseitig beschränkt werden, womit sich auch abgeschlossene Flächen im Raum darstellen lassen.

Für  $u = \text{const}, v \in \mathbb{R}$  bzw.  $v = \text{const}, u \in \mathbb{R}$  ergeben sich Kurven, die auf der Fläche liegen (*isoparametrische Kurven*). Eine Kurve im 3D-Raum lässt sich also darstellen als<sup>3</sup>:

$$\mathbf{x}_c = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x(u) \\ y(u) \\ z(u) \end{pmatrix} \quad u \in [a, b] \subset \mathbb{R} \quad (3.24)$$

In Abbildung 3.16 sind isoparametrische Kurven in regelmäßigen Abständen von  $u$  bzw.  $v$  auf einer Fläche dargestellt. Es lässt sich gut erkennen, wie die Parameter entlang der 3D-Fläche verlaufen. Die parametrische Darstellung erlaubt eine sehr intuitive geometrische Deutung, so lassen sich zum Beispiel natürliche Richtungen entlang der Fläche oder regelmäßige Raster auf der Fläche einfach im Parameterraum definieren. Punkte auf der Fläche lassen sich (im Gegensatz zur impliziten Darstellung) sehr leicht berechnen. Meist besitzen die Koeffizienten der parametrischen Beschreibung eine spezifische geometrische Bedeutung. Weiterhin können Flächen im Gegensatz zur expliziten Darstellung beliebig im

<sup>3</sup>Es sei bemerkt, dass die implizite und explizite Darstellung keine Beschreibung von Kurven in 3D erlaubt.

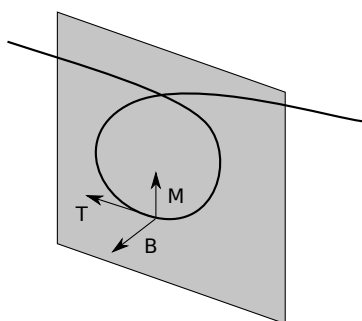


Abbildung 3.17: Begleitendes Dreibein nach FRENET

Raum liegen (z. B. können mehrere  $z$ -Werte zu einer  $x$ - $y$ -Position problemlos dargestellt werden).

**Differentialgeometrie parametrischer Kurven und Flächen.** Parameter können prinzipiell durch beliebige, streng monotone Funktionen ersetzt werden, weshalb unendlich viele mögliche Parametrisierungen für ein und dieselbe geometrische Fläche bzw. Kurve denkbar sind. Insbesondere sei hier die *Parametrisierung auf Bogenlänge* von Kurven erwähnt. Die Länge  $L$  einer Kurve wird berechnet durch [10]:

$$L|_a^b = \int_a^b \|\dot{\mathbf{x}}_c(u)\| du \quad , \quad (3.25)$$

mit der Ableitung

$$\dot{\mathbf{x}}_c(u) = \frac{\partial \mathbf{x}_c(u)}{\partial s} \quad (3.26)$$

Somit lässt sich auch eine Parametrisierung  $u$  auf Bogenlänge finden für die  $\Delta u = u_2 - u_1 = L|_{u_1}^{u_2}$  gilt. Für solche Kurven lassen sich spezielle Vektoren über die Ableitungen definieren:

$$\text{Tangentenvektor} \quad \mathbf{T}(u) := \dot{\mathbf{x}}_c(u) \quad (3.27)$$

$$\text{Hauptnormalenvektor} \quad \mathbf{M}(u) := \frac{\ddot{\mathbf{x}}_c(u)}{\|\ddot{\mathbf{x}}_c(u)\|} \quad (3.28)$$

$$\text{Binormalenvektor} \quad \mathbf{B}(u) := \mathbf{T}(u) \times \mathbf{M}(u) \quad (3.29)$$

Diese Vektoren werden auch als begleitendes Dreibein (*frenet frame*) bezeichnet und sind beispielhaft für einen Kurvenpunkt in Abbildung 3.17 dargestellt. Sie sind offensichtlich nur dann in allen Punkten definiert, wenn die Kurve zweimal differenzierbar ist ( $C^2$ -Kurve) und keine W-Punkte besitzt ( $\ddot{\mathbf{x}}_c(u) \neq 0$ ).

Setzt man weiterhin eine  $C^3$ -Kurve voraus, gelten die FRENET-SERRETSchen Ableitungsgleichungen

$$\begin{pmatrix} \dot{\mathbf{T}}(u) \\ \dot{\mathbf{M}}(u) \\ \dot{\mathbf{B}}(u) \end{pmatrix} = \begin{bmatrix} 0 & \kappa(u) & 0 \\ -\kappa(u) & 0 & \tau(u) \\ 0 & -\tau(u) & 0 \end{bmatrix} \begin{pmatrix} \mathbf{T}(u) \\ \mathbf{M}(u) \\ \mathbf{B}(u) \end{pmatrix} \quad (3.30)$$

Die Parameter Krümmung  $\kappa = \|\ddot{\mathbf{x}}(u)\|$  und Torsion  $\tau = -\mathbf{M}(u)\dot{\mathbf{B}}(u)$  definieren die Kurve abgesehen von der Translation eindeutig [74].

Das begleitende Dreibein kann in der beschriebenen Form auch auf Kurven angewendet werden, die auf einer parametrischen Fläche verlaufen. Man spricht in diesem Fall auch von einem DARBOUX-Frame. Meist sind für die Verarbeitung parametrischer Flächen jedoch nur die *Oberflächennormale* sowie *-tangente* von Interesse. Der Normalenvektor  $\mathbf{n}$  (bzw. Normaleneinheitsvektor  $\mathbf{n}_0$ ) ist definiert als:

$$\mathbf{n} = \frac{\partial \mathbf{x}_s(u, v)}{\partial u} \times \frac{\partial \mathbf{x}_s(u, v)}{\partial v}; \quad \mathbf{n}_0 = \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad (3.31)$$

Die Tangentialebene  $E_t$  ergibt sich aus den beiden Tangenten an den Oberflächenpunkt  $(u_0, v_0)$ :

$$E_t : \quad \mathbf{x}_t(\lambda_1, \lambda_2) = \mathbf{x}_s(u_0, v_0) + \lambda_1 \frac{\partial \mathbf{x}_s}{\partial u} + \lambda_2 \frac{\partial \mathbf{x}_s}{\partial v}; \quad \lambda_1, \lambda_2 \in \mathbb{R} \quad (3.32)$$

**Übergangsbedingungen zwischen parametrischen Kurven.** Komplexe Formen können z.B durch Zusammensetzen zweier beschränkter parametrischer Flächen oder Kurven dargestellt werden. Je nach Anforderung an die Darstellung können verschiedene Bedingungen an die Randbereiche gestellt werden [10]. Diese sollen im Folgenden exemplarisch für zwei Kurven  $x_{c1}(u)$  und  $x_{c2}(v)$  beschrieben werden.

Im einfachsten Fall sollen sich die Randpunkte berühren (segmentierte  $C^0$ -Kurve). Dies wird beschrieben durch die Übergangsbedingung:

$$\mathbf{x}_{c1}(u = u_1) = \mathbf{x}_{c2}(v = v_0) \quad u \in [u_0, u_1]; v \in [v_0, v_1] \quad . \quad (3.33)$$

Weiterhin können Bedingungen an die Differenzierbarkeit gestellt werden, z. B. ergibt sich für stetig differenzierbare Kurven eine segmentierte  $C^1$ -Kurve aus

$$\dot{\mathbf{x}}_{c1}(u = u_1) = \dot{\mathbf{x}}_{c2}(v = v_0), \quad (3.34)$$

und analog für höhere Ableitungen. Für eine segmentierte  $C^2$ -Kurve gilt, dass die begleitenden Dreibeine und Krümmungen sowie Krümmungsmittelpunkte im Übergangspunkt übereinstimmen. Der Übergangspunkt darf kein W-Punkt sein ( $\ddot{\mathbf{x}} \neq 0$ ).

Zusätzlich können geometrische Forderungen gestellt werden, die für die gesamte zusammengesetzte Kurve gelten, z. B. die Minimierung der Biegeenergie [10]

$$\int_a^b \kappa^2(x) dx \rightarrow \min! \quad (3.35)$$

### 3.3.3 Freiformkurven und -flächen

Für das computergestützte Design werden Oberflächenbeschreibungen benötigt, die eine möglichst einfache geometrische Modellierung und Manipulation auch komplexer Flächen erlauben. Dafür sind abschnittsweise definierte parametrische Beschreibungen geeignet, sogenannte *Splines*. Sie sind eine verbreitete Methode zur Beschreibung von Freiformflächen und -kurven [27].

Im Folgenden wird zunächst das Grundkonzept von Binfunktionen (*blending functions*) anhand von Polynom-Splines hergeleitet. Darauf aufbauend wird das Konzept von Kontrollpunkten und das Prinzip wiederholter linearer Approximation (*blossoming*) an BÉZIER-Kurven hergeleitet. Zuletzt werden *B-Spline-Kurven* mit dem Konzept lokal beschränkter Träger dargestellt.

**Polynom-Spline.** Die einzelnen Abschnitte von Polynom-Splines sind als Polynom des Parameters gegeben. Als Kompromiss zwischen Rechenaufwand, Genauigkeit der Interpolation und „Überschwingen“ der angepassten Kurve werden meist Polynome dritten Grades verwendet:

$$\mathbf{x}_c(u) = \mathbf{a}_3 u^3 + \mathbf{a}_2 u^2 + \mathbf{a}_1 u + \mathbf{a}_0 \quad u \in \mathbb{R}; \mathbf{a}_i \in \mathbb{R}^3 \quad (3.36)$$

$$= \mathbf{A} \cdot [t^3 \quad t^2 \quad t \quad 1]^T \quad (3.37)$$

Die Elemente der Koeffizientenmatrix  $A$  können als  $4 \cdot 3 = 12$  Unbekannte des Gleichungssystems (3.37) angesehen werden, wenn man von gegebenen 3D-Punkten  $\mathbf{x}_j$  und zugehörigen Parametern  $u_j$  ausgeht. Fordert man eine  $C^1$ -stetige Kurve entsprechend der Formeln (3.33) und (3.34), so können die Koeffizienten  $\mathbf{a}_i$  durch Vorgabe von zwei Punkten sowie ihrer Ableitungen an den Rändern der Kurve ( $u = 0$  und  $u = 1$ ) gelöst werden. Es ergibt sich also das Gleichungssystem

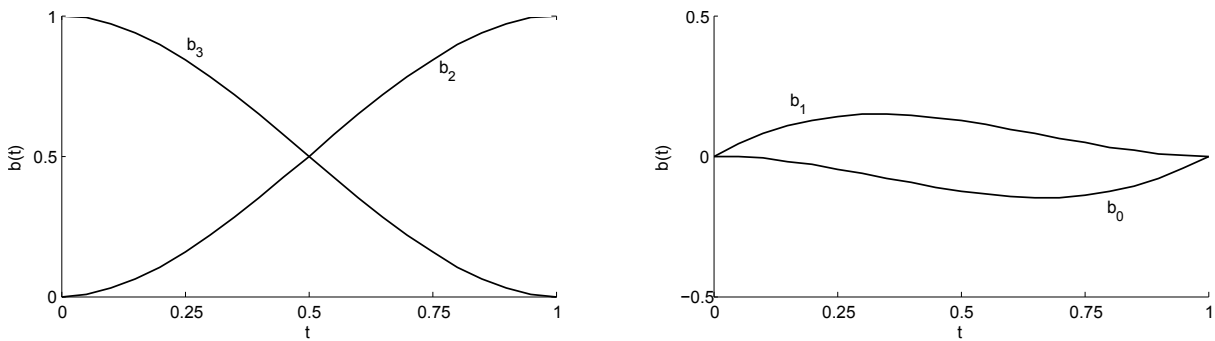
$$\mathbf{x}_c(u = 0) = \mathbf{a}_0 \quad (3.38)$$

$$\mathbf{x}_c(u = 1) = \mathbf{a}_3 + \mathbf{a}_2 + \mathbf{a}_1 + \mathbf{a}_0 \quad (3.39)$$

$$\dot{\mathbf{x}}_c(u = 0) = \mathbf{a}_1 \quad (3.40)$$

$$\dot{\mathbf{x}}_c(u = 1) = 3\mathbf{a}_3 + 2\mathbf{a}_2 + \mathbf{a}_1 \quad (3.41)$$




**Abbildung 3.18:** Binfunktionen einfacher kubischer Splines

oder in Matrixform

$$\begin{aligned}
 \mathbf{G} &= [\mathbf{x}_c(u=0) \quad \mathbf{x}_c(u=1) \quad \dot{\mathbf{x}}_c(u=0) \quad \dot{\mathbf{x}}_c(u=1)] \\
 &= \mathbf{A} \cdot \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}
 \end{aligned} \tag{3.42}$$

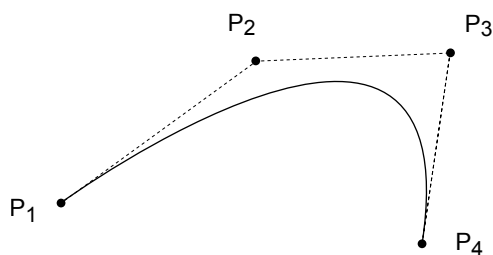
Dabei enthält die sogenannte Geometriematrix  $\mathbf{G}$  die Informationen über die Randpunkte. Löst man diese Gleichung nach der Koeffizientenmatrix  $\mathbf{A}$  auf und setzt sie in die Polynomdefinition (3.37) ein, ergibt sich die Formel für ein kubisches Spline:

$$\mathbf{x}_c(u) = \mathbf{G} \cdot \begin{bmatrix} 2u^3 & -3u^2 & +1 \\ -2u^3 & +3u^2 \\ u^3 & -2u^2 & +u \\ u^3 & -u^2 \end{bmatrix} = \mathbf{G} \cdot \mathbf{b}(u) \tag{3.43}$$

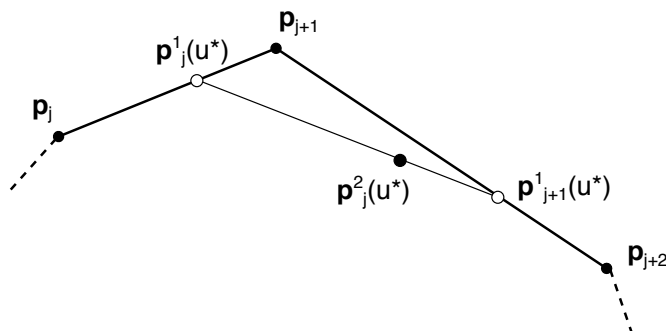
Darin bezeichnet man  $\mathbf{b}(u)$  auch als Binfunktionen (blending functions) oder Basisfunktionen eines Funktionenraumes. Sie sind in Abbildung 3.18 für kubische Splines dargestellt. Jeder Punkt auf der Kurve ergibt sich also aus einer Linearkombination der Basisfunktionen mit der Geometriematrix. Dieses Konzept liegt auch BÉZIER- und B-Splines zu Grunde.

**Bézier-Kurve.** Ein Nachteil von Polynom-Splines ist die geometrische Definition über die Ableitung an den Randpunkten. Dies ist nicht intuitiv interpretierbar. Stattdessen werden für die Geometrie-Matrix einer kubischen BÉZIER-Kurve vier umschreibende Punkte  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$  gewählt, wobei jeweils zwei Punkte die Tangente an den Start- und Endpunkt beschreiben (siehe Abbildung 3.19).

Interessant ist die geometrische Interpretation bzw. Konstruktion der Kurvenpunkte aus den sog. *Kontrollpunkten*  $\mathbf{p}_j$ . So wird das Prinzip der wiederholten linearen Interpo-



**Abbildung 3.19:** Kubische BÉZIER-Kurve und ihre umschreibenden Kontrollpunkte  $\mathbf{p}_i$



**Abbildung 3.20:** Prinzip „Blossoming“ an drei Kontrollpunkten

lation (sogenanntes *Blossoming*-Prinzip) angewendet [110], das auch in Form des DE CASTELJAU-Algorithmus bekannt ist [54]. Es wird im Folgenden anhand von Abbildung 3.20 erklärt.

Um die Kurve an einer bestimmten Parameterposition  $u^* \in [0 \dots 1]$  auszuwerten, wird  $u^*$  als baryzentrische Koordinate zwischen aufeinanderfolgenden Kontrollpunkten  $\mathbf{p}_j$  und  $\mathbf{p}_{j+1}$  aufgefasst. Es ergibt sich daraus ein neuer Punkt  $\mathbf{p}_j^1$ , der im durch  $u^*$  vorgegebenen Verhältnis zwischen den Kontrollpunkten liegt, also

$$\mathbf{p}_j^1 = \mathbf{p}_j + u^* \cdot (\mathbf{p}_{j+1} - \mathbf{p}_j). \quad (3.44)$$

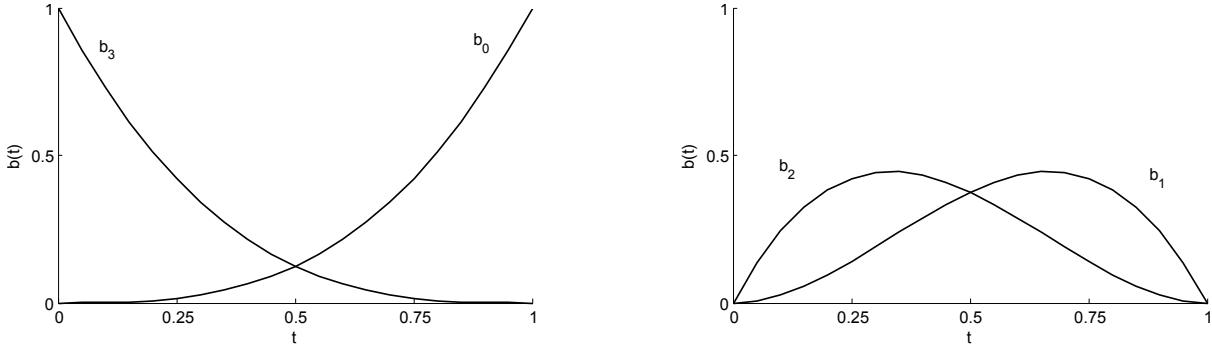
Auf dieselbe Weise können Punkte  $\mathbf{p}_{j+1}^1$  etc. für die folgenden Kontrollpunkte berechnet werden. Für zwei aufeinanderfolgende Punkte dieser neuen Generation kann erneut entsprechend (3.45) linear interpoliert werden, sodass sich der Punkt  $\mathbf{p}_j^2$  ergibt. Verallgemeinert lässt sich ein Punkt der Generation  $\ell$  aus den Vorgängergenerationen berechnen:

$$\mathbf{p}_j^\ell = \mathbf{p}_j^{\ell-1} + u^* \cdot (\mathbf{p}_{j+1}^{\ell-1} - \mathbf{p}_j^{\ell-1}). \quad (3.45)$$

Bei  $\kappa = 4$  Kontrollpunkten lassen sich  $\kappa - 1 = 3$  lineare Interpolationen wiederholen; die verschiedenen Punktgenerationen aus den Interpolationsschritten lassen sich in einem Dreiecksschema darstellen (Tabelle 3.1). Der Punkt  $\mathbf{x}_c(u_i)$  der kubischen BÉZIER-Kurve ergibt sich dann aus der letzten Punktgeneration  $\ell = \kappa - 1 = 3$ .

**Tabelle 3.1:** Dreiecksschema für kubische BÉZIER-Kurve

$\mathbf{p}_1$			
$\mathbf{p}_2$	$\rightarrow \mathbf{p}_1^1(u^*)$		
$\mathbf{p}_3$	$\rightarrow \mathbf{p}_2^1(u^*)$	$\rightarrow \mathbf{p}_1^2(u^*)$	
$\mathbf{p}_4$	$\rightarrow \mathbf{p}_3^1(u^*)$	$\rightarrow \mathbf{p}_2^2(u^*)$	$\rightarrow \mathbf{p}_1^3(u^*)$
	$\ell = 1$	$\ell = 2$	$\ell = 3$



**Abbildung 3.21:** Binfunktionen von kubischen BÉZIER-Kurven

Aus (3.45) lassen sich auch die Binfunktionen für die algebraische Darstellung entsprechend der Form (3.43) ableiten. Diese lassen sich über sogenannte Bernstein-Polynome darstellen:

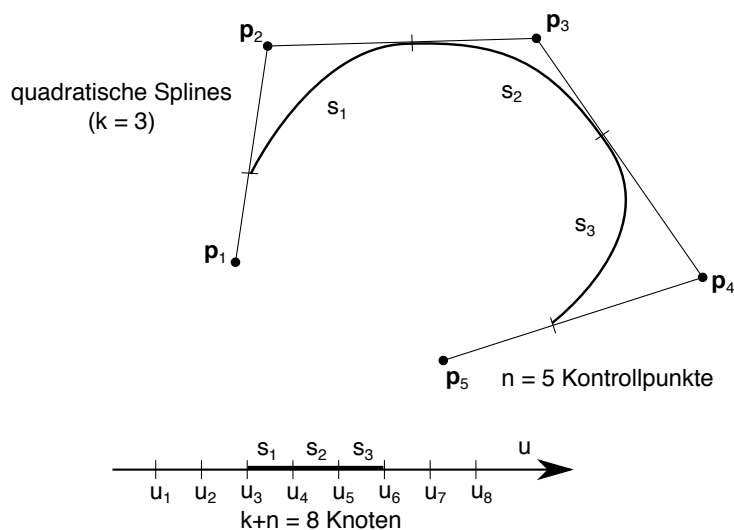
$$b_{j,\kappa}(u) = \binom{\kappa - 1}{j} u^j (1 - u)^{\kappa - j - 1} \quad j = 0, 1, \dots, \kappa - 1 \quad (3.46)$$

Für das Beispiel kubischer BÉZIER-Kurven ( $\kappa = 4$ ) ergeben sich folgende Binfunktionen, die in Abbildung 3.21 dargestellt sind<sup>4</sup>:

$$\mathbf{b}(u) = \begin{bmatrix} (1 - u)^3 \\ 3u(1 - u)^2 \\ 3u^2(1 - u) \\ u^3 \end{bmatrix} = \begin{bmatrix} -u^3 & +3u^2 & -3u & +1 \\ 3u^3 & -6u^2 & +3u & \\ -3u^3 & +3u^2 & & \\ u^3 & & & \end{bmatrix} \quad (3.47)$$

Sollen komplexe Verläufe mit BÉZIER-Kurven dargestellt werden, können mehrere BÉZIER-Kurven unter Berücksichtigung der Randbedingungen (3.33) und (3.34) verknüpft werden. Dadurch wird jedoch die Wahl der Koeffizienten stark eingeschränkt; bei Veränderung eines Kontrollpunktes müssen andere Kontrollpunkte auch angepasst werden, um die Randbedingungen weiterhin zu erfüllen.

<sup>4</sup>Bei einer BÉZIER-Kurve  $\kappa$ -ter Ordnung (d.h. mit  $\kappa$  Kontrollpunkten) ist das zugrundeliegende Polynom vom Grad  $\kappa - 1$ . In der Literatur werden die Bezeichnungen „Ordnung“ und „Grad“ teilweise nicht konsistent verwendet.



**Abbildung 3.22:** Quadratischer B-Spline (Ordnung  $\kappa = 3$ ) mit  $n = 5$  Kontrollpunkten

Eine weitere Lösung ist die Erhöhung des Grades der Kurve und somit auch der Anzahl der Kontrollpunkte. In diesem Falle sind jedoch alle Binfunktionen nach wie vor über den gesamten Parameterbereich definiert. Dies hat zur Folge, dass ein Kontrollpunkt am Anfang der Kurve  $u = 0$  auch Auswirkungen auf den Kurvenverlauf am Ende  $u = 1$  hat.

**B-Spline-Kurven.** Die zuvor genannten Nachteile von BÉZIER-Kurven wurden durch B-Spline-Kurven gelöst. Sie sind eine Erweiterung von BÉZIER-Kurven und nutzen ebenfalls das Blossoming-Prinzip der wiederholten linearen Interpolation zwischen aufeinanderfolgenden Kontrollpunkten. Ein wesentlicher Vorteil von B-Spline-Kurven ist jedoch die bessere lokale Kontrolle über den Kurvenverlauf. Dies wird erreicht, indem von  $n$  Kontrollpunkten immer nur  $\kappa < n$  aufeinanderfolgende in die Interpolation einbezogen werden.

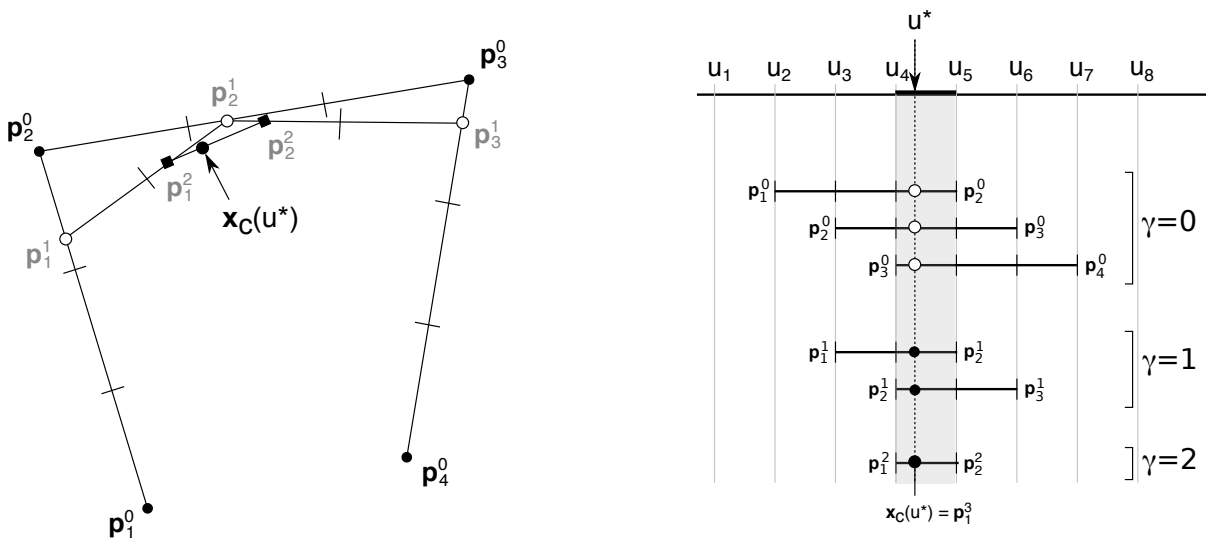
Im Beispiel Abbildung 3.22 werden von 5 Kontrollpunkten  $\mathbf{p}_1 \dots \mathbf{p}_5$  immer  $\kappa = 3$  Punkte zusammengefasst, es ergeben sich also 3 Segmente  $s_1 : \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$ ;  $s_2 : \{\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$ ;  $s_3 : \{\mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5\}$ . Die Zuordnung der Kontrollpunkte zu Kurvenabschnitten geschieht über  $(n + \kappa)$  Knoten  $u_i$  im Parameterraum. Dabei ist der Knotenvektor monoton steigend ( $u_i \leq u_{i+1}$ ). Die Kurve wird nur im Bereich  $u_\kappa \dots u_{n+1}$  aufgespannt; oft werden daher die ersten und letzten Knoten äquivalent gewählt ( $u_1 = \dots = u_\kappa$  und  $u_{n+1} = \dots = u_{n+\kappa}$ ). Dies bewirkt, dass die Kurvenstart- und Endpunkte (anders als in der Abbildung) mit dem ersten und letzten Kontrollpunkt zusammenfallen.

Die dargestellten Segmente  $s_1, s_2, s_3$  entsprechen quadratischen BÉZIER-Kurven. Diese ergeben sich für  $n = \kappa$  und  $u_0 = \dots = u_\kappa = 0$  sowie  $u_{\kappa+1} = \dots = u_{\kappa+n} = 1$  als Sonderfall aus B-Spline-Kurven.

Die geometrische Konstruktion nach dem DE BOOR-Algorithmus [54] ist sehr ähnlich

**Tabelle 3.2:** DE BOOR-Algorithmus zur Konstruktion von B-Splines

<b>gegeben:</b>	
Ordnung	$\kappa$
Kontrollpunkte	$\mathbf{p}_1^0, \mathbf{p}_2^0 \dots \mathbf{p}_n^0$
Knotenvektor	$u_1 \leq u_2 \leq \dots \leq u_{n+\kappa}$
<b>gesucht:</b>	
Kurvenpunkt	$\mathbf{x}_c(u = u^*)$
Schritt 1	Finde Intervall $I$ , in dem $u^*$ liegt: $u_I \leq u^* < u_{I+1}$
Schritt 2	Für Iterationstiefe $j = 0 \dots (\kappa - 2)$ : Für Bereich $i = (I + 1) \dots (I + \kappa - j)$ : $\mathbf{p}_{i-1}^{j+1} = \frac{u_i - u^*}{u_i - u_{i-\kappa+j}} \cdot \mathbf{x}_{i-1}^j + \frac{u^* - u_{i-\kappa+j}}{u_i - u_{i-\kappa+j}} \cdot \mathbf{p}_i^j$
Schritt 3	Rückgabe Kurvenpunkt $\mathbf{x}_c(u^*) = \mathbf{p}_I^{\kappa-1}(u^*)$

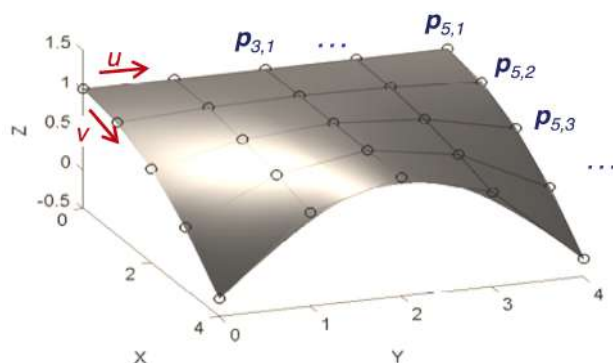


**Abbildung 3.23:** Visualisierung des DE BOOR-Algorithmus für  $n = \kappa = 4$

dem DE CASTELJAU-Algorithmus. Er ist in Tabelle 3.2 algorithmisch beschrieben und in Abbildung 3.23 visualisiert. Dazu sei bemerkt, dass abhängig von der Iterationstiefe  $\ell$  der linearen Interpolation,  $(\kappa - \ell)$  Knoten im Parameterraum für die Interpolation genutzt werden. Die Iteration endet bei  $\ell = \kappa - 2$ , da genau 2 Knoten für eine lineare Interpolation vorhanden sein müssen.

Die Kurvenpunkte liegen im Allgemeinen in der konvexen Hülle des Kontrollpolygons. Für  $\kappa = 2$  entspricht die Kurve dem Kontrollpolygon (d.h. lineare Interpolation) und für  $\kappa = 3$  berührt die Kurve das Kontrollpolygon (z. B. Abbildung 3.22).

Für die algebraische Beschreibung entsprechend (3.43) wird die Geometriematrix  $\mathbf{G}$  aus den Kontrollpunkten  $\mathbf{p}_i$  zusammengesetzt. Die Basisfunktionen  $\mathbf{b}(u)$  werden (bei bekann-



**Abbildung 3.24:** Beispiel für B-Spline-Fläche und zugehörige Kontrollpunkte  $\mathbf{p}_{i,j}$

tem Knotenvektor  $\mathbf{u}$ ) wie folgt definiert:

$$b_i^0(u) = \begin{cases} 1 & u \in [u_i, u_{i+1}[ \\ 0 & \text{sonst} \end{cases} \quad (3.48)$$

$$b_i^j(u) = \frac{u - u_i}{u_{i+j} - u_i} b_i^{j-1}(u) + \frac{u_{i+j+1} - u}{u_{i+j+1} - u_{i+1}} b_{i+1}^{j-1}(u) \quad (3.49)$$

Aus (3.43) lässt sich für B-Spline-Kurven die verbreitete Summenschreibweise ableiten:

$$\mathbf{x}_c(u) = \sum_{i=1}^n \mathbf{p}_i b_i^k(u) \quad (3.50)$$

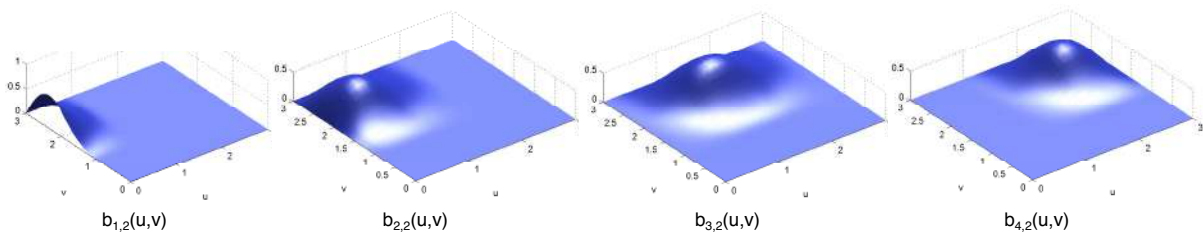
### 3.3.4 B-Spline-Flächen

Die B-Spline-Flächenbeschreibung lässt sich einfach als Tensorprodukt der B-Spline-Kurven entlang zweier parametrischer Richtungen  $u$  und  $v$  ableiten:

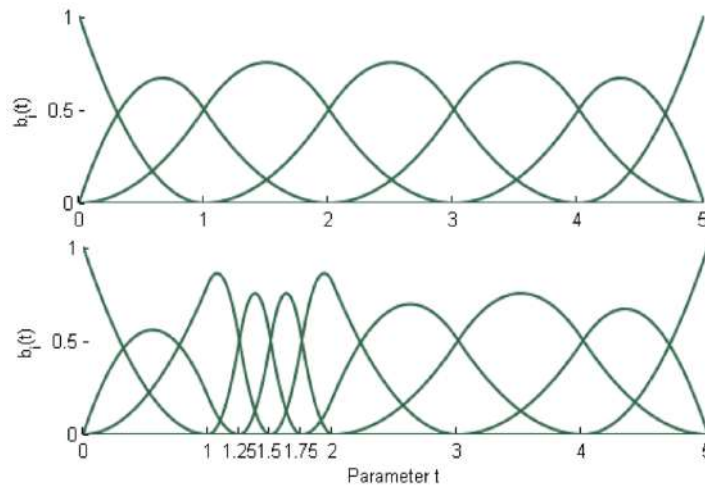
$$\mathbf{x}_s(u, v) = \sum_{i=1}^n \sum_{j=1}^m \mathbf{p}_{i,j} b_i^k(u) b_j^\gamma(v) \quad , \quad (3.51)$$

Die Basisfunktionen sind analog zu (3.49) für die Ordnung  $\kappa \times \gamma$  definiert. Die Fläche wird durch das viereckige Kontrollpolyeder  $\mathbf{p}_{i,j}$  charakterisiert (siehe Abbildung 3.24). Neben der geometrischen Deutung können B-Splines auch als Funktionsapproximatoren im Sinne einer Reihenentwicklung verstanden werden. Dies zeigt sich insbesondere, wenn man die Basisfunktionen in  $u$ - und  $v$ -Richtung zu „Basisflächen“ zusammenfasst:

$$\mathbf{x}_s(u, v) = \sum_{i=1}^n \sum_{j=1}^m \mathbf{p}_{i,j} \cdot b_{i,j}^{\kappa,\gamma}(u, v), \quad (3.52)$$



**Abbildung 3.25:** Beispiel für 2D-Basisfunktionen auf äquidistanten Einheits-Knotenvektoren  $\mathbf{u}, \mathbf{v}$  für kubische Splines  $\kappa = \gamma = 3$



**Abbildung 3.26:** Beispiel für quadratische B-Spline-Basisfunktionen auf äquidistantem Knotenvektor (oben) und mit dichterem Knotenvektor im Bereich  $[1 \dots 2]$  (unten)

also eine gewichtete Summe mit den Kontrollpunkten  $\mathbf{p}_{i,j}$  als Koeffizienten und den 2D-Spline-Basisfunktionen  $b_{i,j}^{\kappa,\gamma}(u, v) = b_i^\kappa(u)b_j^\gamma(v)$ . In Abbildung 3.25 sind Basisfunktionen exemplarisch dargestellt.

**Einfluss der Knotenvektoren.** Der Träger einer Basisfunktion ergibt sich aus den zugehörigen Knoten  $[u_i, \dots, u_{i+\kappa}]$  und  $[v_j, \dots, v_{j+\gamma}]$ . Durch die geeignete Wahl der Knotenvektoren kann also der Einfluss von Kontrollpunkten beeinflusst werden. Liegen die Knoten eng beieinander, so ist der Definitionsbereich der zugehörigen Basisfunktion kleiner und somit auch der Einfluss eines Kontrollpunktes lokaler. Andersherum kann über einen großen Abstand der Knoten der Träger der Basisfunktion vergrößert werden.

Dieser Zusammenhang ist in Abbildung 3.26 für eindimensionale Basisfunktionen dargestellt. Ein dichter Knotenvektorbereich kann folglich starke Variationen der Funktion in diesem Bereich besser darstellen. Bezogen auf die Oberflächendarstellung lassen sich hier starke Krümmungen besser abbilden. Ein großer Knotenabstand eignet sich besser für glatte Oberflächenbereiche. In Abbildung 3.26 lässt sich außerdem der lokale Einfluss der Basisfunktionen erkennen; für die quadratische B-Spline-Basis ( $\kappa = 3$ ) wirken immer nur drei Basisfunktionen gleichzeitig für einen spezifischen Parameterwert (d.h.  $b_i^3(u) > 0$ ).

Der Knotenvektor kann manuell definiert werden (z. B. äquidistant) oder als Teil des Approximationsalgorithmus' bestimmt werden. Um die Ränder der Fläche besser handhaben zu können, werden meist die Randknoten entsprechend der Ordnung der Spline-Ordnung wiederholt. Äquivalente Knoten haben zur Folge, dass die Ränder der Fläche durch die Rand-Kontrollpunkte verlaufen.

**Matrix-Darstellung von Splines.** Zur Auswertung am Rechner ist die Darstellung mit stetigen (rekursiv definierten) Basisfunktionen nicht geeignet. In [109] wird eine Matrix-Darstellung für B-Spline-Kurven und -Oberflächen präsentiert, die sich für die rechen-technische Implementierung anbietet. Die Rekursion entfällt durch Verwendung der Matrix  $\mathbf{M}^\kappa(i)$  und einer Potenz-Vektordarstellung der Quotienten  $\vartheta = \frac{u-u_i}{u_{i+1}-u_i}$  aus Formel (3.49). Für Kurven ergibt sich:

$$\mathbf{x}(u) = \begin{bmatrix} 1 & \vartheta & \vartheta^2 & \dots & \vartheta^{\kappa-1} \end{bmatrix} \cdot \mathbf{M}^\kappa(i) \begin{bmatrix} p_{i-\kappa+1} \\ p_{i-\kappa+2} \\ \vdots \\ p_i \end{bmatrix} \quad (3.53)$$

Für die Berechnung muss bekannt sein, in welchem Knotenabschnitt  $i$  der Parameter  $u$  liegt:  $u \in [u_i, u_{i+1})$ . Für eine B-Spline-Kurve der Ordnung  $\kappa$  ergeben sich dann pro Parameterwert  $\kappa$  Basisfunktionen, die nicht null sind. Anders gesagt zerfällt die Reihenentwicklung in nur  $\kappa$  Basisfunktionen, die die lokalen Eigenschaften um den zu berechnenden Parameterwert  $u$  herum beeinflussen.

Die Matrix  $M^\kappa(i)$  ist zwar wiederum rekursiv definiert, allerdings unabhängig vom konkreten Parameterwert, sodass sie für jeden Knotenabschnitt  $i$  vorbereitet werden kann und nicht zur Laufzeit für einen Parameter berechnet werden muss (siehe Gleichung (5) in [109]).

Für eine B-Spline-Oberfläche in Tensorprodukt-darstellung ergibt sich für die  $x$ -Koordinate:

$$x(u, v) = \vartheta_u \cdot \mathbf{M}_u^\kappa(i) \begin{bmatrix} p_{i-\kappa+1, j-\gamma+1}^x & \dots & p_{i-\kappa+1, j}^x \\ \vdots & \dots & \vdots \\ p_{i, j-\gamma+1}^x & \dots & p_{i, j}^x \end{bmatrix} (\vartheta_v \cdot \mathbf{M}_v^\gamma(j))^T \quad (3.54)$$

$\vartheta_u$  und  $\vartheta_v$  sind die Potenzvektoren der Quotienten der Parameter  $u, v$  und der entsprechenden Knotenpunktvektoren, die Matrizen  $M_u^\kappa \in \mathbf{R}^{\kappa \times \kappa}$  und  $M_v^\gamma \in \mathbf{R}^{\gamma \times \gamma}$  sind die Basismatrizen für B-Splines der Ordnung  $\kappa$  bzw.  $\gamma$  und die  $p_{i,j}$  die  $x$ -Komponente der Kontrollpunkte der B-Spline-Fläche. Die Formeln für die  $y$ - und  $z$ -Koordinaten ergeben sich analog dazu.



## 3.4 Oberflächenmodell und Modellvergleich

Der Modellvergleich ist Kern der modellbasierten Inspektion. Das Oberflächenmodell stellt entsprechend Abbildung 2.2 eine Bezugsfläche dar. Zentrale Aufgabe des Modells ist es somit, neben der geometrischen Oberfläche auch mögliche Formabweichungen der Messung abzubilden. Nur so ist die Erkennung von Oberflächenunvollkommenheiten überhaupt erst ermöglicht. Zusammengefasst lassen sich die folgenden Anforderungen an das Oberflächenmodell ableiten:

### 1) Präzise Darstellung komplexer Oberflächen

- Die geometrische Oberfläche muss mit hinreichender Genauigkeit abgebildet werden. Dies umfasst insbesondere komplexe Oberflächendesigns mit starken lokalen Krümmungen wie z. B. Designkanten.
- Bei Nutzung von Teilmodellen müssen die Übergangsbereiche zwischen zwei Bereichen korrekt und übereinstimmend dargestellt oder geeignet behandelt werden.

### 2) Korrektheit und Sensitivität des Modellvergleichs

- Das Modell muss einen geeigneten Vergleich von Messung und Oberflächenmodell (z. B. Abstandsberechnung) erlauben.
- Der Vergleich muss eine hinreichende Sensitivität haben, typischerweise müssen Defekte von wenigen 10 µm Höhe detektiert werden können.

### 3) Modelladaptivität

- Das Modell muss sich an die individuell vorliegenden Verformungen (z. B. bei Blech- oder Plastikteilen) der Oberfläche anpassen.
- Es muss zulässige Toleranzbereiche abbilden können.

### 4) Robustheit

- Das Modell muss robust gegenüber Messrauschen oder Fehler höherer Ordnung sein.
- Es darf nicht durch lokale Oberflächendefekte beeinflusst werden, da diese detektiert werden sollen.

### 5) Effiziente Berechnung

- Die Anpassung des Modells an die Oberflächenmessung muss hinreichend schnell geschehen.
- Der Vergleich zwischen Messung und Modell (z. B. Abstandsberechnung) muss auch für sehr große gemessene Punktwolken effizient durchgeführt werden können.

Dabei sind einige Anforderungen gegensätzlich, so sollen zwar entsprechend **1)** und **3)** kleinste Oberflächendetails und -variationen möglichst genau dargestellt werden, jedoch dürfen nach **4)** ungewollte kleine Änderungen die Fläche nicht beeinflussen, damit gleichzeitig nach **3)** eine hohe Sensitivität der Fehlererkennung gewährleistet wird.

In Abschnitt 2.3.2 wurde bereits eine Reihe von Methoden vorgestellt. Der Assoziativspeicher nach Lilienblum [87] erfüllt die Anforderungen hinreichend gut und wird in Prüfungssystemen eingesetzt. In Abschnitt 3.4.1 wird seine Funktionsweise sowie Verbesserungspotentiale hinsichtlich der Anforderungen beschrieben. In Abschnitt 3.4.2 wird insbesondere auf Anforderung **2)**, die Abstandsberechnung zwischen Modell und Messung, eingegangen.

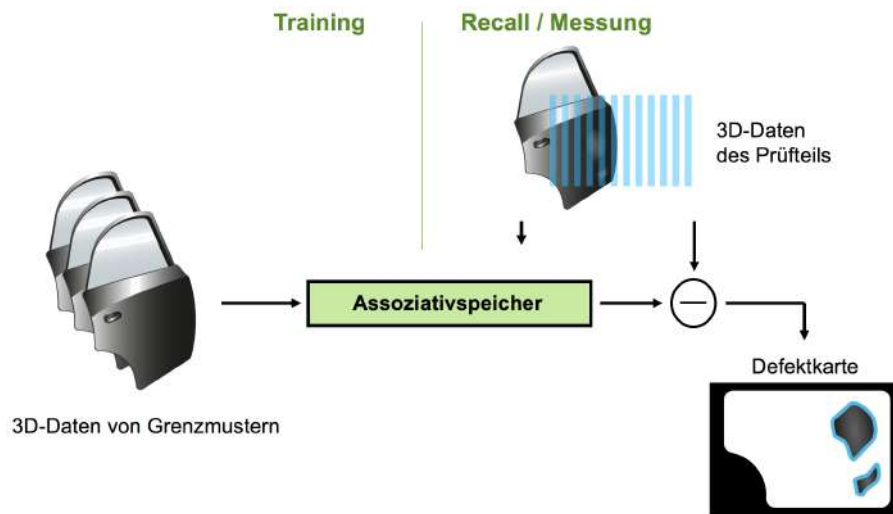
### 3.4.1 Assoziativspeicher als Oberflächenmodell

**Funktionsweise.** In [87] wird der Assoziativspeicher erstmalig als Methode zur Modellierung von Oberflächen vorgeschlagen. Für eine zu prüfende Oberfläche muss ein Trainingsdatensatz, also eine Menge von Messungen verschiedener als „Gutteil“ bewerteter Oberflächen, vorliegen. Diese ergeben sich in der Fertigung von Karosserieteilen z. B. aus der zentralen Analyseprüfung in der Anlaufproduktion (siehe 2.2). Hier werden im Rahmen einer Erstbemusterung Grenzmuster aufgezeichnet, anhand derer die Freigabe der Serienfertigung erfolgt.

Die gemessenen 3D-Punktwolken der Oberfläche werden als vektorisierte Tiefenkarte, also  $z$ -Werte über einem definierten  $x$ - $y$ -Raster, weiterverarbeitet (siehe 3.3.1). Die Art der Vektorisierung ist für den Assoziativspeicher irrelevant, solange sich die Werte auf ein zwischen den Messungen vergleichbares  $x$ - $y$ -Raster beziehen. Dazu werden die Bauteile bei der Messung mechanisch grob ausgerichtet und die Messwerte mittels Registrierungsverfahren merkmalsbasiert feinausgerichtet.

Abbildung 3.27 zeigt die grundlegende Funktionsweise des Assoziativspeichers. In der *Trainingsphase* wird der Assoziativspeicher anhand von registrierten 3D-Oberflächenmessungen fehlerfreier Musterteile angelernt. Dabei werden sogenannte Hauptkomponenten oder Basisvektoren gefunden, die in Linearkombination die Oberflächenvariation in den Trainingsdaten abbilden können. Dieses variable Oberflächenmodell bildet somit die Oberflächenform und ihre zulässigen Varianzen und Toleranzen ab, ohne dass sie formal beschrieben werden müssen.

Beim *Abruf* des Assoziativspeichers, also bei der Inspektion einer gemessenen Oberfläche, werden die 3D-Messwerte ebenfalls zunächst auf die trainierte Oberfläche registriert. Anschließend werden die Tiefenwerte auf die Basisvektoren des Oberflächenmodells abgebildet; es wird eine Linearkombination der Basisvektoren gesucht, mit denen die gemessenen Tiefenwerte mit minimalem Fehler dargestellt werden können. Die Oberflächenform wird dann nur durch die Koeffizienten dieser Linearkombination beschrieben.



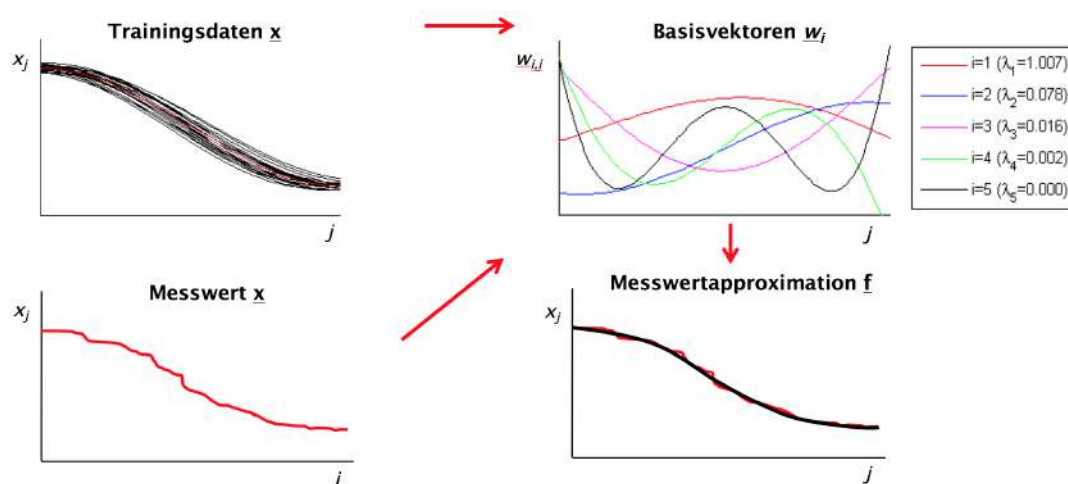
**Abbildung 3.27:** Anwendung des Assoziativspeichers als variables Oberflächenmodell für die Oberflächeninspektion.

Mit dieser Dimensionsreduktion werden „ungewöhnliche“ Varianzen wie Oberflächendefekte und Messrauschen in der Messung nicht abgebildet, sofern sie nicht in den Trainingsdaten vorhanden waren. Oberflächendefekte werden somit effektiv herausgefiltert. Führt man die Rücktransformation durch, erhält man das an die Messung angepasste Modell. Der Vergleich von Messung und angepasstem Modell (siehe Abschnitt 3.4.2) erlaubt so die Segmentierung von kleinsten Fehlerbereichen.

Abbildung 3.28 visualisiert die fehlerkorrigierende Eigenschaft. Hier ist der Eingangsvektor  $\mathbf{x} \in \mathbb{R}^N$  als Menge der Funktionswerte  $x_j$  einer 2D-Kurve auf einem äquidistanten Raster  $j$  dargestellt. Eingangsvektoren werden zunächst um den Mittelwertvektor der Trainingsdaten bereinigt. Die verbleibenden Varianzen werden im Netzwerk durch gewichtete Basisvektoren  $\mathbf{w}_i \in \mathbb{R}^N$  abgebildet (Abbildung 3.28, rechts oben). So ergibt sich beim Abruf mit gestörten Eingangsdaten eine Glättung entsprechend des den Trainingsdaten zugrundeliegenden Modells (Abbildung 3.28, rechts unten).

Als zentraler Parameter des Assoziativspeichers muss die Höhe der Dimensionsreduktion bzw. Anzahl  $N_0$  der Basisvektoren durch den Anwender gewählt werden. Der Parameter  $N_0$  sollte generell so gering wie möglich gehalten werden, um eine möglichst hohe Generalisierungsfähigkeit des Modells zu gewährleisten. Dies bedeutet, dass Spezifika einzelner Messungen und auch zufällige Fehler wie Messrauschen nicht durch das Modell abgebildet werden. Gleichzeitig sollte  $N_0$  so hoch gewählt werden, dass der verbleibende Modellfehler unterhalb einer gewählten Fehlerschwelle bleibt. Dies wird üblicherweise an zusätzlichen Testmessungen der Oberfläche validiert.  $N_0$  ist in jedem Fall kleiner als die Anzahl  $M$  an Messungen, die für das Training verwendet wurden. Für gewöhnliche Karosserieteile wurde zuvor eine Zahl von  $N_0 \approx 20$  empirisch ermittelt [85].

Eine ausführliche mathematische Beschreibung des Assoziativspeichers erfolgt in Ab-



**Abbildung 3.28:** Funktionsweise eines Auto-Assoziativspeichers zur Rekonstruktion am Beispiel einer 2D-Kurve. Eingangsvektor  $\mathbf{x}$  aus Funktionswerten entlang eines äquidistanten Rasters  $j$ . Links oben: Menge der Trainingsdaten und Mittelwertvektor (rot). Rechts oben: Resultierende Gewichtsvektoren der verdeckten Neuronen nach Training (Basisvektoren). Links unten: Gestörter, rauschbehafteter Eingangsvektor  $\mathbf{x}$  für den Recall des Netzwerkes. Rechts unten: Eingangsvektor für den Recall (rot) und durch das Modell geglätteter Ausgangsvektor (schwarz)

schnitt 4.1.

**Bewertung des Assoziativspeichers hinsichtlich der Anforderungen.** Der Assoziativspeicher erfüllt die Anforderungen an ein Modell für die Oberflächeninspektion. Die genaue Darstellung der Oberfläche **1)**, die Modelladaptivität **3)** und Robustheit **4)** sind vor allem von den verwendeten Trainingsdaten abhängig. Für typische Karosserieteile konnten mit ca. 50 Beispielmessungen gute Ergebnisse erzielt, die Modellvariabilität konnte mit etwa 20 Basisvektoren hinreichend gut abgebildet werden [86].

Die Trainingsdaten selbst dürfen keine Defekte enthalten, da diese sonst im Modell abgebildet werden und die Robustheit **4)** beeinflussen können. Die Berechnung zur Laufzeit **5)** ist sehr effizient möglich, da der Modellabruf mit einer einfachen Matrixmultiplikation darstellbar ist. Problematisch ist möglicherweise das Modelltraining mit sehr großen Punktwolken (mehrere Millionen Punkte), da die zugehörigen Matrizen entsprechend groß werden.

Da der Assoziativspeicher mit Tiefenkarten arbeitet, besteht der Modellvergleich **2)** aus der Differenzbildung der  $z$ -Werte. Weicht die Oberflächennormale stark von der Messrichtung ab, entsteht ein systematischer Fehler im Vergleich zur orthogonalen Abstandsberechnung (siehe Abschnitt 3.4.2).

Grundlegende Probleme bei der Korrektheit der Darstellung **1)** treten auf, wenn eine große Oberfläche aus mehreren Teilmessungen und entsprechenden Teilmodellen zusammengesetzt wird. In den Übergangsbereichen zwischen Teilmessungen kommt es meist

zu Klaffungen. Eine Strategie zur Behandlung dieser Übergangsbereiche im Oberflächenmodell ist daher notwendig. Die Ursachen für diese Klaffungen liegen jedoch nicht im Oberflächenmodell begründet, sondern vielmehr in systematischen Fehlern des Messsystems sowie im ungenauen Zusammensetzen durch den Registrierungsalgorithmus.

### 3.4.2 Abstandsberechnung

Der Modellvergleich bedeutet eine Berechnung der Distanz zwischen der gemessenen Oberfläche und dem Oberflächenmodell. Diese hängt von der Darstellung der Oberfläche, der Darstellung des Modells sowie von der gewählten Metrik für die Distanz ab. Voraussetzung ist in jedem Fall, Messung und Modell in ein vergleichbares Koordinatensystem zu bringen. Dazu werden *Registrierungsverfahren* eingesetzt. Sie haben zum Ziel, zwei Oberflächen durch geeignete Transformation möglichst gut miteinander in Deckung zu bringen.

**Punkt-Punkt-Abstand.** Liegen sowohl Messung als auch Modell als Punktwolken vor, so muss für jeden gemessenen Punkt  $\mathbf{x}_{i,j}^m$  zunächst derjenige Modellpunkt  $\mathbf{x}_{k,l}^s$  gefunden werden, für den der euklidische Abstand minimal ist. Der euklidische Punktabstand entspricht dann der gesuchten Distanz zum Modell (siehe auch Abbildung 3.29):

$$d_{i,j} = \|\mathbf{x}_{i,j}^m - \mathbf{x}_{k,l}^s\|_2 \quad (3.55)$$

$$\text{mit } (k, l) : \|\mathbf{x}_{i,j}^m - \mathbf{x}_{k,l}^s\|_2 \leq \|\mathbf{x}_{i,j}^m - \mathbf{x}_{p,q}^s\|_2 \quad \forall (p, q)$$

Die Nächste-Nachbar-Suche ist bei großen Punktwolken sehr rechenintensiv; es existieren Datenstrukturen und Algorithmen, um diese Suche zu beschleunigen (z. B.  $k$ -D-Bäume [14]). Für den Fall, dass beide Punktwolken speziell als Tiefenkarten vorliegen, deren  $x$ - $y$ -Punktraster sich überlappen, lässt sich die Distanzberechnung vereinfachen. Dies ist z. B. bei Verwendung des Assoziativspeichers als Oberflächenmodell möglich. Die Distanz wird dann oft über die Differenz der  $z$ -Koordinaten approximiert (siehe auch Abbildung 3.30):

$$d_{i,j} = |z_{i,j}^m - z_{k,l}^s| \quad \text{mit } k : x_i^m = x_k^s \quad \text{und } l : y_j^m = y_l^s \quad (3.56)$$

Die Berechnung ist deutlich effizienter, da sowohl die Nächste-Nachbar-Suche entfällt, als auch statt der Norm- eine einfache Differenzberechnung stattfindet.

Für beide Punkt-Abstandsberechnungen (siehe Abbildung 3.29 und Abbildung 3.30) weichen die für den Vergleich herangezogenen Modellpunkte vom nächsten Punkt auf der Modelloberfläche ab. Auffällig sind bei der Verwendung der  $z$ -Distanz vor allem Oberflächenbereiche, deren Ausrichtung steil zur  $z$ -Achse verläuft. Die ermittelte Distanz weicht hier stark von der eigentlichen Distanz zur Modellfläche ab. Ein weiteres Problem bei stark geneigten Flächenbereichen ist in dieser Darstellung, dass ein äquidistantes  $x$ - $y$ -Raster

in diesen Bereichen zu stark abweichenden Rasterdistanzen bezogen auf die Oberfläche führt.

**Punkt-Fläche-Abstand.** Das ideale Distanzmaß hinsichtlich der geometrischen Bedeutung ist der Abstand eines gemessenen Punktes  $\mathbf{x}_{i,j}^m$ , orthogonal zur Modelloberfläche  $\mathbf{x}_s(u, v)$  gemessen. Der so ermittelte Lotfußpunkt  $\mathbf{x}_s(u_k, v_k)$  auf der Oberfläche ist gleichzeitig der Punkt der Modelloberfläche mit geringstem Abstand zum Punkt  $\mathbf{x}_{i,j}^m$  (siehe auch Abbildung 3.31):

$$d_{i,j} = \|\mathbf{x}_{i,j}^m - \mathbf{x}_s(u_k, v_k)\|_2 \quad (3.57)$$

$$\text{mit } (u_k, v_k) : \|\mathbf{x}_{i,j}^m - \mathbf{x}_s(u_k, v_k)\|_2 \leq \|\mathbf{x}_{i,j}^m - \mathbf{x}_s(u, v)\|_2 \quad \forall (u, v)$$

Ist das Oberflächenmodell als Punktwolke  $\mathbf{x}_{k,l}^s$  gegeben, kann eine Näherung zur Abstandsdefinition (3.57) bestimmt werden, wenn die zugehörigen Oberflächennormalen  $\mathbf{n}_{k,l}$  bekannt sind. Die lokale Krümmung der Modelloberfläche kann nun berücksichtigt werden, indem der Abstandsvektor auf die Normale projiziert wird:

$$d_{i,j} = \|(\mathbf{x}_{i,j}^m - \mathbf{x}_{k,l}^s) \cdot \mathbf{n}_{k,l}\|_2 \quad (3.58)$$

$$\text{mit } (k, l) : \|\mathbf{x}_{i,j}^m - \mathbf{x}_{k,l}^s\|_2 \leq \|\mathbf{x}_{i,j}^m - \mathbf{x}_{p,q}^s\|_2 \quad \forall (p, q)$$

## 3.5 Auswertung von Tiefenkarten

Die folgenden Filter und Algorithmen sind zum Großteil aus der 2D-Bildverarbeitung entlehnt und werden auf die Matrix der Tiefenwerte  $\mathbf{z} \in \mathbb{R}^{N \times M}$  angewendet. Ziel ist im Allgemeinen die Verbesserung von 3D-Daten oder die Verbesserung einer Distanzkarte, d.h. das Ergebnis der Operation ist ein Bild der selben Größe  $N \times M$ , in dem unerwünschte Merkmale unterdrückt oder erwünschte Merkmale hervorgehoben sind. Die beschriebenen Methoden können zum Teil auch auf die zur Tiefenkarte gehörige Maskierung  $\mathbf{m}$  (siehe Abschnitt 3.3.1) angewendet werden.

**Faltung.** Lineare Filter lassen sich durch einen Filterkern beschreiben. Die Anwendung erfolgt durch diskrete, zweidimensionale Faltung der Tiefen- oder Distanzkarte  $\mathbf{z}$  mit dem Filterkern  $\mathbf{f} \in \mathbb{R}^{(2K+1) \times (2L+1)}$ :

$$\tilde{z}_{i,j} = \sum_{k=-K}^K \sum_{l=-L}^L f_{k,l} \cdot z_{i+k,j+l} \quad (3.59)$$

mit den Einschränkungen

$$\forall K < i \leq N - K, \quad L < j \leq M - L \quad (3.60)$$

$$\nexists m_{i+k,j+l} = 0 \quad (-K \leq k \leq K, -L \leq l \leq L). \quad (3.61)$$

Ein Tiefenwert der gefilterten Tiefenkarte ergibt sich aus der Summe der gewichteten Nachbarwerte, wobei die Gewichtung durch den Filterkern vorgegeben wird. Die Einschränkungen (3.60) und (3.61) sind notwendig, da Formel (3.59) nicht für Randbereiche oder für Bereiche, deren Umgebung ungültige Werte (Maskierung ist „0“) enthält, gilt. Zur Anwendung auf Tiefenkarten wird die Faltung also formuliert als

$$\tilde{z}_{i,j} = \sum_{k \in v_{K,N}(i)} \sum_{l \in v_{L,M}(j)} \frac{f_{k,l}}{\|\mathbf{f} \circ \mathbf{m}_{i,j}\|_1} \cdot z_{i+k,j+l}, \quad (3.62)$$

mit der Norm

$$\|\mathbf{f} \circ \mathbf{m}_{i,j}\|_1 = \frac{1}{\sum_{k \in v_{K,N}(i)} \sum_{l \in v_{L,M}(j)} f_{k,l} \cdot m_{i+k,j+l}} \quad (3.63)$$

und der Menge valider Indizes

$$v_{K,N}(i) = \{k \in \mathbb{Z} \mid (-K \leq k \leq K) \wedge (1 \leq i+k \leq N)\} \quad (3.64)$$

$$v_{L,M}(j) = \{l \in \mathbb{Z} \mid (-L \leq l \leq L) \wedge (1 \leq j+l \leq M)\}. \quad (3.65)$$

Vereinfacht gesagt wird der Filterkern normiert, sodass die über einem gültigen Tiefenwert liegenden Elemente (mit Maskierung „1“) in Summe 1 ergeben.

**Hoch- und Tiefpassfilter.** Eine spezifische, verbreitete Form von Faltungsfiltren sind Hoch- und Tiefpassfilter. Sie unterdrücken langwellige bzw. kurzwellige Signalanteile in der Tiefen- oder Distanzkarte. Anschaulich lässt sich das für die verschiedenen Formen der Gestaltabweichung beschreiben (siehe Abbildung 2.1): Nimmt man die Distanzkarte als Differenz zwischen Istoberfläche und geometrischer Oberfläche an, so können durch ein geeignetes Tiefpassfilter die Welligkeiten und Rauheiten entfernt werden, sodass lediglich die Formabweichungen betrachtet werden können. Durch ein geeignetes Hochpassfilter können niederfrequente Formabweichungen unterdrückt werden, sodass die Welligkeiten und Oberflächenunvollkommenheiten betrachtet werden können. Zu bemerken ist, dass Gestaltabweichungen üblicherweise auch hochfrequente Anteile beinhalten, sodass die Trennung mittels Hoch- und Tiefpassfilter praktisch nur in sehr einfachen Fällen möglich ist.

Übliche Filterkerne zur Tiefpassfilterung sind das Box-Filter (*Mittelwertfilter*) oder das GAUSS-Filter. In Abbildung 3.32 ist ein Gauß-Filterkern und seine Wirkung auf eine

Oberfläche dargestellt. Statt der Faltung mit einem Hochpass-Filterkern kann auch die Differenz zwischen der Tiefenkarte und ihrer Tiefpassfilterung verwendet werden.

**Rangordnungsfiler.** Einige Arten von Filtern lassen sich nicht durch eine lineare Faltung mit einem Filterkern beschreiben, sondern durch eine nichtlineare Operation. Typisches Beispiel dafür sind Rangordnungsfiler, bei denen die Elemente der Tiefenkarte, die vom Filterkern abgedeckt werden, zunächst nach ihrer Größe sortiert werden. Dem zugehörigen Element in der Ausgangskarte kann dann z. B. der Minimal-, Maximal- oder Medianwert der geordneten Werte zugewiesen werden. Der Medianfilter wird oft zur Filterung einzelner Ausreißerpunkte genutzt.

**Morphologische Operatoren.** Morphologische Operatoren [63] sind im Gegensatz zu faltungsbasierten Filtern nicht signaltheoretisch motiviert, sondern durch die Beeinflussung der geometrischen Form der Oberfläche. Fasst man eine gemessene Oberfläche  $\mathbf{z} \in \mathbb{R}^{M \times N}$  als Tiefenprofil auf, so entspricht die Erosion einer Abtragung hoher Oberflächenbereiche und Randbereiche. Wendet man die Erosion auf die Maskierung  $\mathbf{m} \in \{0, 1\}^{M \times N}$  an, so entspricht dies einer Beschneidung der Randbereiche. Die Ausprägung der Abtragung bzw. die Größe der Beschneidung wird durch ein Strukturelement  $\mathbf{s}_z \in \mathbb{R}^{(2K+1) \times (2L+1)}$  vorgegeben. Die Dilation entspricht einem Auffüllen von Tälern und einer Erweiterung der Randbereiche. Sie ist in Abbildung 3.33 beispielhaft dargestellt. Die Form und Größe des Auffüllens bzw. Erweiterns wird ebenso durch ein Strukturelement  $\mathbf{s}_z$  vorgegeben.

Größenerhaltende Operatoren ergeben sich aus der Verkettung von Erosion und anschließender Dilation zum Opening bzw. Dilation und anschließender Erosion zum Closing. Dadurch können Bereiche voneinander getrennt bzw. miteinander verbunden werden, die sich nur entlang kleiner Kantenbereiche berühren.



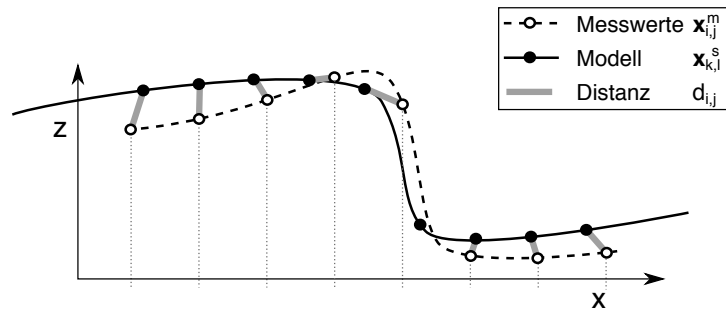


Abbildung 3.29: Abstandsberechnung zwischen Modell (Punktwolke) und 3D-Messung (Punktwolke)

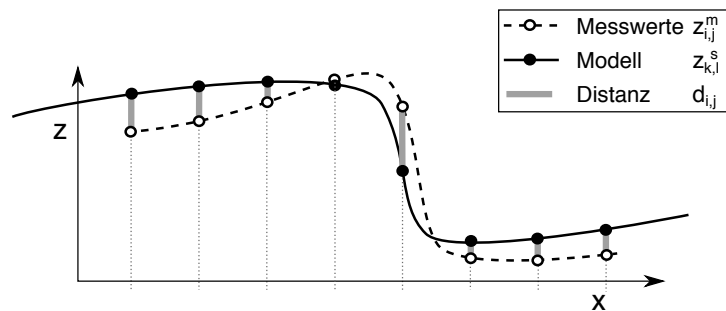


Abbildung 3.30: Abstandsberechnung zwischen Modell (Tiefenkarte) und 3D-Messung (Tiefenkarte) im deckungsgleichen Raster

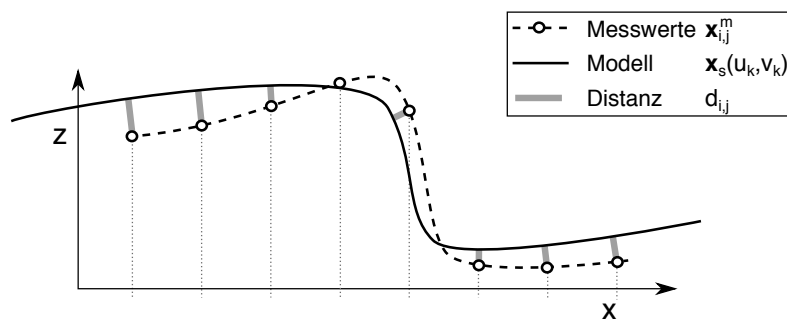


Abbildung 3.31: Abstandsberechnung zwischen Modell (parametrische Fläche) und 3D-Messung (Punktwolke) im deckungsgleichen Raster

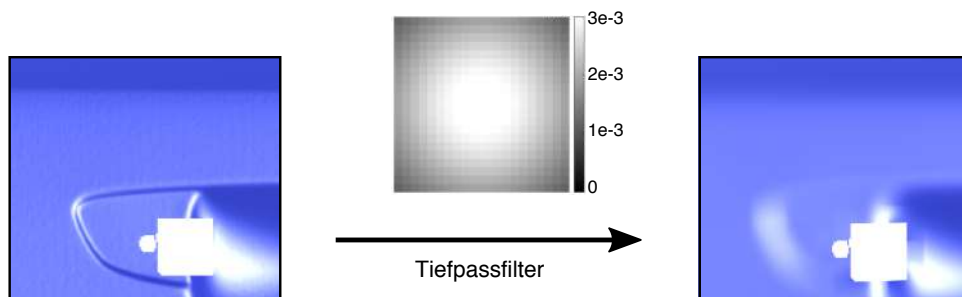
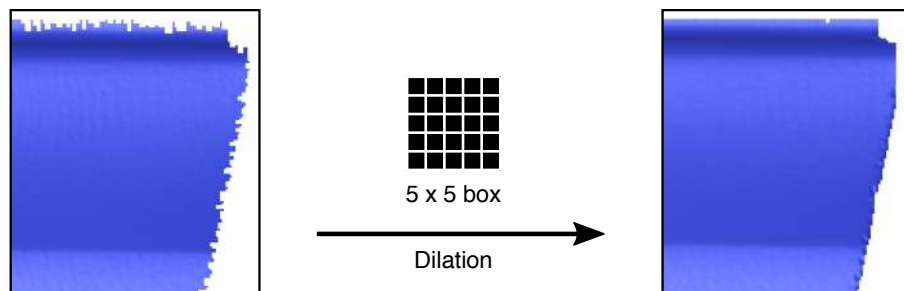


Abbildung 3.32: Ausschnitt der Oberfläche „Türgriffmulde“ (links), Gauß-Filterkern der Größe  $21 \times 21$  (Mitte) und Oberfläche nach Anwendung des Tiefpassfilters (rechts)



**Abbildung 3.33:** Ausschnitt einer Oberfläche mit Randartefakten durch Messung („Flatterränder“, links), Filtermaske für Dilation (Mitte) und Oberfläche nach Anwendung der Dilation (rechts)

---

## Kapitel 4

# Der Assoziativspeicher als Oberflächenmodell

---

In Abschnitt 3.4.1 wurde bereits ein intuitives Verständnis zur Funktionsweise des Assoziativspeichers nach Lilienblum [85] gegeben. In diesem Kapitel wird zunächst ein erweitertes mathematisches Verständnis des Assoziativspeichers (Abschnitt 4.1) und darauf aufbauend ein neuer Ansatz zur effizienten Berechnung der Basisvektoren gegeben (Abschnitt 4.2). Abschließend wird in Abschnitt 4.3 ein neu entwickelter hierarchischer Ansatz dargestellt, durch den die Approximationsgenauigkeit des Assoziativspeichers verbessert wird. Dieser Abschnitt umfasst mathematische Grundlagen zur Multiskalenanalyse, die Beschreibung des entwickelten Verfahrens, sowie die Evaluationsergebnisse anhand von Beispielen.

### 4.1 Mathematische Beschreibung

Die mathematische Beschreibung des Assoziativspeichers wird nun aus verschiedenen Sichtweisen dargestellt: als künstliches neuronales Netz (KNN, 4.1.1), als Hauptkomponententransformation (PCA, 4.1.2) und als Karhunen-Loève-Transformation (KLT, 4.1.3). In den mathematischen Beschreibungen werden folgende Bezeichner konsistent verwendet:

$N$	Dimensionalität der Daten (= Anzahl Punkte)
$N_0$	Anzahl der reduzierten Dimensionen
$M$	Anzahl der Oberflächenmessungen im Datensatz
$\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_M] \in \mathbb{R}^{N \times M}$	Datensatz, Matrix aus Punktwolken
$\mathbf{x}_k \in \mathbb{R}^N$	$k$ -te Punktwolke
$\mathbf{W}_{N_0} = [\mathbf{w}_1 \cdots \mathbf{w}_{N_0}]^T \in \mathbb{R}^{N_0 \times N}$	Transformationsmatrix zur Dimensionsreduktion
$\mathbf{w}_i \in \mathbb{R}^N$	$i$ -ter Transformationsvektor

### 4.1.1 Auto-Assoziative Neuronale Netze

Künstliche neuronale Netze (KNN) abstrahieren die Informationsverarbeitung in natürlichen neuronalen Netzen, wie zum Beispiel dem menschlichen Gehirn. Dadurch ist es möglich, Muster oder Modelle zu lernen, die einer Menge von Trainingsdaten zugrundeliegen. Überwacht lernende KNN sind aus schichtweise miteinander vernetzten Neuronen aufgebaut. Der Ausgangswert eines einzelnen Neurons ergibt sich dabei aus einer gewichteten Summe der Eingangswerte, die anschließend einer geeigneten Aktivierungsfunktion übergeben wird. Die Neuronen sind dabei meist in einer  $N_{in}$ -dimensionalen Eingangsschicht und  $N_{out}$ -dimensionalen Ausgangsschicht strukturiert, die über  $m \geq 0$  verdeckte Schichten beliebiger Dimensionen  $N_i$  verbunden sein können. Ziel des *Trainings* eines KNN ist es, die Gewichtung am Eingang aller Neuronen so zu wählen, dass zwischen vorgegebenen Ein- und Ausgabewerten (Trainingsdaten) ein möglichst geringer Fehler entsteht. Die Topologie des KNN (d.h. Anzahl der Schichten und Neuronen je Schicht) sowie die Transferfunktionen werden vor dem Training geeignet festgelegt. Beim *Abruf (Recall)* eines trainierten KNN werden unbekannte Eingangsdaten übergeben. Die resultierenden Ausgangsdaten folgen dem implizit zugrundeliegenden Modell bzw. Muster, das aus den Trainingsdaten gelernt wurde.

Ein Spezialfall sind *Auto-Assoziative KNN*. Sie besitzen gleiche Ein- und Ausgangsdimensionen ( $N_{in} = N_{out} = N$ ), meist eine verdeckte Schicht mit stark reduzierter Dimension ( $N_0 \ll N_{in}$ ) und Neuronen mit linearer Aktivierungsfunktion. Ziel ist die Abbildung von Daten auf sich selbst. Abbildung 4.1 zeigt die Topologie eines Auto-Assoziativen KNN. Zur Verdeutlichung der Funktionsweise sind die Ausgangswerte  $a_j$  der verdeckten Schicht explizit dargestellt. Daraus ergibt sich für diese Zwischenwerte die mathematische Beschreibung

$$a_i = \sum_{j=1}^N w_{i,j} \cdot x_j \quad (4.1)$$

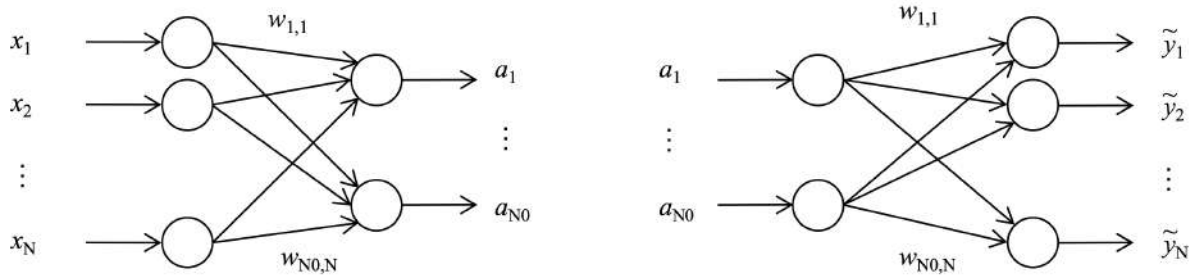
und für die Ausgangswerte

$$\tilde{y}_j = \sum_{i=1}^{N_0} w_{i,j} \cdot a_i. \quad (4.2)$$

Das gesamte Netzwerk lässt sich mit der Gewichtsmatrix  $\mathbf{W}_{N_0} = [w_{i,j}] \in \mathbb{R}^{N_0 \times N}$  zusammenfassen als

$$\tilde{\mathbf{y}} = \mathbf{W}_{N_0}^T \mathbf{W}_{N_0} \cdot \mathbf{x}. \quad (4.3)$$

Die Berechnung der Gewichte erfolgt z. B. über den Backpropagation-Algorithmus [1] aus einer Menge von Trainingsdaten  $\{\mathbf{x}_k \in \mathbb{R}^N | k = 1 \dots M\}$ . Durch die Wahl der Anzahl verdeckter Neuronen  $N_0$  kann das Modell zwischen präziser Approximation bei geringer



**Abbildung 4.1:** Topologie eines Auto-Assoziativen künstlichen neuronalen Netzes für  $N$ -dimensionale Eingangsvektoren

Generalisierbarkeit (viele verdeckte Neuronen  $N_0 \lesssim N$ ) und schlechter Approximation bei hoher Generalisierbarkeit (wenige verdeckte Neuronen  $N_0 \gtrsim 1$ ) eingestellt werden.

Aus den Eigenschaften ergeben sich verschiedene Anwendungsfälle für Auto-Assoziative KNN: Die Dimensionsreduktion erlaubt die *Kodierung* bzw. *Kompression* von Daten. Die Auto-Assoziativität erlaubt die *Fehlerkorrektur* gestörter Eingangsdaten.

### 4.1.2 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (engl. *Principal Component Analysis, PCA*) ist ein statistisches Verfahren zur Bestimmung einer orthogonalen, linearen Transformation, mit der ein Datensatz entlang seiner maximalen Varianzen ausgerichtet wird. Ziel ist die Dimensionsreduktion des Datensatzes auf die wesentlichen varianzbeschreibenden Komponenten. Die resultierenden Hauptachsen stehen daher orthogonal zueinander und indizieren die Richtung maximaler Varianz der Daten in absteigender Reihenfolge (siehe Beispiel Abbildung 4.2). Dabei wird von einem mittelwertfreien Datensatz ausgegangen.

Die Berechnung erfolgt über die Eigenwertzerlegung der Kovarianzmatrix  $\mathbf{X}\mathbf{X}^T$  der Trainingsdaten:

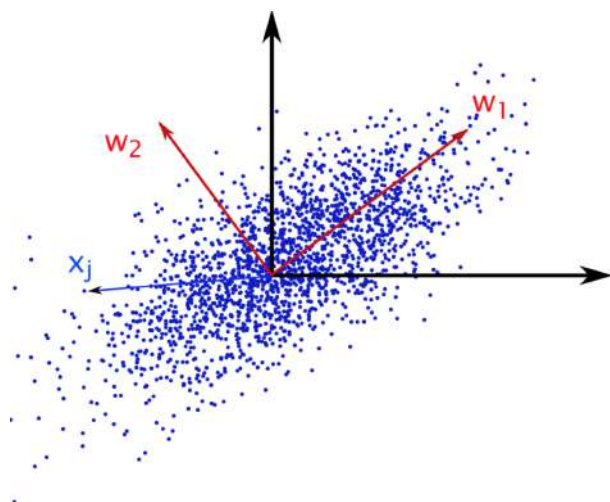
$$\mathbf{X}\mathbf{X}^T = \mathbf{W}^T \cdot \mathbf{\Lambda} \cdot \mathbf{W} \quad (4.4)$$

$$\text{mit } \mathbf{W} = [w_{i,j}] \in \mathbb{R}^{N \times N} \quad \text{Matrix der Eigenvektoren}$$

$$\mathbf{\Lambda} = \text{diag}(\lambda_i) \quad \text{Matrix der Eigenwerte}$$

Oft wird die Berechnung über die Singulärwertzerlegung (SVD) durchgeführt. Darauf wird in Abschnitt 4.2 näher eingegangen.

Die resultierenden Eigenvektoren  $\mathbf{w}_i$  entsprechen den Hauptachsen. Diese zeigen den Anteil der einzelnen Datendimensionen an der Varianz. Die Höhe der zugehörigen Eigenwerte  $\lambda_i$  korreliert mit dem Anteil des Vektors an der Gesamtvarianz. Üblicherweise wer-



**Abbildung 4.2:** Beispiel einer Hauptachsentransformation. Für den zweidimensionalen Datensatz, bestehend aus der Menge  $\{\mathbf{x}_k \in \mathbb{R}^N \quad (N = 2)\}$  (blaue Punkte), indizieren  $\mathbf{w}_1$  und  $\mathbf{w}_2$  (rote Vektoren) die Hauptachsen

den Eigenvektoren und -werte absteigend nach Eigenwerten sortiert. Eigenvektoren bzw. Hauptachsen mit hohen zugehörigen Eigenwerten sind relevanter für die Beschreibung systematischer Varianzen des zugrundeliegenden Datensatzes; Eigenvektoren mit niedrigem zugehörigen Eigenwert bilden eher zufällige Abweichungen einzelner Punktwolken ab.

In der Anwendung werden daher nicht alle  $N$  Hauptkomponenten verwendet, sondern nur die  $N_0$  Hauptkomponenten von höchster Relevanz. Die Anzahl  $N_0$  wird oft empirisch bestimmt; eine mögliche Strategie ist es, das Erreichen eines gewissen Anteils  $\rho_{\text{var}}$  an der Gesamtvarianz zu fordern:

$$N_0 = \arg \min_{N_0} \left| \sum_{j=1}^{N_0} \lambda_j - \rho_{\text{var}} \sum_{i=1}^N \lambda_i \right| \quad (4.5)$$

Die Dimensionsreduktion ergibt sich also durch Projektion der Eingangsdaten  $\mathbf{x}$  auf die reduzierte Eigenvektormatrix  $\mathbf{W}_{N_0}$ :

$$\mathbf{a} = \mathbf{W}_{N_0} \cdot \mathbf{x} \quad (4.6)$$

Vergleicht man dies mit der Beschreibung als auto-assoziatives neuronales Netzwerk, so lässt sich eine Analogie der Kompressionseigenschaft des Netzwerkes zur Dimensionsreduktion der PCA erkennen. Die Ausgabe der verdeckten Schicht (4.1) entspricht also der Projektion auf die Hauptkomponenten (4.6). Entsprechend kann eine Rückprojektion von  $\mathbf{a} \in \mathbb{R}^{N_0}$  in den  $N$ -dimensionalen Raum vorgenommen werden. Im Gegensatz zur linearen Beschreibung der PCA erlaubt ein künstliches neuronales Netz durch geeignete Wahl der

Transferfunktionen sowie zusätzlicher verdeckter Schichten eine nicht-lineare Abbildung und die Verallgemeinerung auf eine nicht-lineare Hauptkomponentenanalyse [78].

### 4.1.3 Karhunen-Loève-Transformation

Die Karhunen-Loève-Transformation ist eine Reihenentwicklung, mit der eine aus einem stochastischen Prozess resultierende Funktion  $x(u)$  approximiert werden kann. Die Approximation lautet dann [92]

$$y(u) = \sum_i a_i \cdot w_i(u) \approx x(u) \quad (4.7)$$

mit den Koeffizienten der Reihenentwicklung  $a_i$  und den Eigenfunktionen  $w_i(u)$ . Die Eigenfunktionen sind dabei angepasste Entwicklungsfunktionen, die sich aus einer Menge von Realisationen  $x_k(u)$  des stochastischen Prozesses ergeben. Liegen die Eigenfunktionen vor, besteht das Approximationsproblem für eine konkret vorliegende Funktion  $x(u)$  aus der Bestimmung der Koeffizienten:

$$a_i = \int_U w_i(u) x(u) du \quad (4.8)$$

Für den Spezialfall diskreter Funktionen bzw. vektorwertiger Daten zerfällt dieses allgemeine Skalarprodukt in ein Vektorskalarprodukt, das bereits aus der Hauptkomponentenanalyse (4.6) bekannt ist.

**Bestimmung der Eigenfunktionen.** Für die Bestimmung der Eigenfunktionen wird angenommen, dass ein Satz von Funktionen  $x_k(u)$  in einer Grundapproximation vorliegt:

$$x_k(u) = \sum_i p_{k,i} \cdot b_i(u). \quad (4.9)$$

Die Funktion wird also durch den Koeffizientenvektor  $\mathbf{p}_k = [p_{k,1} \ p_{k,2} \ \dots]$  charakterisiert. Die  $b_i$  stellen die Basis des Funktionenraumes der  $x_k$  dar. Für diskrete Messwerte entsprechen diese Basisfunktionen Dirac-Impulsen an bestimmten Abtastpositionen, oder anders gesagt den Einheitsvektoren des Eingangsraumes. Genau wie im vektoriellen Fall von orthogonalen Einheitsvektoren ausgegangen wird, kann für die Basisfunktionen Orthonormiertheit gefordert werden:

$$\int_U b_i(u) \cdot b_j(u) du = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (4.10)$$

Die zu ermittelnden Eigenfunktionen können dann auch in diesem Funktionenraum dargestellt werden und werden durch die Eigenvektoren  $\mathbf{e}_r = [e_{r,1} \ e_{r,2} \ \dots]$  charakterisiert:

$$w_r(u) = \sum_i e_{r,i} b_i(u) \quad (4.11)$$

Als Ansatz für die Herleitung dient die Minimierung des quadratischen Approximationsfehlers [92]. Als Lösung folgt das Eigenwertproblem

$$C \mathbf{e}_r = \lambda_r \mathbf{e}_r \quad (4.12)$$

mit der symmetrischen Kovarianzmatrix der Koeffizienten

$$C = \frac{1}{n-1} \sum_{i=1}^n \mathbf{p}_i \mathbf{p}_i^T \quad (4.13)$$

**KLT für Tensorprodukt Darstellungen.** Für den Fall eines 2D-Definitionsbereiches  $x(u, v)$  kann ein Tensorproduktansatz zur Beschreibung verwendet werden [85]:

$$x(u, v) = \sum_{i=1}^n \sum_{j=1}^m p_{i,j} b_i(u) b_j(v), \quad (4.14)$$

mit der Matrix  $\mathbf{P} = [p_{i,j}]$ . Für die Basisfunktionen  $b_i(u)$  und  $b_j(v)$  wird jeweils Orthonormalität gefordert. Die approximierende Reihenentwicklung ist in diesem Fall

$$x(u, v) \approx f(u, v) = \sum_i \sum_j a_{i,j} w_i(u) w_j(v) \quad (4.15)$$

mit den angepassten Entwicklungsfunktionen

$$w_i(u) = \sum_p e_{i,p}^u b_p(u) \quad (4.16)$$

$$w_j(v) = \sum_q e_{i,q}^v b_q(v) \quad (4.17)$$

Die Eigenvektoren für die Entwicklungsfunktionen ergeben sich aus den Eigenwertproblemen [85]

$$\frac{1}{m} \frac{1}{p-1} \sum_{k=1}^M (\mathbf{P}_k \mathbf{P}_k^T) \cdot \mathbf{e}_i^u = \lambda_i^u \mathbf{e}_i^u \quad (4.18)$$

$$\frac{1}{n} \frac{1}{p-1} \sum_{k=1}^M (\mathbf{P}_k^T \mathbf{P}_k) \cdot \mathbf{e}_j^v = \lambda_j^v \mathbf{e}_j^v \quad (4.19)$$



Die Koeffizienten  $a_{i,j}$  der Reihenentwicklung (4.15) berechnen sich aufgrund der Orthogonalität der Basisfunktionen aus

$$a_{i,j} = \int_U \int_V w_i(u)w_j(v)x(u,v)dudv \quad (4.20)$$

$$= (\mathbf{e}_j^v)^T \cdot \mathbf{P}^T \cdot \mathbf{e}_i^u \quad (4.21)$$

$$\mathbf{A} = [a_{i,j}] = \mathbf{E}_u^T \cdot \mathbf{P} \cdot \mathbf{E}_v \quad (4.22)$$

mit

$$\mathbf{E}_u = [\mathbf{e}_1^v \quad \mathbf{e}_2^v \quad \dots] \quad \text{und} \quad (4.23)$$

$$\mathbf{E}_v = [\mathbf{e}_1^u \quad \mathbf{e}_2^u \quad \dots] \quad (4.24)$$

**Vergleich zu diskreten Methoden.** Vergleicht man die Koeffizientenberechnung der Reihenentwicklung (4.7) bzw. (4.20) mit der Dimensionsreduktion der PCA (4.6), so kann die Karhunen-Loève-Entwicklung als Verallgemeinerung der PCA auf einen stochastischen Prozess in einem kontinuierlichen Funktionenraum verstanden werden. Die Trainingsdaten entsprechen den Realisationen dieses stochastischen Prozesses.

Waren bei der PCA die einzelnen Dimensionen der Eingangsdaten voneinander unabhängig, so ergibt sich nun im Gegensatz dazu eine lokale Struktur der Daten aus den Basisfunktionen und deren Parametern  $(u, v)$ . Zur Erklärung sei auf Abbildung 3.28 verwiesen. Hier wurden als Eingangsvektor beispielhaft Werte einer Kurve verwendet. Für diskrete Verfahren ist die Anordnung der Eingangsdimensionen unerheblich. Die einzelnen Werte der dargestellten Kurve könnten in ihrer Reihenfolge beliebig vertauscht werden; die Interpretation als „Kurve“, also eine mögliche lokale Struktur der Daten, ist in dieser Betrachtung nicht relevant und wurde lediglich aus Gründen der Anschaulichkeit gewählt. Daher war hier eine Unterscheidung zwischen 1D- und 2D-Transformation nicht notwendig.

Bei den diskreten Verfahren versteht man unter „Dimensionalität“ die Länge des Eingangsvektors. Dies würde im Falle der KLT der Anzahl der Basisfunktionen entsprechen. Der Begriff „Dimensionalität der Daten“ bezieht sich hier jedoch vielmehr auf die zugrundeliegende Struktur der Daten, man spricht also von einer 1D-Transformation für Kurven und 2D-Transformation für Flächen.

## 4.2 Effizientes Training des Assoziativspeichers

Moderne 3D-Oberflächensensoren können Millionen von 3D-Punkten je Messung generieren. Der Speicher- und Rechenaufwand der Eigenwertzerlegung (EVD) nach Formel (4.4) wird im Wesentlichen von der Größe der Kovarianzmatrix  $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{N \times N}$  bestimmt. Der

Speicherbedarf hängt also quadratisch von der Anzahl  $N$  der 3D-Punkte je Messung ab. Der Rechenaufwand typischer Lösungsverfahren für die Matrixzerlegung skaliert mit  $\mathcal{O}(N^3)$  [35].

Im Folgenden wird eine effizientere Berechnung der EVD für das Training des Assoziativspeichers hergeleitet. Diese basiert auf den Ausführungen in [47].

### 4.2.1 Duale Eigenwertzerlegung

Für typische Anwendungen ist die Anzahl der Messungen  $M$  deutlich geringer als die Anzahl  $N$  der Punkte je Messung ( $M \ll N$ ). Die EVD lässt sich in diesem Fall deutlich effizienter über die Singulärwertzerlegung (SVD) der  $N \times M$  Datenmatrix  $\mathbf{X}$  berechnen [60]:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (4.25)$$

Dabei sind  $\mathbf{U} \in \mathbb{R}^{N \times N}$  und  $\mathbf{V} \in \mathbb{R}^{M \times M}$  quadratische, orthogonale Matrizen. Die Singulärwertmatrix  $\mathbf{\Sigma} \in \mathbb{R}^{N \times M}$  ist dünn besetzt und enthält Singulärwerte nur entlang der Hauptdiagonalen.

Die EVD kann jedoch auch über die duale Eigenwertzerlegung (DEVD) der  $M \times M$  Kovarianzmatrix  $\mathbf{X}^T \mathbf{X}$  berechnet werden. Parallel zu (4.4) ergibt sich dann:

$$\mathbf{X}^T \mathbf{X} = \overline{\mathbf{W}}_n \overline{\mathbf{\Lambda}}_n, \overline{\mathbf{W}}_n^T \quad (4.26)$$

wobei die Matrizenbezeichner der DEVD zur Unterscheidung im Folgenden mit einem Oberstrich markiert werden.

Um die Beziehung zwischen EVD, SVD und DEVD zu verstehen, werden nun die reduzierten SVD-Matrizen  $\mathbf{U}_{N_0} \in \mathbb{R}^{N \times N_0}$  und  $\mathbf{V}_{N_0} \in \mathbb{R}^{M \times N_0}$  eingeführt. Analog zur reduzierten Eigenvektormatrix in (4.6) enthalten sie die ersten  $N_0$  Spalten von  $\mathbf{U}$  bzw.  $\mathbf{V}$ ; die Singulärwertmatrix  $\mathbf{\Sigma}_n \in \mathbb{R}^{N_0 \times N_0}$  enthält ebenso nur die ersten  $N_0$  Eigenwerte entlang der Diagonalen. Wie auch bei der EVD bzw. DEVD werden diese üblicherweise absteigend angeordnet.

Werden Daten auf die reduzierten Matrizen projiziert, ist das mit einer Dimensionsreduktion der Daten gleichbedeutend. Bei geeigneter Wahl von  $N_0$  kann dabei eine hinreichend gute Approximation der Daten  $\mathbf{X}$  gewährleistet werden. Für  $N_0 = \text{rank}(\mathbf{X})$  erhalten wir eine ideale, nicht-approximierte Abbildung. Dies lässt sich formulieren als

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}_{N_0}\mathbf{\Sigma}_{N_0}\mathbf{V}_{N_0}^T \quad \text{für } N_0 = \text{rank}(\mathbf{X}) \quad (4.27)$$

Berechnet man für eine singularwert-zerlegte Matrix nach (4.25) die Kovarianzmatrix, erhält man:

$$\begin{aligned}\mathbf{X}\mathbf{X}^T &= (\mathbf{U}_{N_0}\boldsymbol{\Sigma}_{N_0}\mathbf{V}_{N_0}^T)(\mathbf{U}_{N_0}\boldsymbol{\Sigma}_{N_0}\mathbf{V}_{N_0}^T)^T \\ &= \mathbf{U}_{N_0}\boldsymbol{\Sigma}_{N_0}\underbrace{\mathbf{V}_{N_0}^T \cdot \mathbf{V}_{N_0}}_{=\mathbf{I} \text{ (orthonormal)}}\boldsymbol{\Sigma}_{N_0}^T\mathbf{U}_{N_0}^T\end{aligned}\quad (4.28)$$

Der Vergleich mit der EVD (4.4) ergibt den Zusammenhang zwischen den Matrizen der SVD und EVD:

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}_{N_0}\boldsymbol{\Sigma}_{N_0}\boldsymbol{\Sigma}_{N_0}^T\mathbf{U}_{N_0}^T = \mathbf{W}_{N_0}\boldsymbol{\Lambda}_{N_0}\mathbf{W}_{N_0}^T \quad (4.29)$$

$$\Rightarrow \mathbf{W}_{N_0} = \mathbf{U}_{N_0} \quad \text{und} \quad (4.30)$$

$$\boldsymbol{\Lambda}_{N_0} = \boldsymbol{\Sigma}_{N_0}\boldsymbol{\Sigma}_{N_0}^T \quad (4.31)$$

und nach analoger Herleitung den Zusammenhang zwischen SVD und DEVD:

$$\Rightarrow \overline{\mathbf{W}}_{N_0} = \mathbf{V}_{N_0} \quad \text{und} \quad (4.32)$$

$$\overline{\boldsymbol{\Lambda}}_{N_0} = \boldsymbol{\Sigma}_{N_0}^T\boldsymbol{\Sigma}_{N_0} = \boldsymbol{\Lambda}_{N_0}. \quad (4.33)$$

Um nun die Beziehung zwischen EVD und DEVD zu verstehen, können die hergeleiteten Zusammenhänge (4.30), (4.32) und (4.33) über die Definition der SVD (4.27) verbunden werden:

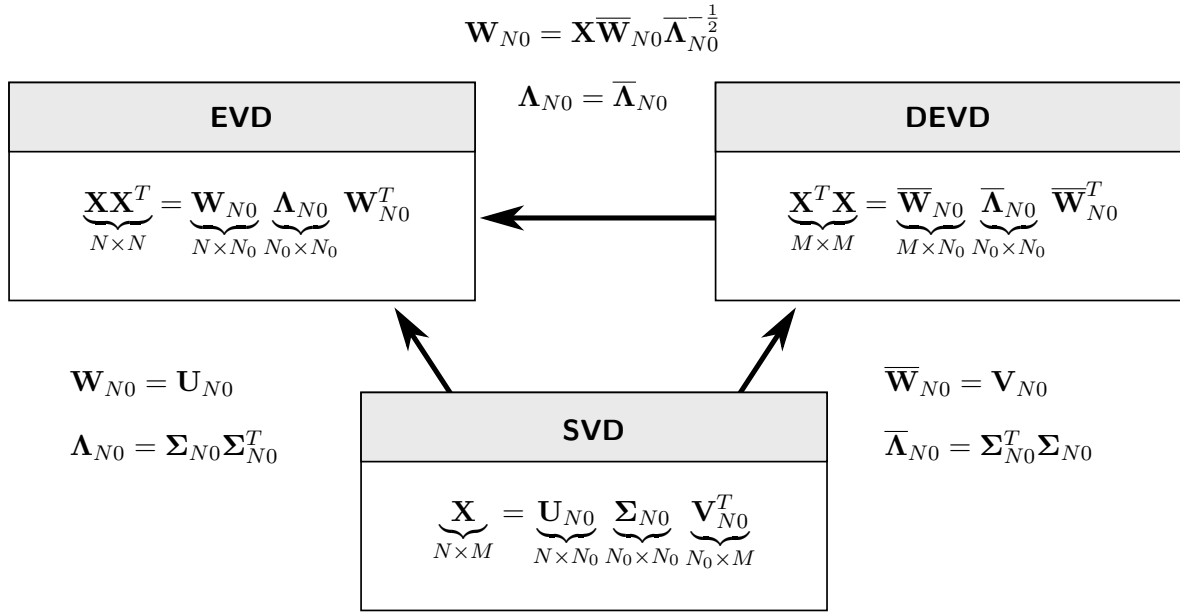
$$\mathbf{X} = \mathbf{U}_{N_0}\boldsymbol{\Sigma}_{N_0}\mathbf{V}_{N_0}^T = \mathbf{W}_{N_0}\boldsymbol{\Lambda}_{N_0}^{-\frac{1}{2}}\overline{\mathbf{W}}_{N_0}^T \quad (4.34)$$

Will man dies nach einer Eigenvektormatrix umstellen, kann man den Zusammenhang  $\mathbf{W}_{N_0}^{-1} = \mathbf{W}_{N_0}^T$  aufgrund der Orthonormalität nutzen:

$$\Rightarrow \mathbf{W}_{N_0} = \mathbf{X}\overline{\mathbf{W}}_{N_0}(\overline{\boldsymbol{\Lambda}}_{N_0})^{-\frac{1}{2}} \quad \text{mit} \quad (4.35)$$

$$\text{abs}(\overline{\boldsymbol{\Lambda}}_{N_0}^{-\frac{1}{2}}) = \text{abs}(\boldsymbol{\Lambda}_{N_0}^{-\frac{1}{2}}) = \text{diag}\left(\frac{1}{\sqrt{\lambda_1}}, \dots, \frac{1}{\sqrt{\lambda_{N_0}}}\right) \quad (4.36)$$

Es sei bemerkt, dass die Eigenvektoren eventuell nicht vorzeichengerecht übereinstimmen. Dies liegt in dem Zusammenhang (4.33) begründet. Eigenwerte lassen sich zwar aus den Singularwerten ableiten. Die Umkehrung davon ist jedoch nicht ohne Informationsverlust möglich. Der Zusammenhang der Eigenwerte der DEVD und EVD gilt entsprechend nur



**Abbildung 4.3:** Zusammenhang zwischen (reduzierter) Eigenwertzerlegung EVD, dualer Eigenwertzerlegung DEVD und Singulärwertzerlegung SVD

hinsichtlich der Absolutbeträge. Für die praktische Anwendung ist dies nicht relevant, da dies über die Vorzeichen der Koeffizienten  $\mathbf{a}$  bei der Projektion (4.6) aufgegriffen wird.

Die hergeleiteten Zusammenhänge sind in Abbildung 4.3 kompakt zusammengefasst. Es zeigt sich, dass die Eigenwertzerlegung aus der dualen Eigenwertzerlegung mittels (4.35) rekonstruiert werden kann; beide Ansätze führen gleichermaßen zur gesuchten reduzierten Eigenwertmatrix  $\mathbf{W}_{N_0}$ . Im Vergleich zur EVD muss bei der DEVD eine deutlich kleinere  $M \times M$ -Matrix zerlegt werden, deren Größe von der Anzahl der Messungen, statt von der Anzahl der 3D-Punkte je Messung, abhängt.

## 4.2.2 Ergebnisse

Die verschiedenen Berechnungsverfahren für das Training des Assoziativspeichers wurden an Beispieldaten getestet, um die Effizienz der Berechnung einschätzen zu können. Dazu wurden zu der Oberflächenmessung eines Karosserieteils „Türgriffmulde“ (vgl. auch Abbildung 1.2) zunächst mehrere 1000 Oberflächen durch Rekombination aus vorhandenen 95 Messungen simuliert, um den Einfluss großer Trainingsdatensätze auf die Rechenzeit besser einschätzen zu können. Die Informationen aus den Trainingsdaten wurden auf  $N_0 = 20$  Eigenvektoren reduziert, für einen Ausschnitt der Fläche von  $N = 10000$  Punkten sowie die vollständige Punktwolke mit ca.  $N \approx 119000$  Punkten.

Die Rechenzeiten wurden für die *Vorbereitung* der Matrizen, die eigentliche *Berechnung* sowie die *Nachbereitung* der Matrizen bestimmt. Die Vorbereitung besteht aus dem Zusammensetzen der Datenmatrizen aus den Trainingsdaten; das ist die reine Datenmatrix

$\mathbf{X}$  bei der SVD sowie die Kovarianzmatrizen bei EVD und DEVD. Die Berechnung entspricht der jeweiligen Matrixzerlegung. Die Nachbereitung ist nur für die DEVD notwendig und meint die Berechnung der Eigenvektoren aus den dualen Eigenvektoren entsprechend Formel (4.35).

Die Rechenzeiten wurden für verschiedene  $N$  und  $M$  und für die drei Berechnungsmethoden auf einem Rechner mit Dual Core Intel Core i5 (2,6 GHz, 8 GB RAM) ermittelt. Die Zeiten sind in Tabelle 4.1 dargestellt und exemplarisch in Abbildung 4.4 visualisiert. Zu bemerken ist, dass die Eigenwertzerlegung für sehr große Punktwolken aufgrund des hohen Speicherbedarfs nicht direkt berechenbar ist. Die DEVD skaliert zwar etwas schlechter für sehr viele Messungen, sie ist jedoch trotz des nötigen Nachbereitungsschrittes deutlich effizienter als die Berechnung mittels SVD.

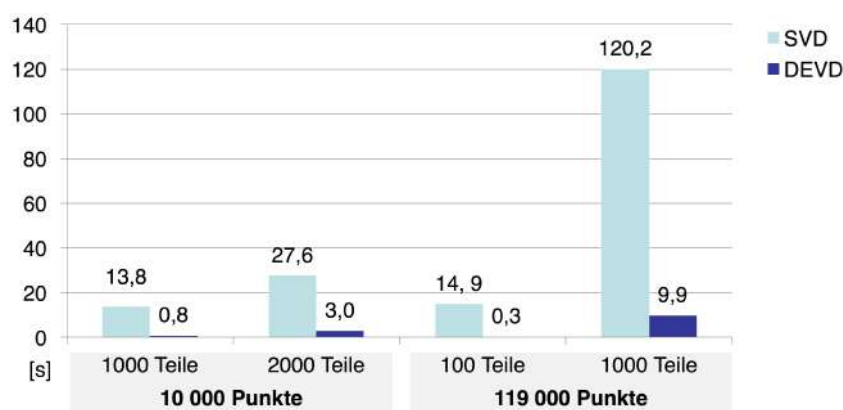
**Tabelle 4.1:** Rechenzeiten für Vorbereitung (Vorb.), Berechnung (Ber.), Nachbereitung (Nachber.) und Summe in [s] für die Methoden EVD, SVD und DEVD bei unterschiedlicher Anzahl an Punkten  $N$ , Anzahl gemessener Teile  $M$  und  $N_0 = 20$

		$M$	10 000			119 000		
		$N$	1000	2000	3000	100	500	1000
EVD	Vorb.		4.318	7.411	10.059	–	–	–
	Ber.		4.294	4.365	3.717	–	–	–
	<b>Summe</b>		8.612	11.775	13.776	–	–	–
SVD	Vorb.		0.039	0.128	0.112	0.037	0.182	0.374
	Ber.		13.801	27.487	39.185	14.837	59.922	119.798
	<b>Summe</b>		13.839	27.615	39.297	14.873	60.104	120.173
DEVD	Vorb.		0.706	2.673	5.623	0.187	2.370	9.048
	Ber.		0.060	0.207	0.397	0.011	0.019	0.065
	Nachber.		0.068	0.131	0.187	0.114	0.395	0.769
	<b>Summe</b>		0.835	3.010	6.207	0.312	2.784	9.882

### 4.3 Hierarchischer Ansatz für den Assoziativspeicher

In Abschnitt 3.4.1 wurde der Assoziativspeicher bereits hinsichtlich der Anforderungen bewertet. Ein grundsätzlicher Zielkonflikt besteht zwischen der Genauigkeit der Darstellung, Modelladaptivität und Robustheit der Oberflächendarstellung. Einerseits sollen Bereiche mit starken lokalen Krümmungen sehr präzise abgebildet werden, andererseits dürfen Messrauschen und kleinste Oberflächenfehler nicht abgebildet werden.

Neben den Trainingsdaten ist die Anzahl der reduzierten Dimensionen  $N_0$  für die geometrischen Eigenschaften des Modells entscheidend. Als systematischer Parameter ermöglicht er die Einstellung zwischen Robustheit und Modelladaptivität bzw. Genauigkeit. Dieser wirkt jedoch global, für alle Punkte auf der Oberfläche gleichermaßen.



**Abbildung 4.4:** Gesamtrechenzeit in [s] für SVD und DEVD-Ansatz für unterschiedliche Anzahl an Punkten  $N$ , Anzahl gemessener Teile  $M$  und  $N_0 = 20$

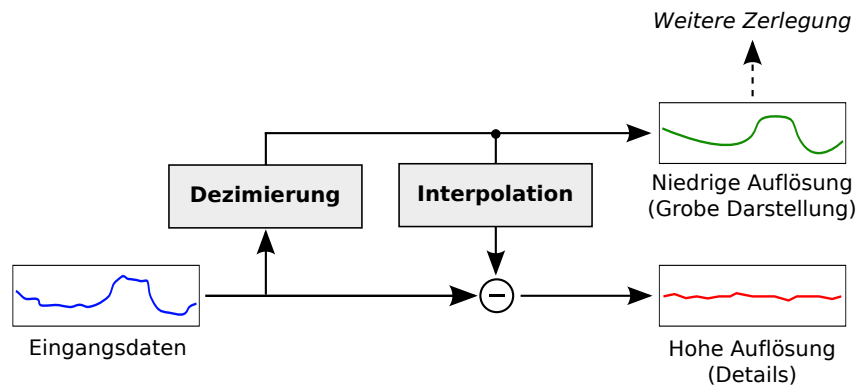
Eine mögliche Verbesserung des Assoziativspeichers besteht also darin, den Parameter  $N_0$  für lokale Bereiche individuell einstellen zu können. So kann er für kritische Bereiche, wie z. B. an starken Krümmungen, höher eingestellt werden, ohne dass die Robustheit des gesamten Modells dadurch beeinflusst wird. Der hier vorgestellte hierarchische Ansatz ermöglicht genau diese lokale Gewichtung und wurde erstmalig in [47] vorgestellt.

### 4.3.1 Multiskalen-Analyse

Multiskalen-Ansätze unterteilen einen Lebesgue-Funktionsraum in verschachtelte Teilräume, wobei die Basisfunktionen auf jeweils kleiner werdenden Trägern definiert sind. So entsteht ein sogenannter *Skalenraum*. Er ermöglicht es, hochfrequente Signalanteile (d.h. Details) in einer hochaufgelösten Skale dazustellen und niederfrequente (grobe) Signalanteile in einer niedrigaufgelösten Skale. Bezogen auf die Basisfunktionen bedeutet das einen kleinen Trägerbereich für hochaufgelöste Skalen und einen großen Trägerbereich für niedrigaufgelöste Skalen. Es ergibt sich ein *Ort-Frequenzraum*.

Ein typisches Beispiel ist die Wavelet-Zerlegung. Der Funktionenraum dieser Zerlegung ergibt sich aus Skalierung und Verschiebung einer Wavelet-Basisfunktion. Das zu untersuchende Signal wird auf diesen Funktionenraum abgebildet. Dabei erlaubt die Skalierungseigenschaft eine Darstellung in Frequenzkomponenten, während durch die Verschiebung gleichzeitig die Ortsinformation erhalten bleibt (*Lokalisierungseigenschaft*). Diese grundlegenden Eigenschaften werden durch zahlreiche weitere Zerlegungsschemen erfüllt. Beispiele sind die Zerlegung in eine Bildpyramide aus der Bildverarbeitung oder die Teilbandkodierung aus der Audiosignalverarbeitung [131].

Abbildung 4.5 zeigt das Konzept einer typischen pyramidalen Zerlegung. Im ersten Schritt, der sogenannten *Dezimierung* werden Eingangsdaten zunächst reduziert. Dazu wird zunächst eine Glättung mittels Tiefpass angewendet und anschließend eine Unterabtastung



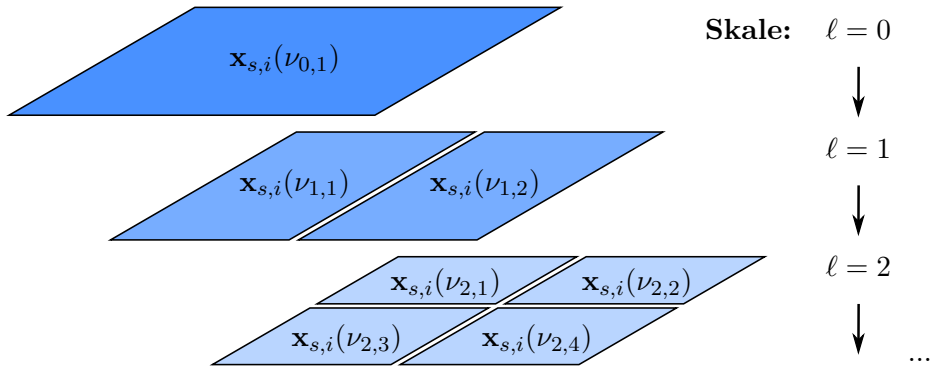
**Abbildung 4.5:** Pyramidale Zerlegung von Eingangsdaten durch iterative *Dezimierung* und *Interpolation* [131]

durchgeführt. Es ergibt sich eine niedrigaufgelöste, gröbere Darstellung der Eingangsdaten. Es schließt sich ein *Interpolationsschritt* an. Dabei wird die reduzierte Darstellung für die Rekonstruktion in der ursprünglichen Skale verwendet, indem sie überabgetastet wird. Der Interpolationsfehler enthält somit nur den hochaufgelösten Informationsanteil für das jeweilige Skalenlevel. Dasselbe Prinzip der Dezimierung und Interpolation wird erneut auf das niedrigaufgelöste Signal angewendet. Dieser Vorgang wiederholt sich abermals, woraus sich letztendlich die Multiskalendarstellung ergibt.

### 4.3.2 Beschreibung des entwickelten Ansatzes

**Multiskalen-Unterteilung für den Assoziativspeicher.** Der Gedanke der Multiskalen-Analyse soll nun auf den Assoziativspeicher übertragen werden, um zu einem hierarchischen Assoziativspeicher zu kommen. Die Basisfunktionen des Assoziativspeichers sind nicht durch Ortsfrequenzen motiviert, sondern durch die statistischen Eigenschaften der Trainingsdaten bzw. Varianzkomponenten (Abschnitt 4.1). Dementsprechend bezieht sich auch die Erweiterung der Basisfunktionen auf mehrere Skalen auf die statistischen Eigenschaften der Trainingsdaten. Anders als bei der Fourier- oder Wavelet-Zerlegung sind die Grundlage für die Zerlegung keine beliebigen ähnlichen Funktionen (wie zum Beispiel Sinusfunktionen oder Haar-Wavelet); die Eigenvektoren sind individuell angepasst an die Kovarianz der Trainingsdaten und somit eine *optimale Basis* hinsichtlich des quadratischen Fehlers [58]. Die Darstellung erfolgt also nicht in einem Orts-Frequenzraum sondern in einem *Orts-Varianzraum*.

Die Lokalisierungseigenschaft wird dadurch erzielt, dass der Trägerbereich der Oberfläche und der Eigenvektoren in Teilbereiche für jedes Skalenlevel unterteilt wird. Diese *Multiskalen-Unterteilung* ist schematisch in Abbildung 4.6 dargestellt. Die Trägerbereiche sind für eine diskrete Punktwolke durch eine Menge von nicht-überlappenden Indizes gegeben, auf denen die Punktwolken  $\mathbf{x}_s$  und Eigenvektoren  $\mathbf{w}$  definiert sind. Dabei stellt



**Abbildung 4.6:** Prinzip der Multiskalen-Unterteilung: Zerlegung der Oberfläche  $\mathbf{x}_{s,i}$  durch Teilung der Träger  $\nu_{\ell,j}$  auf verschiedenen Skalen  $\ell$

$\nu_{\ell,t}$  die Multiskalen-Unterteilung auf dem  $t$ -ten Trägerbereich und der Skale  $\ell$  dar. Für die Unterteilung der Träger gilt:

$$\nu_{\ell,t} \subset \{1, \dots, m\}; \quad \nu_{\ell,j} \cap \nu_{\ell,k} = \emptyset \quad \forall j \neq k \quad (4.37)$$

$$\text{mit} \quad \bigcup_t \nu_{\ell+1,t} \subseteq \bigcup_t \nu_{\ell,t}. \quad (4.38)$$

Dabei gewährleistet die Bedingung (4.38) die hierarchische Unterteilung: die Vereinigung aller Teilbereiche muss in der nächsthöheren Skale mindestens beinhaltet sein. Als Unterteilung können beliebig geformte Bereiche gewählt werden; sie ist nicht auf die in Abbildung 4.6 dargestellten rechteckigen Bereiche beschränkt.

**Training des hierarchischen Assoziativspeichers.** Das Oberflächenmodell wird nun anhand der Trainingsdaten angepasst, wobei eine Verfeinerung, ähnlich wie bei der Bildpyramide aus Abbildung 4.5, in den Teilbereichen kleinerer Skalen vorgenommen wird<sup>5</sup>. Dabei ist intuitiv verständlich, dass es sinnvoll ist, mit einer groben Formbeschreibung zu beginnen und diese auf Formdetails zu verfeinern. Mit jeder Skale nimmt also die Gesamt-Modellkomplexität zu. Daraus wurde der Algorithmus in Tabelle 4.2 entworfen.

Die Größe und Form der Teilbereiche kann sich in jeder Skale ändern und wird zunächst als gegeben angenommen. Mit  $\ell = 0$  wird der vollständige, ursprüngliche Trägerbereich  $\nu_{0,1} = \{1, \dots, N\}$  referenziert, mit der die gesamte Oberfläche beschrieben werden kann. Die Anzahl der Eigenvektoren  $N_{\ell,t}$  je Teilbereich muss ebenfalls vorgegeben werden und ist ein entscheidender Parameter des Algorithmus'. Er muss für den jeweiligen Teilbereich so gewählt werden, dass eine hinreichende Oberflächenapproximation für die wichtigsten Formmerkmale gegeben ist, jedoch feinere Formmerkmale erst mit Basisfunktionen der

<sup>5</sup>Eine ähnliche Idee verfolgt Mackey [90] mit einem *Deflation*-Ansatz für die PCA.



nachfolgenden Skalen abgebildet werden. Im späteren Verlauf wird ein Algorithmus zur automatischen Parameterwahl von  $\nu_{\ell,t}$  und  $N_{\ell,t}$  hergeleitet.

**Abruf des hierarchischen Assoziativspeichers.** Nach der Trainingsphase kommt der hierarchische Assoziativspeicher als Modell zur Oberflächenapproximation zum Einsatz. Dabei wird die zu approximierende Messung entsprechend (4.3) auf die Eigenvektoren einer Skale projiziert. Der verbleibende Modellfehler (*Residuum*) wird wiederum auf die Eigenvektoren der Teilbereiche der nächsten Skale abgebildet usw. Daraus ergibt sich der Algorithmus in Abbildung 4.3.

Ähnlich wie beim Zerlegungsschema der Bildpyramide aus Abbildung 4.5 findet ein Dezimierungs- und Interpolationsschritt bei der Projektion auf die Eigenvektoren und Rückprojektion in den Oberflächenraum statt.

**Algorithmus für die automatische optimale Unterteilung.** Die Parameter des Trainingsverfahrens, die Multiskalen-Unterteilung  $\nu_{\ell,t}$  für die Oberfläche sowie die Anzahl der Eigenvektoren  $N_{\ell,t}$  je Unterteilung, sind entscheidend für die Approximationseigenschaften des hierarchischen Assoziativspeichers. Für die praktische Anwendung ist es sinnvoll, die  $\nu_{\ell,t}$  und  $N_{\ell,t}$  automatisch bestimmen zu können. Dazu wird der Algorithmus aus Tabelle 4.2 nun so erweitert, dass lediglich zwei skalare Parameter manuell vorgegeben werden müssen.

Zunächst soll ein intuitiver Zugang zur Parameterselektion gegeben werden. Es ist sinnvoll, die Oberfläche in Bereiche ähnlicher Krümmung zusammenzufassen. So würden z. B. flache und stark gekrümmte Bereiche voneinander getrennt werden. Beim manuellen Vorgehen kann die Modellqualität nach jeder Parameterwahl empirisch über die Residuen geprüft und angepasst werden. Die Fehlerkarten  $\mathbf{d}$  (z. B. Zeile 6 in Tabelle 4.3) geben ein Indiz für die Lokalisierung von Bereichen mit hohen verbleibenden Fehlern. So können zusammenhängende Bereiche indentifiziert werden.

Wie bereits in Abschnitt 3.4.1 diskutiert, sollte die Anzahl der Eigenvektoren  $N_{\ell,t}$  jeweils so gering wie möglich gehalten werden. Gleichzeitig soll eine maximale Fehlerschwelle nicht überschritten werden. Ob der Fehler nun durch weitere Unterteilung oder durch Erhöhung der Anzahl von Basisvektoren reduziert wird, kann heuristisch anhand verschiedener Faktoren abgeschätzt werden: Höhe des verbleibenden Fehlers, laterale Ausdehnung von Fehlerbereichen, Form und Krümmung der Fehlerbereiche, Absolutzahl von Basisfunktionen oder Gesamtzahl an Skalen. Die manuelle Parameterwahl ist ein entsprechend aufwändiger Vorgang.

In Formel (4.5) wurde eine mögliche Abschätzung für die Anzahl  $N_0$  an Eigenvektoren anhand der Eigenwerte  $\lambda_j$  vorgeschlagen. Diese Abschätzung gilt auch für alle Teilbereiche und deren Eigenwerte, sodass die Parameter  $N_{\ell,t}$  für alle Unterteilungen ermittelt werden können. Dazu ist lediglich die Wahl eines globalen Parameters notwendig, der Anteil an der Gesamtvarianz  $\rho_{var}$ . Dieser ist gleichzeitig ein Maß für die relative Approximationsgenauigkeit der Trainingsdaten.

**Tabelle 4.2:** Algorithmus: Training eines hierarchischen Oberflächenmodells mit gegebener Multiskalen-Unterteilung  $\nu_{\ell,t}$ 

<p><b>Input</b> : <math>M</math> Messungen (Trainingsdaten) <math>\mathbf{x}_{s,i}</math> (<math>i = 1, \dots, M</math>),                  Multiskalen-Unterteilung <math>\nu_{\ell,t}</math> mit  <math>\ell = 0 \dots M</math> Skalen,  <math>t = 1 \dots T_\ell</math> Teilbereiche auf Skale <math>\ell</math>,                  Anzahl der Basisfunktionen <math>N_{\ell,t}</math></p> <p><b>Output:</b> Eigenvektormatrizen <math>\mathbf{W}_{\ell,t}</math></p> <pre> 1 <math>\mathbf{y}_{s,i} \leftarrow \mathbf{x}_{s,i}, \quad \forall i = 1 \dots M</math> 2 <b>for</b> <math>\ell = 0 \dots L</math> <b>do</b> // Iteriere Skalen 3   <b>for</b> <math>t = 1 \dots T_\ell</math> <b>do</b> // Iteriere Teilbereiche 4     <math>\mathbf{W}_{\ell,t} \leftarrow n_{\ell,t}</math> Eigenvektoren von <math>\mathbf{y}_{s,i}(\nu_{\ell,t})</math> // z. B. EVD (4.4) 5     <b>for</b> <math>i = 1 \dots M</math> <b>do</b> // Iteriere Trainingsdaten 6       <math>\tilde{\mathbf{y}}_{s,i}(\nu_{\ell,t}) \leftarrow</math> Projiziere <math>\mathbf{y}_{s,i}(\nu_{\ell,t})</math> auf <math>\mathbf{W}_{\ell,t}</math> // siehe Projektionen (4.3) 7       <math>\mathbf{d}_i(\nu_{\ell,t}) \leftarrow \mathbf{y}_{s,i}(\nu_{\ell,t}) - \tilde{\mathbf{y}}_{s,i}(\nu_{\ell,t})</math> // siehe Abschnitt 3.4.2 8     <b>end</b> 9   <b>end</b> 10  <math>\mathbf{y}_{s,i} \leftarrow \mathbf{d}_i, \quad \forall i = 1 \dots M</math> // Residuen für nächste Skale 11 <b>end</b>                 </pre>
---

**Tabelle 4.3:** Algorithmus: Abruf eines trainierten hierarchischen Oberflächenmodells

<p><b>Input</b> : Gemessene Oberfläche <math>\mathbf{x}_{s,0}</math>,                  Multiskalen-Unterteilung <math>\nu_{\ell,t}</math>,                  Eigenvektormatrizen <math>\mathbf{W}_{\ell,t}</math></p> <p><b>Output:</b> Defektkarte <math>\mathbf{d}_0</math></p> <pre> 1 <math>\mathbf{y}_s \leftarrow \mathbf{x}_{s,0}</math> 2 <b>for</b> <math>\ell = 0 \dots L</math> <b>do</b> // Iteriere Skalen 3   <b>for</b> <math>t = 1 \dots T_\ell</math> <b>do</b> // Iteriere Teilbereiche 4     <math>\tilde{\mathbf{y}}_s(\nu_{\ell,t}) \leftarrow</math> Projiziere <math>\mathbf{y}_s(\nu_{\ell,t})</math> auf <math>\mathbf{W}_{\ell,t}</math> // siehe Projektionen (4.3) 5     <math>\mathbf{d}(\nu_{\ell,t}) \leftarrow \mathbf{y}_s(\nu_{\ell,t}) - \tilde{\mathbf{y}}_s(\nu_{\ell,t})</math> // siehe Abschnitt 3.4.2 6   <b>end</b> 7   <math>\mathbf{y}_s \leftarrow \mathbf{d}</math> 8 <b>end</b>                 </pre>
--

Um die Bereiche zu ermitteln, für die eine weitere Unterteilung notwendig ist, können Defektkarten verwendet werden. Diese ergeben sich aus den Residuenvektoren  $\mathbf{d}_i$  und stellen somit den Approximationsfehler für einen Teilbereich  $t$  bei einer Skale  $\ell$  dar. Unter der Annahme, dass die Trainingsdaten repräsentativ für typische Teilevariationen sind, können dadurch Bereiche gefunden werden, die nicht hinreichend genau approximiert werden und daher weiter unterteilt werden sollten. Um zu einer automatisierten Ermittlung dieser Bereiche zu kommen, wird ein Schwellenwert  $\tau_e$  für den Approximationsfehler definiert. Daraus kann eine Maskierung  $\mathbf{M}$  der Tiefenkarte ermittelt werden (vgl. Abschnitt 3.3.1):

$$\mathbf{M}(j) = \begin{cases} 1 & \bigvee_{i=1}^M \mathbf{d}_i(j) \geq \tau_e \\ 0 & \text{sonst} \end{cases}. \quad (4.39)$$

Aus dieser Maskierung lassen sich nun zusammenhängende Bereiche finden, in denen mindestens eine der Messungen aus dem Trainingsdatensatz die Schwelle  $\tau_e$  für den Approximationsfehler überschreitet. Durch geeignete Anwendung morphologischer Operationen (z. B. Opening, siehe Abschnitt 3.5) können die maskierten Bereiche geglättet und zu kleine Bereiche entfernt werden. Aus der Maskierung können nun die Pixelindizes  $\nu_{\ell+1,t}$  für die Unterteilung der nächsten Skale abgeleitet werden.

Der Gesamtalgorithmus für die automatische optimale Unterteilung ist in Tabelle 4.4 dargestellt. Das Verfahren benötigt neben den Trainingsdaten  $\mathbf{x}_{s,i}$  nun lediglich die zwei skalaren Parameter  $\rho_{var}$  und  $\tau_e$ . Mit  $\rho_{var}$  wird die Anzahl der Eigenvektoren je Skale eingestellt. Kleine Werte führen zu weniger Eigenvektoren, einer schlechteren Approximation bei einer Skale und somit zu einer größeren Skalentiefe. Die Binarisierungsschwelle  $\tau_e$  kann auch als relative Schwelle  $\tau_{rel}$  zum maximalen Approximationsfehler  $\max(\mathbf{d})$  für einen Teilbereich und eine Skale gegeben sein.  $\tau_{rel}$  bestimmt also ebenso die Größe der Teilbereiche. Weiterhin kann eine minimale Größe für Teilbereiche gegeben sein, um eine Unterteilung in sehr kleine Bereiche zu vermeiden. Der Abruf des so trainierten hierarchischen Assoziativspeichers entspricht dem zuvor hergeleiteten Algorithmus aus Tabelle 4.3.

### 4.3.3 Ergebnisse

**Überblick.** Die Algorithmen wurden in MATLAB umgesetzt und anhand mehrerer Messungen von Karosserieteilen getestet. Die Messungen wurden mit einem photogrammetrischen Vier-Kamera-Aufbau und Musterprojektion durchgeführt (kombiniertes Gray-Code- und Phase-Shift-Verfahren, siehe Abschnitt 3.2.2). Die Tiefenauflösung des Sensors liegt bei  $10\ \mu\text{m}$  auf einem regelmäßigen  $1\ \text{mm} \times 1\ \text{mm}$   $x$ - $y$ -Raster im Objektkoordinatensystem. Das Messfeld umfasst eine Oberfläche von  $520\ \text{mm} \times 380\ \text{mm}$ , die Daten wurden nach Messung auf ein Messfeld von etwa  $350\ \text{mm} \times 350\ \text{mm}$  beschnitten. Die Messungen

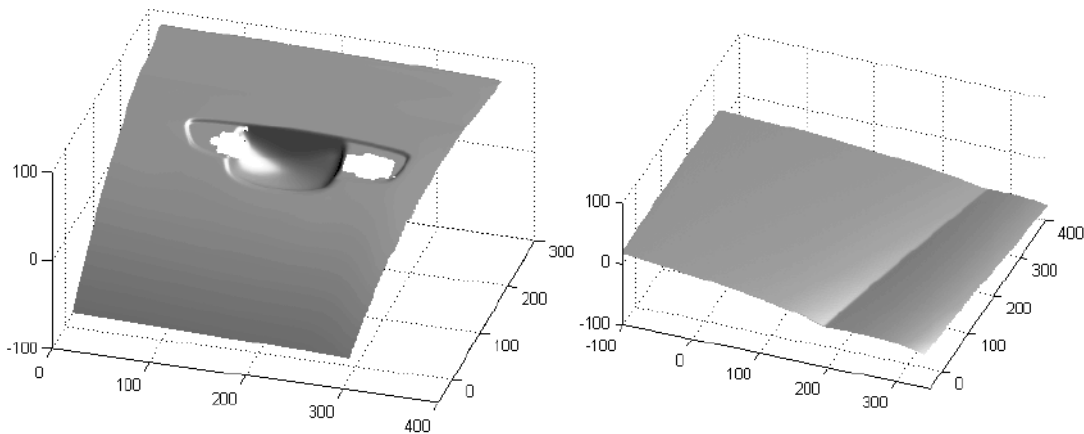
**Tabelle 4.4:** Algorithmus: Training eines hierarchischen Oberflächenmodells mit automatischer, optimaler Wahl der Anzahl an Eigenvektoren  $N_{\ell,t}$  und der Multiskalen-Unterteilung  $\nu_{\ell,t}$ 

```

Input :  $M$  Messungen (Trainingsdaten)  $\mathbf{x}_{s,i}$  ( $i = 1, \dots, M$ ),
          relative Approximationsgenauigkeit  $\rho_{var}$ ,
          relative Binarisierungsschwelle  $\tau_{rel}$ 
Output: Multiskalen-Unterteilung  $\nu_{\ell,t}$ ,
          Eigenvektormatrizen  $\mathbf{W}_{\ell,t}$ 

1  $\nu_{0,1} \leftarrow \{1 \dots N\}$  // Wähle ganze Oberfläche
2  $T_0 \leftarrow 1$  // Ein Bereich für  $\ell = 0$ 
3  $\mathbf{y}_{s,i} \leftarrow \mathbf{x}_{s,i}, \quad \forall i = 1 \dots M$ 
4  $\ell \leftarrow 0$ 
5 while  $\ell < L_{max}$  do // Iteriere Skalen
6   for  $t = 1 \dots T_\ell$  do // Iteriere Teilbereiche
7      $\widetilde{\mathbf{W}}_{\ell,t} \leftarrow$  Eigenvektoren von  $\mathbf{y}_{s,i}(\nu_{\ell,t})$ 
8      $\Lambda_{\ell,t} \leftarrow$  Eigenwerte von  $\mathbf{y}_{s,i}(\nu_{\ell,t})$  // z. B. EVD (4.4)
9      $N_{\ell,t} \leftarrow f(\Lambda_{\ell,t}; \rho_{var})$  // siehe (4.5)
10     $\mathbf{W}_{\ell,t} \leftarrow$  Dimensionsreduktion  $\widetilde{\mathbf{W}}_{\ell,t}$  auf  $N_{\ell,t}$ 
11    for  $i = 1 \dots M$  do // Iteriere Messungen
12       $\tilde{\mathbf{y}}_{s,i}(\nu_{\ell,t}) \leftarrow$  Projiziere  $\mathbf{y}_{s,i}(\nu_{\ell,t})$  auf  $\mathbf{W}_{\ell,t}$ 
13      // siehe Projektionen (4.3)  $\mathbf{d}_i(\nu_{\ell,t}) \leftarrow \mathbf{y}_{s,i}(\nu_{\ell,t}) - \tilde{\mathbf{y}}_{s,i}(\nu_{\ell,t})$ 
14      // siehe Abschnitt 3.4.2
15    end
16     $\tau_e \leftarrow \tau_{rel} \cdot \max(\mathbf{d}_i)$ 
17     $\mathbf{M} \leftarrow$  Maskierung der Bereiche  $\mathbf{d}_i \geq \tau_e$  // siehe Gleichung (4.39)
18     $T_{\ell+1} \leftarrow$  Anzahl zusammenhängender Bereiche in  $\mathbf{M}$ 
19     $\nu_{\ell+1,s} \leftarrow$  Indizes der zusammenhängenden Bereiche aus  $\mathbf{M}$ 
20    // für alle  $s = 1 \dots T_{\ell+1}$ 
21    if  $T_{\ell+1} = 0$  then
22      | exit // Algorithmus beendet
23    end
24  end
25   $\mathbf{y}_{s,i} \leftarrow \mathbf{d}_i \quad \forall i = 1 \dots M$  // Residuen für nächste Skale
26   $\ell \leftarrow \ell + 1$ 

```



**Abbildung 4.7:** 3D-Visualisierungen der untersuchten Oberflächen „Türgriffmulde“ (links) und „Frontklappe“ (rechts)

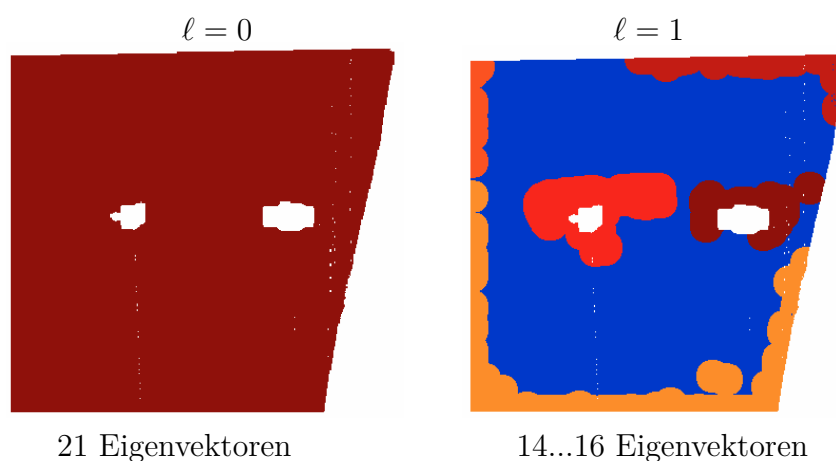
wurden zuvor mit dem Iterative Closest Point (ICP)-Algorithmus auf 1...2 mm genau ausgerichtet. Der verbleibende systematische Fehler durch ungenaue Ausrichtung kann auch durch den Assoziativspeicher abgebildet werden und ist daher nicht kritisch.

Die vermessenen Teile sind in Abbildung 4.7 dargestellt. Für das Teil „Türgriffmulde“ lagen 100 Referenzmessungen fehlerfreier Teile vor. Davon wurden 50 Messungen zufällig für das Training gewählt. Für das Teil „Frontklappe“ lagen 50 Grenzmuster für das Training vor, sowie 21 Messungen mit Oberflächenfehlern. Bei Grenzmustern handelt es sich um Teile, die möglicherweise Fehler enthalten können. Sie wurden durch professionelle Auditoren begutachtet und noch akzeptiert. Das Verfahren wurde auf  $L_{max} = 3$  Skalen beschränkt und mit  $\rho_{var} = 97\%$  für die Approximationsgenauigkeit sowie  $\tau_{rel} = 25\%$  des maximalen Approximationsfehlers für die Binarisierungsschwelle parametrisiert.

Die Rechenzeiten für das Training hängen im Wesentlichen von der Anzahl an Skalen und Teilbereichen sowie deren Größe ab. Auf einem Dual Core Intel Core i5 Desktop-Rechner (2.6 Ghz, 8 GB RAM) benötigte die Berechnung für das Training etwa 3 s („Türgriffmulde“) bzw. 6.5 s („Frontklappe“). Die trainierten Modelle wurden abgespeichert und konnten so für den Abruf verwendet werden; dieser nimmt deutlich unter 1 s in Anspruch (ca. 0.1 s für das Teil Türgriffmulde und  $< 0.3$  s für das Teil Frontklappe).

**Approximationsgüte.** Zunächst soll anhand der Testmessungen der Türgriffmulde evaluiert werden, mit welcher Grundapproximationsgüte zu rechnen ist. Insbesondere sollte bei den Teilen ohne Oberflächendefekte keine falsch-positive Defekterkennung stattfinden, also keine systematische Abweichung durch das Oberflächenmodell eingeführt werden.

In Abbildung 4.8 sind die Teilbereiche und die zugehörige Anzahl an Eigenvektoren dargestellt, die sich aus dem Trainingsalgorithmus (Tabelle 4.4) für das Teil „Türgriffmulde“ ergeben. Als Basisapproximation ergibt sich ein Assoziativspeicher mit  $N_{0,1} = 21$  Basisvektoren ( $\ell = 0$ ). Dies entspricht den Erkenntnissen vorheriger empirischer Untersuchungen. Für etwa 30% der Oberfläche werden weitere  $N_{1,t} = 14 \dots 16$  Basisvektoren



**Abbildung 4.8:** Ergebnis der automatischen Multiskalen-Unterteilung mit Algorithmenparametern  $\tau_{rel} = 0.25$ ,  $\rho_{var} = 0.97$ . Die Farben stellen unterschiedliche Teilbereiche dar; die nicht relevanten Bereiche sind blau dargestellt. **Links:** Skala  $\ell = 0$  (Gesamtfläche), **rechts:** Skala  $\ell = 1$

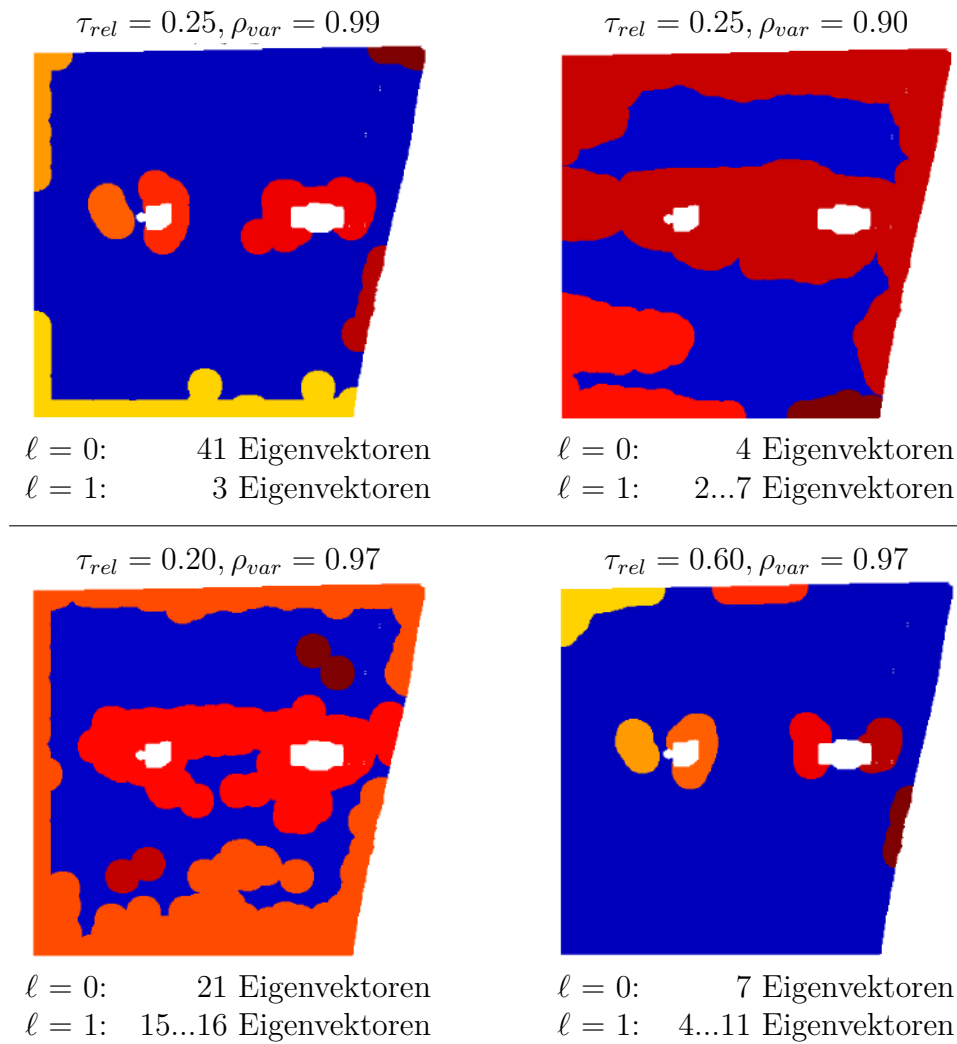
definiert ( $\ell = 1$ ). Zusätzlich werden für weniger als 0.1% der Fläche weitere  $N_{2,t} = 3 \dots 4$  Eigenvektoren durch das Verfahren vorgeschlagen ( $\ell = 2$ ). Diese kommen durch lokale Diskontinuitäten oder Messfehler zustande und werden aufgrund ihrer geringen flächigen Ausdehnung verworfen. Es zeigt sich, dass insbesondere Randbereiche und Bereiche mit hohen Krümmungen an der Trügriffmulde von einer weiteren Multiskalen-Unterteilung profitieren.

Die Tabelle 4.5 gibt einen quantitativen Vergleich der Assoziativspeicher für  $N_0 = 20$ ,  $N_0 = 30$ ,  $N_0 = 40$  sowie des hierarchischen Assoziativspeichers. Es sind die mittleren Spitzenwerte und Standardabweichungen der Defektkarten für die Testmessungen dargestellt. Es zeigt sich, dass auch bei einer sehr hohen Anzahl  $N_0$  der hierarchische Ansatz bessere Ergebnisse zeigt, insbesondere hinsichtlich minimaler und maximaler Abweichung.

**Einfluss der Parameter.** Der in Tabelle 4.4 dargestellte Algorithmus wird durch die relative Approximationsgenauigkeit  $\rho_{var}$  sowie die relative Binarisierungsschwelle  $\tau_{rel}$  parametrisiert. Durch systematische Änderung wurde ihr Einfluss auf das Ergebnis untersucht.

**Tabelle 4.5:** Spitzenwerte und Standardabweichung der Defektkarten für verschiedene Assoziativspeicher-Ansätze. Mittelwert der 50 Testmessungen „Türgriffmulde“ in  $\mu\text{m}$

[ $\mu\text{m}$ ]	Asso.	Asso.	Asso.	hierarch.
	( $N_0 = 20$ )	( $N_0 = 30$ )	( $N_0 = 40$ )	Asso.
Std.abw.	12.9	11.7	11.0	10.6
Min.	-511.6	-517.8	-425.4	-299.7
Max.	341.8	334.1	342.0	281.1
Peak-Peak	853.4	851.9	767.4	580.9



**Abbildung 4.9:** Ergebnisse der Multiskalen-Unterteilung für die jeweils erste Skale ( $\ell = 1$ ) und verschiedene Parameter  $\tau_{rel}, \rho_{var}$ . Die Farben stellen unterschiedliche Teilbereiche dar; die nicht relevanten Bereiche sind blau dargestellt.  $M = 50$  Messungen für das Training

In Abbildung 4.9 sind die Formen der Teilbereiche sowie die resultierenden Anzahlen an Basisvektoren  $N_{\ell,t}$  für verschiedene Parameterwerte dargestellt. In den oberen beiden Abbildungen wurde der Wert  $\rho_{var}$  variiert, in den unteren der Wert  $\tau_{rel}$ . Die Untersuchungen bauen auf der vorherigen, in Abbildung 4.8 dargestellten Untersuchung auf.

Es zeigt sich, dass eine hohe Approximationsgenauigkeit  $\rho_{var}$  (Abbildung 4.9, links oben) zur Überanpassung auf der Ausgangsskala führt. Die Anzahl der Basisvektoren ist hier nah bei der Anzahl der Trainingsdaten ( $M = 50$ ), die Dimensionsreduktion ist sehr gering. Die nächsthöhere Skale trägt kaum zur Gesamtapproximation bei; die Vorteile des hierarchischen Assoziativspeichers werden kaum genutzt. Wählt man den Parameterwert  $\rho_{var}$  kleiner (Abbildung 4.9, rechts oben), entsteht eine tiefere Hierarchie. Jede einzelne Skale beinhaltet weniger Basisvektoren und trägt dadurch weniger zur Gesamtapproximi-

mation bei. Die dargestellten Teilbereiche sind dementsprechend großflächiger, um den Teilbereichen der darunterliegenden Skalen Rechnung zu tragen.

Ein kleiner Wert für den relativen Binarisierungsfehler  $\tau_{rel}$  (Abbildung 4.9, links unten) führt ebenso zu großflächigeren Teilbereichen, allerdings bei einer höheren Anzahl von Eigenvektoren je Skale. Ein hoher Wert  $\tau_{rel}$  führt hingegen zu einer deutlich schlechteren Gesamtapproximation (Abbildung 4.9, rechts unten) und hohen verbleibenden Modellfehlern. Es werden weder die hierarchischen Eigenschaften ausgenutzt (kleine Hierarchie, kleine Teilbereiche), noch wird eine gute Approximation in den einzelnen Skalen erzielt (wenige Eigenvektoren).

Zusammengefasst sollte also ein zu hoher Wert  $\rho_{var}$  vermieden werden; ein kleinerer Wert führt zu einer tieferen Hierarchie, aber akzeptabler Gesamtapproximation. Für  $\tau_{rel}$  sollten ebenfalls hohe Werte vermieden werden, kleine Werte führen zu akzeptablen Ergebnissen bei leichter Überanpassung bzw. zu großen Teilbereichen.

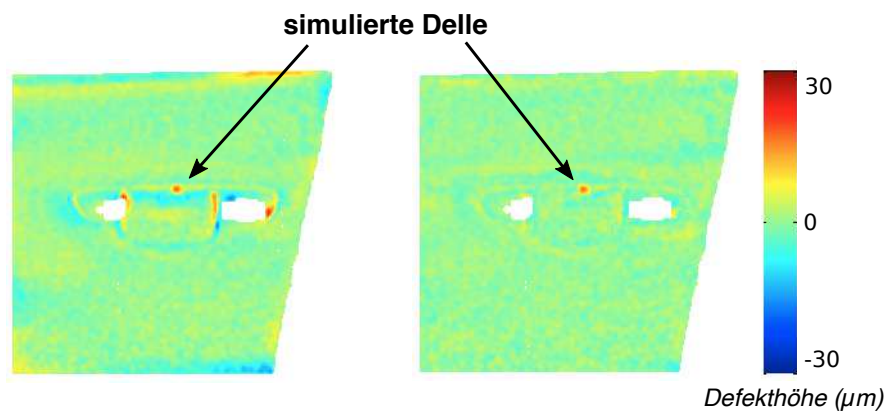
**Defekterkennung.** Um den Einfluss auf die Defekterkennung zu bewerten, wurde eine Delle durch ein gaußförmiges Modell (Höhe: 20  $\mu\text{m}$ , Ausdehnung: 30 mm  $\times$  10 mm) in den 3D-Daten einer Testoberfläche simuliert und untersucht (siehe auch Abbildung 5.7). Abbildung 4.10 zeigt den Vergleich zwischen dem herkömmlichen Assoziativspeicher (links) und dem hierarchischen Ansatz (rechts). Dabei wurden für den hierarchischen Ansatz die Algorithmusparameter  $\rho_{var} = 0.95$  and  $\tau_{rel} = 0.25$  gewählt. Die Ergebnisse sind als Defektkarten  $\mathbf{d}$  in Falschfarbendarstellung dargestellt, wobei die Differenz der  $z$ -Werte für die Berechnung der Defektkarten genutzt wurde.

Die Delle liegt nahe an Bereichen mit Kanten und starken Krümmungen, ihre Höhe ist mit 20  $\mu\text{m}$  nahe an der Auflösungsgrenze des Sensors. Für den herkömmlichen Assoziativspeicher ist die Detektion der Oberflächenfehler nicht ohne Weiteres möglich. In der Defektkarte (Abbildung 4.10, links) sind zahlreiche ähnliche Bereiche erkennbar, die als falsch-positive Fehler segmentiert werden würden. Der hierarchische Assoziativspeicher bietet hier eine deutlich bessere Approximation und ist dennoch sensitiv für die Fehlererkennung.

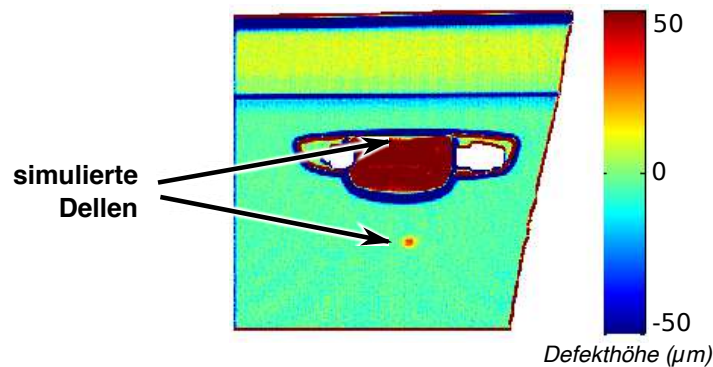
Als Vergleichsexperiment wurde eine konventionelle Analyse im Ortsfrequenzraum durchgeführt. In Abbildung 4.11 ist die Defektkarte für eine Tiefpassfilterung (6 mm  $\times$  6 mm GAUSS-Filterkern) dargestellt. Für leicht gekrümmte Flächen lassen sich Defekte auch mit diesem Ansatz erkennen (untere Delle in Abbildung 4.11); innerhalb komplex geformter Bereiche ist eine Fehlererkennung nicht möglich.

Für das Teil „Frontklappe“ sind Messungen mit realen Oberflächendefekten verfügbar, die zu Testzwecken gezielt auf das Blechteil eingebracht wurden. Der Vorteil des hierarchischen Assoziativspeichers ist hier im direkten Vergleich der Defektkarten in Abbildung 4.12 deutlich sichtbar. Insbesondere entlang der Designkante ist die Grundapproximation beim herkömmlichen Assoziativspeicher (Abbildung 4.12, links) für  $N_0 = 20$  nicht hinreichend. Der Oberflächendefekt geht hier im systematischen Approximationsfehler unter und ist so nicht erkennbar. Der hierarchische Ansatz (Abbildung 4.12, rechts)

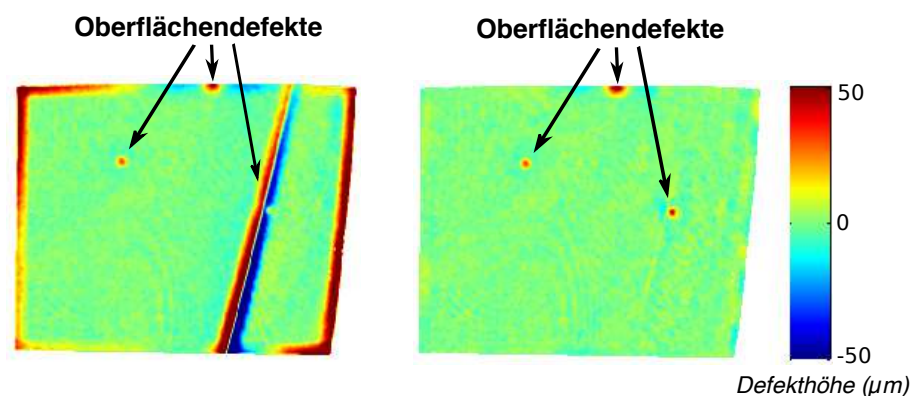




**Abbildung 4.10:** Defektkarte „Türgriffmulde“ mit simulierter Delle (Höhe  $20\ \mu\text{m}$ ), nachbearbeitet mit Hoch- und Tiefpassfilter (GAUSS-Filterkern, Größe  $30\ \text{mm}$  bzw.  $3\ \text{mm}$ ). **Links:** Herkömmlicher Assoziativspeicher mit  $N_0 = 20$  Basisvektoren, **rechts:** hierarchischer Assoziativspeicher ( $\rho_{var} = 0.95, \tau_{rel} = 0.25$ )



**Abbildung 4.11:** Defektkarte „Türgriffmulde“ mit zwei simulierten Dellen (Höhe  $20\ \mu\text{m}$ ), nach Vergleich mit einer Tiefpassfilterung (GAUSS-Filterkern Größe  $6\ \text{mm} \times 6\ \text{mm}$ )



**Abbildung 4.12:** Defektkarte „Frontklappe“ mit echten Oberflächendefekten, nachbearbeitet mit Hoch- und Tiefpassfilter (GAUSS-Filterkern, Größe  $30\ \text{mm}$  bzw.  $3\ \text{mm}$ ). **Links:** Herkömmlicher Assoziativspeicher mit  $N_0 = 20$  Basisvektoren, **rechts:** hierarchischer Assoziativspeicher ( $\rho_{var} = 0.95, \tau_{rel} = 0.25$ )

bietet auch für dieses Teil eine bessere Approximation und ist dennoch sensitiv für die Defektdetektion.

---

## Kapitel 5

# Spline-Methoden für die Oberflächeninspektion

---

In den folgenden Abschnitten werden Verfahren präsentiert, die verschiedene Aspekte der Oberflächeninspektion durch Verwendung einer B-Spline-basierten Oberflächenbeschreibung verbessern. Grundlage für die Verwendung der B-Spline-Oberflächenbeschreibung ist die Approximation von Messwerten (Abschnitt 5.1). In Abschnitt 5.2 wird der Assoziativspeicher für die B-Spline-Basis hergeleitet, sowie Validierungsergebnisse an Beispielen dargestellt. Abschnitt 5.3 stellt ein präzises Verfahren zur Registrierung von Flächen basierend auf der B-Spline-Beschreibung vor. Dies umfasst ebenfalls die Darstellung von Validierungsergebnissen an Beispielen von Karosserieteilen.

## 5.1 B-Spline-Flächen für die Oberflächeninspektion

### 5.1.1 Grundlagen der Flächenapproximation

Ziel der Flächenapproximation ist die als 3D-Punktwolke vorliegende, gemessene Oberfläche als B-Spline-Fläche darzustellen, wobei die Summe der Fehlerquadrate zwischen Messpunkten und Spline-Fläche minimal sein soll.

Dabei ergeben sich einige grundsätzliche Herausforderungen, um eine möglichst gute Approximation mit geringer quadratischer Fehlersumme zu erreichen:

#### 1) Zuordnung von 3D-Punkten zu Parametern der Spline-Fläche

Entscheidend für eine korrekte Repräsentation ist der Erhalt der Topologie. Benachbarte Messpunkte müssen auch im Parameterraum benachbart sein, um eine topologisch korrekte Darstellung zu gewährleisten. Abgesehen davon stellt die Parameterzuordnung einen Freiheitsgrad dar, ist aber eng mit der Wahl der Knotenvektoren verbunden.

## 2) Wahl der Knotenvektoren

Wie in 3.3.4 dargestellt, entscheidet die Wahl des Knotenvektors über den lokalen Träger der Basisfunktionen. Der Knotenvektor sollte dichter gewählt werden, wenn lokal starke, variierende Krümmungen der Oberfläche vorliegen. Mit dem Extremfall äquivalenter Knoten im Parameterraum können Diskontinuitäten der Oberfläche abgebildet werden. Werden Knoten lokal zu dicht gewählt, kommt es zur Überanpassung und zum Schwingen der approximierenden Fläche zwischen den Messpunkten.

## 3) Länge der Knotenvektoren bzw. Anzahl Kontrollpunkte

Die Gesamtzahl der Knoten hängt direkt mit den Überlegungen aus Punkt 2 zusammen. Ziel ist eine effiziente Darstellung der Oberfläche; also bei möglichst geringer Anzahl an Kontrollpunkten den kleinstmöglichen Approximationsfehler zu erreichen.

In der folgenden Herleitung wird vereinfachend angenommen, dass **1)** die Zuordnung der 3D-Punkte in den Parameterraum bekannt ist, **2)** der Knotenvektor bereits geeignet gewählt wurde und somit auch **3)** die Anzahl der Kontrollpunkte bekannt ist. Im Anschluss werden geeignete Strategien für den Allgemeinfall beschrieben.

**Formulierung des Problems für Kurven.** Zunächst soll das 2D-Problem für die Approximation mittels Spline-Kurve  $\mathbf{x}_c(t)$  beschrieben werden. Dazu wird Formel (3.50) für die Kurvendarstellung verwendet:

$$\mathbf{x}_c(u) = \sum_{i=1}^n \mathbf{p}_i b_i^\kappa(u)$$

mit  $\mathbf{p}_i$  Kontrollpunkte  
 $b_i^\kappa(u)$  Basisfunktion  $\kappa$ ten Grades je Kontrollpunkt  
 $\mathbf{u} = (u_1, \dots, u_{n+\kappa})$  den Basisfunktionen zugrundeliegender Knotenvektor

Entsprechend der zuvor beschriebenen Herausforderungen wird hier vereinfacht angenommen, dass zu jedem 2D-Punkt  $\mathbf{x}_k$  der zugehörige Spline-Parameter  $u_k$  bekannt ist sowie die Basisfunktionen  $b_i^\kappa(u)$ , ihr zugrundeliegender Knotenvektor  $\mathbf{u}$  und die daraus resultierende Anzahl Kontrollpunkte  $n$ . Die quadratische Fehlerminimierung zwischen Punkten und Spline-Kurve kann dann formuliert werden als

$$E = \sum_{k=1}^N \|\mathbf{x}_c(u_k) - \mathbf{x}_k\|^2 \rightarrow \min_{\mathbf{p}_i}! \quad (5.1)$$

(siehe dazu auch [38].) Die Lösung des Minimierungsproblems (5.1) ergibt die  $n$  Normalgleichungen:

$$\frac{\partial E}{\partial \mathbf{p}_I} = 0 \quad \forall I = 1 \dots n \quad (5.2)$$

Durch Einsetzen von (3.50) erhält man

$$\begin{aligned} \frac{\partial}{\partial \mathbf{p}_I} \left( \sum_{k=1}^N \left[ \sum_{i=1}^n \mathbf{p}_i b_i^\kappa(u_k) - \mathbf{x}_k \right]^2 \right) &= 0 \\ \sum_{k=1}^N 2 \left[ \sum_{i=1}^n \mathbf{p}_i b_i^\kappa(u_k) - \mathbf{x}_k \right] \frac{\partial \mathbf{p}_i b_i^\kappa(u_k)}{\partial \mathbf{p}_I} &= 0 \end{aligned}$$

mit den Ableitungen

$$\frac{\partial p_i}{\partial p_I} = \begin{cases} 1 & i = I \\ 0 & i \neq I \end{cases} \quad (5.3)$$

Dies lässt sich umformen zu

$$\sum_{i=1}^n \mathbf{p}_i \sum_{k=1}^N b_i^\kappa(u_k) b_I^\kappa(u_k) = \sum_{k=1}^N \mathbf{x}_k b_I^\kappa(u_k) \quad \forall I = 1 \dots n. \quad (5.4)$$

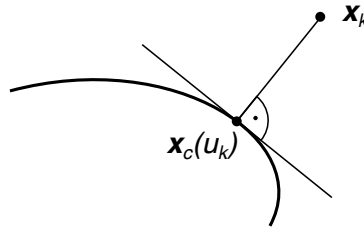
Dies kann auch als lineares Gleichungssystem beschrieben werden:

$$\begin{aligned} \begin{bmatrix} b_1^\kappa(u_1) & \dots & b_1^\kappa(u_N) \\ \vdots & & \vdots \\ b_n^\kappa(u_1) & \dots & b_n^\kappa(u_N) \end{bmatrix} \begin{bmatrix} b_1^\kappa(u_1) & \dots & b_n^\kappa(u_1) \\ \vdots & & \vdots \\ b_1^\kappa(u_N) & \dots & b_n^\kappa(u_N) \end{bmatrix} \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} = \\ \begin{bmatrix} b_1^\kappa(u_1) & \dots & b_1^\kappa(u_N) \\ \vdots & & \vdots \\ b_n^\kappa(u_1) & \dots & b_n^\kappa(u_N) \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} \end{aligned} \quad (5.5)$$

$$\mathbf{B}^T \mathbf{B} \mathbf{P} = \mathbf{B}^T \mathbf{X} \quad (5.6)$$

und kann für die unbekanntenen Kontrollpunkte  $\mathbf{P}$  gelöst werden.

**Parametrisierung.** In (5.1) wurde vorausgesetzt, dass zu den Punkten  $\mathbf{x}_k$  die „zugehörigen“ Parameter  $u_k$  bekannt sind. Zugehörig kann z. B. bedeuten, dass die Differenz



**Abbildung 5.1:** Wahl des Parameters  $u_k$  als Lotfußpunkt von  $\mathbf{x}_k$  auf  $\mathbf{x}_c$

$\|\mathbf{x}_c(u_k) - \mathbf{x}_k\|$  minimal ist, also der entsprechende Punkt auf der Kurve  $\mathbf{x}_c(u_k)$  der Lotfußpunkt von  $\mathbf{x}_k$  auf  $\mathbf{x}_c$  ist (siehe Darstellung in Abbildung 5.1). Die Suche nach geeigneten  $u_k$  erfordert einen erhöhten Rechenaufwand.

Verändert man die Kontrollpunkte  $\mathbf{p}_i$ , so ändert sich die Kurve  $c$ , die Parameter  $u_k$  entsprechen also nicht mehr den geforderten Bedingungen. Beachtet man die Abhängigkeit von Kontrollpunkten und den daraus resultierenden Parameterwerten ergibt sich ein nichtlineares Optimierungsproblem. Alternativ kann auch eine nachträgliche *Parameterkorrektur* der  $u_k$  durchgeführt werden und das veränderte Gleichungssystem (5.6) erneut gelöst werden. Führt man diese Abfolge iterativ durch, stehen die Fehlervektoren  $\mathbf{x}_c(u_k) - \mathbf{x}_k$  senkrecht auf der Kurve.

**Flächenapproximation: Least-Squares-Ansatz.** Wie im Fall von Kurven können auch Flächen über den Ansatz quadratischer Fehlerminimierung (Least-Squares) über alle Punkte approximiert werden. Die Herleitung der Normalgleichungen verläuft analog. Statt  $\mathbf{x}_c$  aus Gleichung (3.50) wird in die Minimierungsbedingung (5.1) aus Gleichung 3.51  $\mathbf{x}_s$  für eine parametrische B-Spline-Fläche eingesetzt. Es wird ebenfalls angenommen, dass zu einem Punkt  $\mathbf{x}_k$  die zugehörigen Parameterwerte  $(u_k, v_k)$  bekannt sind. Es ergibt sich:

$$\sum_{i=1}^n \sum_{j=1}^m \mathbf{p}_{i,j} \sum_{k=1}^N b_i^\kappa(u_k) b_j^\gamma(v_k) b_I^\kappa(u_k) b_J^\gamma(v_k) = \sum_{k=1}^N \mathbf{x}_k b_I^\kappa(u_k) b_J^\gamma(v_k) \quad (5.7)$$

$$\forall I = 1 \dots n, J = 1 \dots m.$$

Die davon abgeleitete Matrixgleichung entspricht der Matrixgleichung für die Kurvenapproximation (5.6), mit dem Unterschied, dass bei den Basismatrizen  $\mathbf{B}$  und dem Punktvektor  $\mathbf{P}$  der 2D-Parameterraum der Fläche berücksichtigt werden muss:

$$\mathbf{B} = \begin{bmatrix} b_1^\kappa(u_1) b_1^\gamma(v_1) & b_1^\kappa(u_1) b_2^\gamma(v_1) & \dots & b_n^\kappa(u_1) b_{m-1}^\gamma(v_1) & b_n^\kappa(u_1) b_m^\gamma(v_1) \\ b_1^\kappa(u_2) b_1^\gamma(v_2) & b_1^\kappa(u_2) b_2^\gamma(v_2) & \dots & b_n^\kappa(u_2) b_{m-1}^\gamma(v_2) & b_n^\kappa(u_2) b_m^\gamma(v_2) \\ \vdots & & & & \vdots \\ b_1^\kappa(u_N) b_1^\gamma(v_N) & b_1^\kappa(u_N) b_2^\gamma(v_N) & \dots & b_n^\kappa(u_N) b_{m-1}^\gamma(v_N) & b_n^\kappa(u_N) b_m^\gamma(v_N) \end{bmatrix} \quad (5.8)$$

$$\mathbf{P} = (p_{1,1} \ p_{1,2} \ \dots \ p_{1,m} \ p_{2,1} \ \dots \ p_{n,m-1} \ p_{n,m})^T \quad (5.9)$$

**Flächenapproximation: Lofting-Ansatz.** Die Größe der Basismatrix  $\mathbf{B}$  beim Ansatz der quadratischen Fehlerminimierung ergibt sich als Produkt der Anzahl gemessener Punkte  $N$  und Anzahl Spline-Kontrollpunkte  $n \cdot m$ . Bei mehreren Millionen Punkten führt dies zu sehr großen Gleichungssystemen. Deutlich effizientere Lösungsansätze sind sogenannte *Skinning*- oder *Lofting*-Methoden [103, 104, 106, 107]. Die Grundidee ist es, eine interpolierende oder approximierende Fläche an eine Reihe von Querschnittskurven anzupassen.

Voraussetzung hierfür ist eine geordnete Punktvolke, mit Messpunkten in einem vollständigen, möglichst gleichmäßigen Raster (z. B.  $z$ -Werte über einem  $x$ - $y$ -Raster). Dies ist eine entscheidende Einschränkung im Vergleich zum Least-Squares-Ansatz, bei dem die Messpunkte beliebig angeordnet sein dürfen. Weiterhin müssen Knotenvektoren und Anzahl von Kontrollpunkten gegeben sein. Zunächst erfolgt eine Approximation von Kurven mit jeweils gleichen Knotenvektoren, entlang einer Richtung im Raster der Punktvolke. Dazu wird der Ansatz quadratischer Fehlerminimierung (5.6) verwendet. Anschließend werden die so entstandenen Kontrollpunkte der Kurven entlang der anderen Richtung approximiert. Die resultierenden Kontrollpunkte werden zur Flächennäherung verwendet.

Das Problem lässt sich formulieren als (vgl. auch [9]):

$$\mathbf{x}_s(u, v_k) = \sum_{i=1}^n \underbrace{\left( \sum_{j=1}^m p_{i,j} b_j^\gamma(v_k) \right)}_{\phi_{i,k}} b_i^\kappa(u), \quad (5.10)$$

wobei die  $\phi_{i,k}$  die Kontrollpunkte für die  $k$ -te Kurvenapproximation entlang der  $u$ -Parameterichtung sind. Die Kontrollpunkte  $p_{i,j}$  für die Fläche ergeben sich, indem eine Kurvenapproximation über die Kontrollpunkte  $\phi_{i,k}$  der zuvor bestimmten Kurven durchgeführt wird. Dabei wird für alle Kurven von der gleichen Parameterzuordnung der Messpunkte ausgegangen (ebenso für die Approximation in Querrichtung). Dies erklärt die Forderung eines möglichst gleichmäßigen Rasters der Messpunkte.

Es lässt sich zeigen[75], dass sich diese aufeinanderfolgenden Approximationsschritte zu einem simultanen Approximationsschritt zusammenfassen lassen können:

$$\mathbf{P} = \left[ (\mathbf{B}_u^T \mathbf{B}_u)^{-1} \mathbf{B}_u^T \right] \cdot \mathbf{X} \cdot \left[ \mathbf{B}_v (\mathbf{B}_v^T \mathbf{B}_v)^{-1} \right] \quad (5.11)$$

mit

$$\text{Kontrollpunktmatrix } \mathbf{P} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,m} \\ \vdots & & \vdots \\ p_{n,1} & \cdots & p_{n,m} \end{bmatrix} \quad (5.12)$$

$$\text{Messpunktmatrix } \mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,M} \\ \vdots & & \vdots \\ x_{N,1} & \cdots & x_{N,M} \end{bmatrix} \quad (5.13)$$

$$\text{Basismatrix in } u \quad \mathbf{B}_u = \begin{bmatrix} b_{u,1}^\kappa(u_1) & \cdots & b_{u,1}^\kappa(u_N) \\ \vdots & & \vdots \\ b_{u,n}^\kappa(u_1) & \cdots & b_{u,n}^\kappa(u_N) \end{bmatrix} \quad (5.14)$$

$$\text{Basismatrix in } v \quad \mathbf{B}_v = \begin{bmatrix} b_{v,1}^\gamma(v_1) & \cdots & b_{v,1}^\gamma(v_M) \\ \vdots & & \vdots \\ b_{v,m}^\gamma(v_1) & \cdots & b_{v,m}^\gamma(v_M) \end{bmatrix} \quad (5.15)$$

Im Vergleich zum Least-Squares-Ansatz zeigt sich, dass die Basismatrix  $\mathbf{B} \in \mathbb{R}^{(N \cdot M) \times (n \cdot m)}$  in zwei deutlich kleinere Basismatrizen  $\mathbf{B}_u \in \mathbb{R}^{N \times n}$  und  $\mathbf{B}_v \in \mathbb{R}^{M \times m}$  zerfällt. Dies macht die Berechnung wesentlich effizienter.

**Wahl des Knotenvektors.** Die Auswirkung des Knotenvektors auf die Oberflächenform wurde in 3.3.4 beschrieben. Im einfachsten Fall kann ein äquidistanter Knotenvektor gewählt werden, dessen Knotenzahl von der Anzahl der Messpunkte abhängt. Oft wird aber eine minimale Approximationsgüte gefordert, d.h. der Approximationsfehler darf eine gewisse Schwelle nicht überschreiten. Dazu muss der Knotenvektor adaptiv an die gemessene Oberfläche angepasst werden.

Es existiert eine Reihe von Verfahren [107], wobei grundsätzlich iterativ vorgegangen wird. Ausgehend von einer initialen Annahme werden die Knotenvektoren inkrementell angepasst; nach jedem Schritt werden die resultierenden Kontrollpunkte bestimmt und der verbleibende Approximationsfehler entlang der Fläche ermittelt. Die Strategien sind durch zwei grundsätzliche Vorgehensweisen motiviert.

1. **Wachsender Knotenvektor:** Die initialen Knotenvektoren sind von minimaler Länge bzw. sehr klein. In jedem Schritt werden solange Knoten auf geeignete Weise hinzugefügt, bis der Approximationsfehler ein gewisses Kriterium unterschreitet. Gebe diese Knotenvektoren zurück.
2. **Knotenreduktion:** Die initialen Knotenvektoren sind von maximaler Länge bzw. sehr groß. In jedem Schritt werden solange Knoten auf geeignete Weise entfernt, bis der Approximationsfehler ein gewisses Kriterium überschreitet. Gebe die letzten Knotenvektoren zurück, die die Approximationsgüte erfüllen.



### 5.1.2 Flächenapproximation für 3D-Messwerte

Bei der Anwendung der Flächenapproximation auf nicht-ideale 3D-Punktwolken müssen weitere Teilprobleme gelöst werden. Gemessene Oberflächen enthalten möglicherweise Löcher oder Randbereiche ohne gültige Tiefenwerte, die geeignet behandelt werden müssen. Weiterhin wurde ein Verfahren zur Bestimmung von Knotenvektoren entworfen, das die Anforderungen der Oberflächeninspektion hinsichtlich Approximationsgenauigkeit erfüllt.

**Auffüllen von Löchern in der Tiefenkarte.** Oberflächen können durch eine beliebige Form begrenzt sein und auch Löcher (z. B. durch Stanzung) enthalten. Die gemessene Punktwolke enthält dann nicht für jeden Punkt des regelmäßigen  $x$ - $y$ -Rasters einen gültigen Tiefenwert. Diese Bereiche können durch Maskierung von der Bearbeitung ausgeschlossen werden (siehe Abschnitt 3.3.1). Einige Verfahren, z. B. der Lofting-Ansatz zur Flächenapproximation, erwarten jedoch eine kontinuierlich definierte Oberfläche. Die betroffenen Bereiche müssen daher mit beliebigen aber sinnvollen<sup>6</sup> Werten aufgefüllt werden.

Der entwickelte Algorithmus führt, motiviert durch den Lofting-Ansatz für die Approximation, eine zeilenweise Extrapolation durch. Dabei wird eine  $C^1$ -Stetigkeit an den Übergängen gefordert (vgl. Abschnitt 3.3.2). Um die Robustheit der Ableitungen an den Randbereichen zu erhöhen, werden sie über mehrere Pixel hinweg ermittelt. Stärkere Anforderungen an die Glattheit sind denkbar, dies erhöht jedoch die notwendige Anzahl der Randpunkte für die robuste Schätzung von Krümmungsparametern.

Die aufgefüllten Bereiche werden mit einem Tiefpassfilter geglättet, um insbesondere Diskontinuitäten quer zu den extrapolierten Zeilen zu reduzieren. Die gemessenen Werte selbst bleiben davon unberührt, d. h. sie werden entsprechend der Maskierung auf den ursprünglichen Werten beibehalten, sodass die Glättung nur auf die gefüllten Bereiche wirkt. Der Gesamtalgorithmus ist in Tabelle 5.1 dargestellt und für das Teil „Türgriffmulde“ in Abbildung 5.2 visualisiert.

**Bestimmung optimaler Knotenvektoren.** Das entwickelte Verfahren zur iterativen Ermittlung optimaler Knotenvektoren ist eine Variante der in [107] beschriebenen Methode. Dabei wurde grundsätzlich der Ansatz des wachsenden Knotenvektors gewählt. Dadurch wird eine minimale Oberflächenbeschreibung, also möglichst wenige Kontrollpunkte, und somit eine geringe Modellkomplexität gewährleistet.

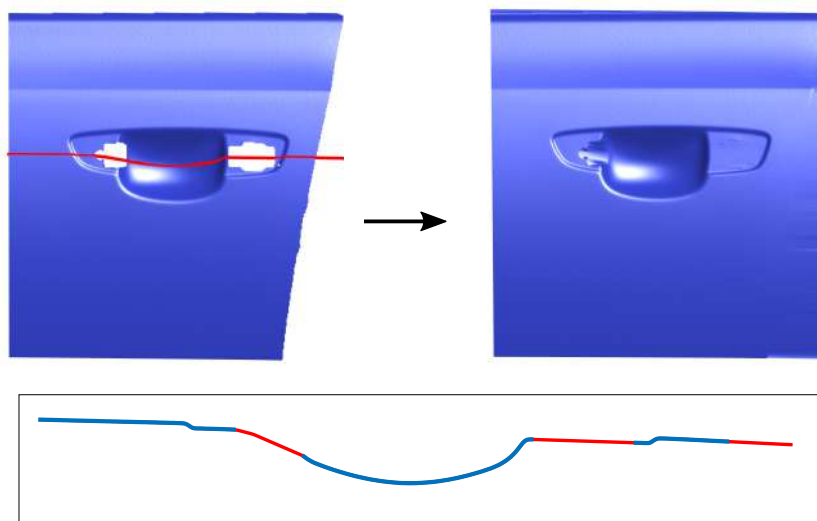
Ein typisches Problem bei diesem Ansatz ist die lokale Überanpassung durch zu dichte Knotenvektoren. Knoten werden vor allem an Stellen starker lokaler Krümmung hinzugefügt, um eine bessere Approximation dieser Gebiete zu gewährleisten. Gleichzeitig kann es bei zu dichten Knoten zur Oszillation kommen<sup>7</sup>, wodurch der Approximationsfehler

<sup>6</sup>„Sinnvolle Füllwerte“ bedeutet zum Beispiel, dass keine Diskontinuitäten oder Sprünge eingebracht werden, die deutlich stärkere Krümmungen im Vergleich zur Umgebung darstellen.

<sup>7</sup>Diese Erscheinung ist in der Approximationstheorie auch als RUNGESches Phänomen bekannt.

**Tabelle 5.1:** Algorithmus: Füllen von Löchern in Tiefenkarten

<p><b>Input</b> : Tiefenkarte mit Löchern <math>\mathbf{z} = [z_{i,j}]</math>,                  Maskierung der Löcher <math>\mathbf{m} = [m_{i,j}]</math>,                  Tiefpass-Filterkern <math>f_{TP}</math></p> <p><b>Output:</b> Aufgefüllte Tiefenkarte <math>\tilde{\mathbf{z}} = [\tilde{z}_{i,j}]</math></p> <pre> 1  <math>\tilde{\mathbf{z}} \leftarrow \mathbf{z}</math> 2  for <math>i \in [1 \dots n]</math> do 3      <math>h \leftarrow 0</math> 4      for <math>j = 1 \dots (m - 1)</math> do 5          if <math>m_{i,j} = 1</math> then 6              if <math>m_{i,j-1} = 0</math> then 7                  if <math>h = 0</math> then 8                      <math>\tilde{z}_{i,1 \dots j} \leftarrow \text{Extrapoliere}(\mathbf{z}_i, 1 \dots j)</math> 9                  else 10                     <math>\tilde{z}_{i,h \dots j} \leftarrow \text{Interpoliere}(\mathbf{z}_i, h \dots j)</math> 11                 end 12             else 13                 <math>h \leftarrow j</math> 14             end 15         end 16     end 17     if <math>m_{i,j} = 0</math> then 18         <math>\tilde{z}_{i,h \dots m} \leftarrow \text{Extrapoliere}(\mathbf{z}_i, h \dots m)</math> 19     end 20 end  21 <math>\tilde{\mathbf{z}} \leftarrow \text{Filter}(\tilde{\mathbf{z}}, f_{TP})</math> 22 for <math>i \in [1 \dots n]</math> and <math>j \in [1 \dots m]</math> do 23     if <math>m_{i,j} = 1</math> then 24         <math>\tilde{z}_{i,j} \leftarrow z_{i,j}</math> 25     end 26 end</pre>	<p><i>// Kopiere Tiefenkarte</i></p> <p><i>// Iteriere über Zeilen</i></p> <p><i>// Startposition des Lochs</i></p> <p><i>// Iteriere entlang der Zeile</i></p> <p><i>// Aktuelle Position: kein Loch</i></p> <p><i>// Loch am Anfang der Zeile</i></p> <p><i>// Loch in der Mitte der Zeile</i></p> <p><i>// Loch am Ende der Zeile</i></p> <p><i>// Glättung der Füllung</i></p> <p><i>// Ursprüngliche Werte behalten</i></p>
--	--



**Abbildung 5.2:** Algorithmus zum Füllen von Löchern in Tiefenkarten. **Links:** Tiefenkarte „Türgriffmulde“, **rechts:** Ergebnis nach Auffüllen, **unten:** Ergebnis des Füllalgorithmus für eine Zeile (blau: Messwerte, rot: aufgefüllte Werte)

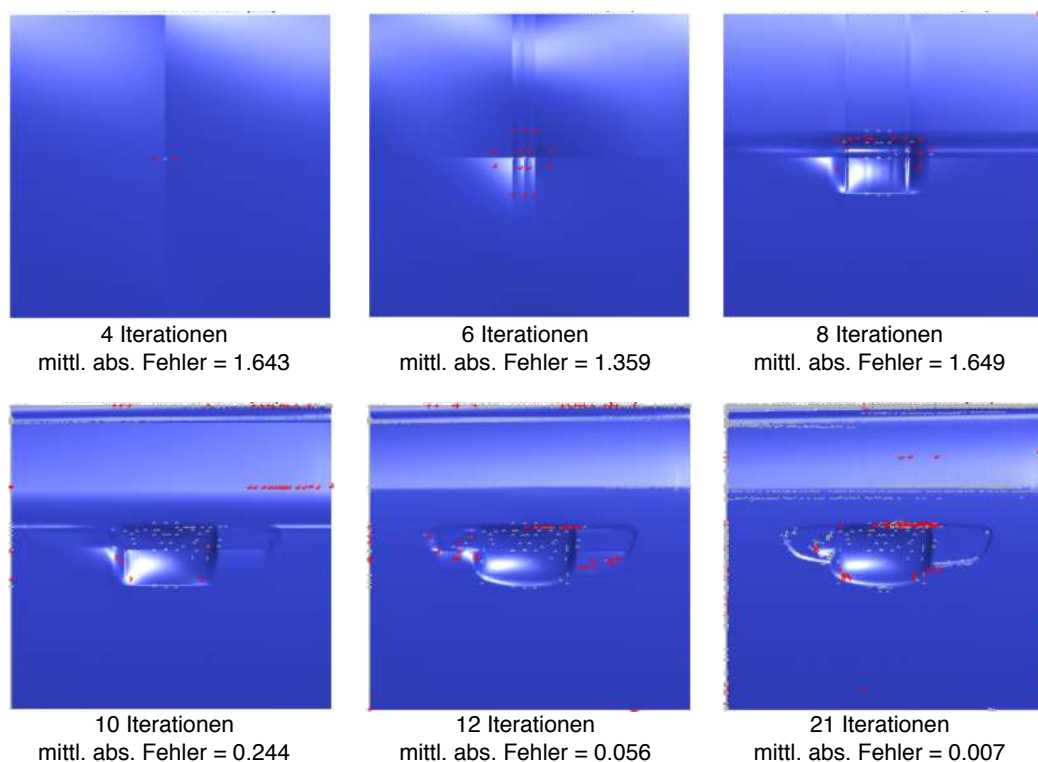
weiter steigt, was wiederum das Hinzufügen von Knoten zur Folge hat. Dies kann zur Divergenz des Verfahrens führen.

Grundgedanke des entwickelten Verfahrens ist es, die Knotenpositionen an einem maximalen Knotenvektor  $\mathbf{u}_{full}$  zu orientieren. Hinzugefügte Knoten können nicht dichter als in diesem voll besetzten Knotenvektor liegen. Das Hinzufügen von Knoten wird in jedem Iterationsschritt durch folgende Kriterien bestimmt:

- Lokale Extrema der Residuenkarte sind Kandidaten für neue Knoten;
- der Fehler an dieser Stelle muss eine absolute Restschwelle  $e_{abs}$  überschreiten;
- der Fehler an dieser Stelle muss eine Schwelle relativ zum globalen Extremum überschreiten ( $\alpha \cdot \max \|\mathbf{d}\|$ ).

Dabei entscheidet der Parameter  $\alpha$  über die Zahl der hinzugefügten Knoten je Iterationsschritt. Für  $\alpha = 1$  wird nur exakt ein Knoten (beim globalen Fehlerextremum) hinzugefügt; für  $\alpha = 0$  werden Knoten an sämtlichen lokalen Extrema hinzugefügt. Dadurch kann der Algorithmus zwischen schneller Konvergenzgeschwindigkeit und minimaler Knotenkonfiguration eingestellt werden. Da die Berechnung nur einmalig (offline) für jedes Teil durchgeführt wird, kann  $\alpha \approx 0.6 \dots 0.9$  hoch gewählt werden. So wird eine Konvergenz des Verfahrens bei minimaler Knotenkonfiguration erzielt.

Der Gesamtalgorithmus ist in Tabelle 5.2 dargestellt. In Abbildung 5.3 sind B-Spline-Flächen für das Teil „Türgriffmulde“ dargestellt, die sich in den Zwischenschritten des Verfahrens ergeben haben. Die je Iterationsschritt hinzugefügten Knoten und zugrundeliegenden Extremstellen lassen sich intuitiv anhand der Krümmungs- und Kantenbereiche



**Abbildung 5.3:** Iterationsschritte des Algorithmus zur Bestimmung optimaler Knotenvektoren. Rot: Extremstellen der Residuenkarte, die im Iterationsschritt zu neuen Knoten geführt haben, grau: bisher hinzugefügte Extremstellen

der Oberfläche erklären. Das Verfahren endet nach 23 Iterationsschritten mit einem maximalen Approximationsfehler von  $e_{abs} = 5 \mu\text{m}$ .

## 5.2 Spline-basierter Assoziativspeicher

Bei der Analyse des Assoziativspeichers in Abschnitt 3.4.1 wurde die Korrektheit des Modellvergleichs als eine Herausforderung identifiziert. Bei der herkömmlichen Formulierung des Assoziativspeichers findet der Vergleich entlang der  $z$ -Achse des übergeordneten Sensorkoordinatensystems statt (Abbildung 3.30). Dies spiegelt den tatsächlichen Oberflächenfehler nur für den Fall exakt wider, wenn die betrachtete Koordinatenrichtung nahezu senkrecht auf der Objektoberfläche steht. Für stark gekrümmte Bauteile kann dies auch bei günstig gewählter Auflage nur eingeschränkt sichergestellt werden. Gleichzeitig ist dadurch auch die Sensitivität eingeschränkt; es besteht eine Abhängigkeit zwischen der Höhe der noch detektierbaren Fehler von den örtlich umgebenden Krümmungsbereichen.

Eine Lösung hierfür ist der Modellvergleich in einem *angepassten, lokalen Oberflächen-*

**Tabelle 5.2:** Algorithmus: Bestimmung optimaler Knotenvektoren

```

Input : Messung, Tiefenkarte  $\mathbf{z} \in \mathbb{R}^{n \times m}$ ,
          Absolute Fehlerschwelle  $e_{abs}$ ,
          Relative Fehlerschwelle  $\alpha$ ,
          Spline-Ordnungen  $\kappa, \gamma$ 
Output: Optimale Knotenvektoren  $\mathbf{u}, \mathbf{v}$ 

1  $\mathbf{u}_{full} \leftarrow$  Äquidistanter Knotenvektor mit  $n$  Elementen
2  $\mathbf{v}_{full} \leftarrow$  Äquidistanter Knotenvektor mit  $m$  Elementen
3  $\mathbf{u} \leftarrow$  Äquidistanter Knotenvektor mit  $\kappa$  Elementen
4  $\mathbf{v} \leftarrow$  Äquidistanter Knotenvektor mit  $\gamma$  Elementen

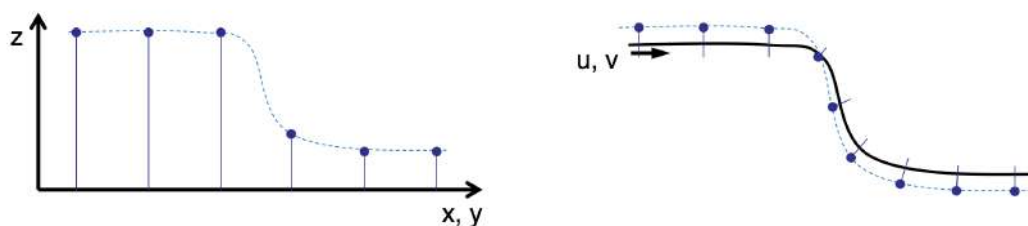
5 for  $I = 1 \dots I_{max}$  do // Maximale Iterationszahl
6    $\mathbf{x}_s(u, v) \leftarrow$  Approximation Spline-Fläche (mit Knoten  $\mathbf{u}, \mathbf{v}$ ) an  $\mathbf{z}$ 
7    $\mathbf{d} \leftarrow$  Modellvergleich( $\mathbf{x}_s, \mathbf{z}$ )
8   if  $\max \|d\| < e_{abs}$  then
9     | exit // Algorithmus beendet
10  end
11   $\{u_i, v_i\}_{max} \leftarrow$  Positionen lokaler Extrema in  $\mathbf{d}$ 
12   $\{u_i, v_i\}_{max} : \text{Entferne } (u_i, v_i) \text{ mit } \|d(u_i, v_i)\| < e_{abs}$ 
13   $\{u_i, v_i\}_{max} : \text{Entferne } (u_i, v_i) \text{ mit } \|d(u_i, v_i)\| < \alpha \cdot \max \|\mathbf{d}\|$ 
14  Ergänze  $\mathbf{u}$  um nächste Nachbarn von  $\mathbf{u}_{full}$  zu  $\{u_i\}_{max}$ 
15  Ergänze  $\mathbf{v}$  um nächste Nachbarn von  $\mathbf{v}_{full}$  zu  $\{v_i\}_{max}$ 
16 end

```

*koordinatensystem.* Die Distanzmessung findet dann entlang des begleitenden Dreibeins statt (*Frenet-* bzw. *Darboux-*Frame, vgl. Abbildung 3.17), dies entspricht dem in Abschnitt 3.4.2 vorgestellten Punkt-Fläche-Abstand, orthogonal zur Oberfläche. Die beiden Varianten sind in Abbildung 5.4 im Vergleich dargestellt. Um die Messung in einem lokalen Koordinatensystem zu ermöglichen, muss die Oberflächenbeschreibung angepasst werden.

In diesem Abschnitt wird daher der Assoziativspeicher für eine B-Spline-Oberflächenbeschreibung hergeleitet. Die Darstellungen beruhen im Wesentlichen auf den Arbeiten in [45, 52]. Die B-Spline-Beschreibung bietet neben einem lokalen Oberflächenkoordinatensystem weitere Vorteile:

- Kompakte Flächendarstellung mit wenigen Parametern (Kontrollpunkten),



**Abbildung 5.4:** Messung in einem globalen Sensorkoordinatensystem entlang der  $z$ -Achse (links) und in einem angepassten lokalen Koordinatensystem entlang der Oberflächennormale (rechts)

- Lokalisierung des Parametereinflusses durch kompakte lokale Träger,
- Möglichkeit zur lokalen Gewichtung durch Verdichtung der Knotenvektoren,
- Kontinuierliche Oberflächenbeschreibung,
- Einfache, effiziente Auswertung und viele verfügbare Algorithmen.

### 5.2.1 Orthogonale B-Spline-Basis

Für die Herleitung des Assoziativspeichers mit B-Spline-Basis wird auf die Formulierung des Problems als Karhunen-Loève-Transformation zurückgegriffen (Abschnitt 4.1.3). Diese ermöglicht es, die Vorteile der kontinuierlichen Oberflächenbeschreibung zu nutzen. Da die Beschreibung jedoch auf orthogonalen Basisfunktionen aufbaut, müssen die B-Spline-Basisfunktionen zunächst in orthogonale Form gebracht werden. Hierzu kann das GRAM-SCHMIDT-Verfahren [65] verwendet werden. Dabei müssen jedoch Integralgleichungen gelöst werden, was mit einem hohen numerischen Aufwand verbunden ist.

Alternativ dazu wurde in [113] eine Orthogonalisierung ausgehend auf der Matrix-Darstellung von Splines (3.54) hergeleitet. Die benötigten Integrale (allgemeines Skalarprodukt) zerfallen dabei in rein lineare Matrixoperationen. Hierzu wird die sog. GRAM-Matrix  $\Sigma_G$  benötigt:

$$\begin{aligned} \Sigma_G &= \begin{bmatrix} \int b_1^\kappa(u)b_1^\kappa(u)du & \int b_1^\kappa(u)b_2^\kappa(u)du & \dots & \int b_1^\kappa(u)b_n^\kappa(u)du \\ \int b_1^\kappa(u)b_2^\kappa(u)du & \int b_2^\kappa(u)b_2^\kappa(u)du & & \vdots \\ \vdots & & \ddots & \\ \int b_1^\kappa(u)b_n^\kappa(u)du & \dots & & \int b_n^\kappa(u)b_n^\kappa(u)du \end{bmatrix} \\ &= \sum_i \mathbf{M}^\kappa(i)^T \cdot \Delta_\kappa \cdot \mathbf{M}^\kappa(i) \end{aligned} \quad (5.16)$$

mit der Hilbert-Matrix  $\Delta_\kappa$  der Größe  $\kappa \times \kappa$

$$\Delta_\kappa = \begin{bmatrix} 1 & \frac{1}{2} & \dots & \frac{1}{\kappa} \\ \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{\kappa+1} \\ \vdots & \vdots & \dots & \vdots \\ \frac{1}{\kappa} & \frac{1}{\kappa+1} & \dots & \frac{1}{2\kappa-1} \end{bmatrix} \quad (5.17)$$

Für eine beliebige Zerlegung, die die Bedingung

$$\mathbf{\Lambda}^T \cdot \mathbf{\Lambda} = \Sigma_G^{-1} \quad (5.18)$$

erfüllt, ist  $\mathbf{\Lambda}$  die Orthogonal-Transformationsmatrix. Somit können die B-Spline-Basisfunktionen orthogonalisiert werden (hier zunächst für Kurven):

$$\mathbf{x}_c(u) = \begin{bmatrix} 1 & \vartheta & \vartheta^2 & \dots & \vartheta^{\kappa-1} \end{bmatrix} \cdot \mathbf{M}^\kappa(i) \cdot \mathbf{\Lambda}^T \begin{bmatrix} p_{i-\kappa+1}^\perp \\ p_{i-\kappa+2}^\perp \\ \vdots \\ p_i^\perp \end{bmatrix} \quad (5.19)$$

Dabei lässt sich der nun benötigte Kontrollpunktvektor  $\mathbf{p}^\perp$  für orthogonalisierte Basen mittels folgender Transformation aus den üblichen Kontrollpunkten berechnen:

$$\mathbf{p}^\perp = (\mathbf{\Lambda}^T)^{-1} \cdot \mathbf{p} \quad (5.20)$$

Die resultierende Darstellung in orthogonaler Basis ist äquivalent zur B-Spline-Basis. Die GRAM-Matrix  $\Sigma_G$  und die orthogonale Transformationsmatrix  $\mathbf{\Lambda}$  ist lediglich vom Knotenvektor der B-Spline-Basis  $b_i^\kappa(u)$  abhängig. Da ähnliche Flächen die gleichen Knotenvektoren nutzen, können die Matrizen  $\Sigma_G$  und  $\mathbf{\Lambda}$  vorberechnet werden. Die Orthogonalisierung lässt sich nun als einfache lineare Transformation darstellen.

### 5.2.2 Herleitung der KLT für B-Spline-Basen

In Abschnitt 4.1.3 wurde der Assoziativspeicher als Karhunen-Loève-Transformation beschrieben. Zusammengefasst sind die folgenden Schritte notwendig:

1. **Training:** Lösen der Eigenwertprobleme (4.18) und (4.19),
2. **Hintransformation:** Für eine zu untersuchende Messung die Koeffizienten zur Reihenentwicklung bestimmen (4.22),
3. **Rücktransformation:** Approximation der Messung durch die Reihenentwicklung (4.15).

Zunächst muss hierfür die zuvor beschriebene Orthogonalisierung der B-Spline-Basis durchgeführt werden. Aus (5.19) lässt sich die Darstellung in orthogonaler B-Spline-Basis für Tensorproduktflächen in der Matrixschreibweise ableiten:

$$\mathbf{x}_s(u, v) = \left( \vartheta_u \cdot \mathbf{M}_u^\kappa(i) \cdot \mathbf{\Lambda}_u^T \right) \cdot \mathbf{P}_\perp \cdot \left( \mathbf{\Lambda}_v [\vartheta_v \cdot \mathbf{M}_v^\gamma(j)]^T \right) \quad (5.21)$$

$\mathbf{\Lambda}_u$  und  $\mathbf{\Lambda}_v$  sind dabei die Orthogonaltransformationsmatrizen für die Basen in  $u$ - bzw.  $v$ -Richtung. Die Transformation der Kontrollpunktmatrix erfolgt analog zu 5.20 mit:

$$\mathbf{P}_\perp = \left( \mathbf{\Lambda}_u^T \right)^{-1} \cdot \mathbf{P} \cdot \mathbf{\Lambda}_v^{-1} \quad (5.22)$$

Mit dieser orthogonalisierten Kontrollpunktmatrix  $\mathbf{P}_\perp$  können die Eigenwertprobleme (4.18) und (4.19) umgeschrieben werden:

$$\frac{1}{m} \frac{1}{p-1} \sum_{k=1}^p \left( \mathbf{\Lambda}_u^T \right)^{-1} \mathbf{P}_k \mathbf{\Lambda}_v^{-1} \left( \left( \mathbf{\Lambda}_u^T \right)^{-1} \mathbf{P}_k \mathbf{\Lambda}_v^{-1} \right)^T \mathbf{e}_i^u = \lambda_i^u \mathbf{e}_i^u \quad (5.23)$$

$$\vdots$$

$$\frac{1}{m} \frac{1}{p-1} \left( \mathbf{\Lambda}_u^T \right)^{-1} \left[ \sum_{k=1}^p \mathbf{P}_k \Sigma_G^v \mathbf{P}_k^T \right] \mathbf{\Lambda}_u^{-1} \mathbf{e}_i^u = \lambda_i^u \mathbf{e}_i^u \quad (5.24)$$

$$\frac{1}{n} \frac{1}{p-1} \left( \mathbf{\Lambda}_v^T \right)^{-1} \left[ \sum_{k=1}^p \mathbf{P}_k^T \Sigma_G^u \mathbf{P}_k \right] \mathbf{\Lambda}_v^{-1} \mathbf{e}_j^v = \lambda_j^v \mathbf{e}_j^v \quad (5.25)$$

Mit den so ermittelten Eigenvektormatrizen kann die eigentliche Karhunen-Loève-Transformation aus Formel (4.22) durchgeführt werden. Dabei werden wieder die orthogonalisierten Kontrollpunkten  $\mathbf{P}_\perp$  genutzt:

$$\mathbf{A} = \mathbf{E}_u^T \left( \mathbf{\Lambda}_u^T \right)^{-1} \cdot \mathbf{P} \cdot \mathbf{\Lambda}_v^{-1} \mathbf{E}_v \quad (5.26)$$

Anhand der berechneten KLT-Koeffizienten findet die Filterung, also die Dimensionsreduktion statt. Anschließend kann die Rücktransformation (4.15) angewendet werden. Dabei ist zu beachten, dass die Eigenfunktionen auf orthogonalisierten B-Spline-Basen beruhen (vgl. mit Formel (5.19)). In Matrixschreibweise ergibt sich:

$$x_s(u, v) \approx f(u, v) = \sum_i \sum_j a_{i,j} w_i(u) w_j(v) \quad (5.27)$$

$$= \sum_i \sum_j a_{i,j} \left( \vartheta_u \mathbf{M}_u^\kappa(i) \mathbf{\Lambda}_u^T \mathbf{e}_i^u \right) \left( \vartheta_v \mathbf{M}_v^\gamma(j) \mathbf{\Lambda}_v^T \mathbf{e}_j^v \right) \quad (5.28)$$

$$= \left( \vartheta_u \mathbf{M}_u^\kappa(i) \mathbf{\Lambda}_u^T \mathbf{E}_u \right) \cdot \mathbf{A} \cdot \left( \vartheta_v \mathbf{M}_v^\gamma(j) \mathbf{\Lambda}_v^T \mathbf{E}_v \right)^T \quad (5.29)$$



Setzt man die Koeffizienten (5.26) ein, so ergibt sich für die gesamte Reihenentwicklung in Matrixschreibweise:

$$f(u, v) = \left[ \vartheta_u \mathbf{M}_u^\kappa(i) \boldsymbol{\Lambda}_u^T \mathbf{E}_u \mathbf{E}_u^T (\boldsymbol{\Lambda}_u^{-1})^T \right] \cdot \mathbf{P} \cdot \left[ \boldsymbol{\Lambda}_v^{-1} \mathbf{E}_v \mathbf{E}_v^T \boldsymbol{\Lambda}_v (\mathbf{M}_v^\gamma(j))^T \vartheta_v^T \right] \quad (5.30)$$

Gleichung (5.30) beschreibt die Gesamttransformation. Sie kann aufgefasst werden als eine Verkettung von Teiltransformationen der Kontrollpunkte  $\mathbf{P}$ . Zusammengefasst werden die folgenden Transformationen nacheinander von innen nach außen durchgeführt (siehe auch Abbildung 5.5):

1. Transformation in den orthogonalen B-Spline-Raum ( $\boldsymbol{\Lambda}_u^{-1}, \boldsymbol{\Lambda}_v^{-1}$ )
2. Transformation in den Karhunen-Loève-Eigenraum ( $\mathbf{E}_u^T, \mathbf{E}_v^T$ )
3. Rücktransformation aus dem Karhunen-Loève-Eigenraum ( $\mathbf{E}_u, \mathbf{E}_v$ )
4. Rücktransformation aus dem orthogonalen B-Spline Raum ( $\boldsymbol{\Lambda}_u, \boldsymbol{\Lambda}_v$ )
5. Reihenentwicklung auf den Basisfunktionen ( $\vartheta_u \mathbf{M}_u^\kappa(i), \vartheta_v \mathbf{M}_v^\gamma(j)$ )

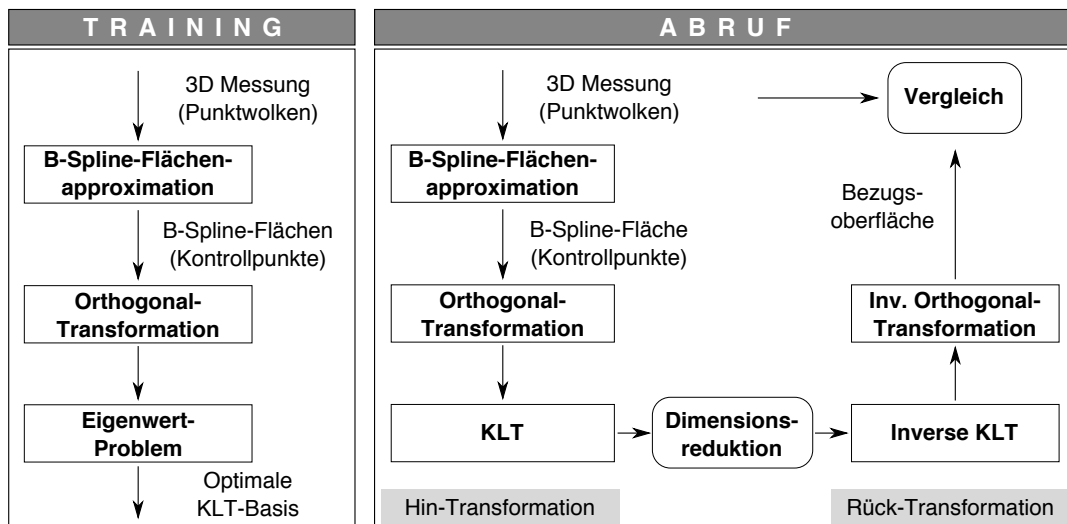
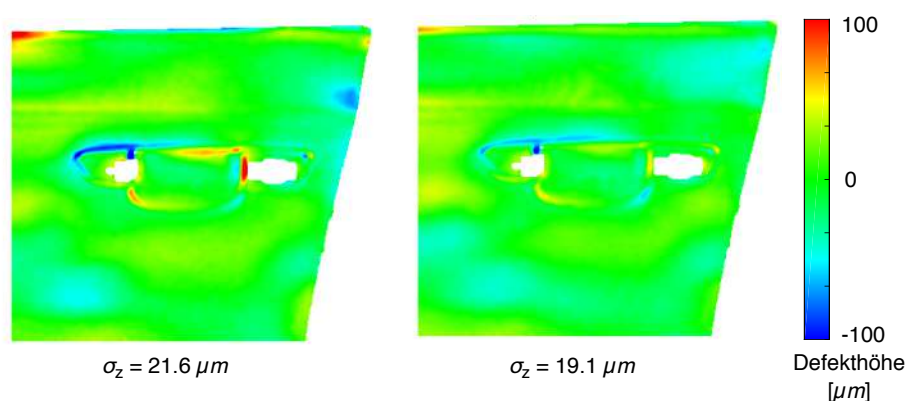


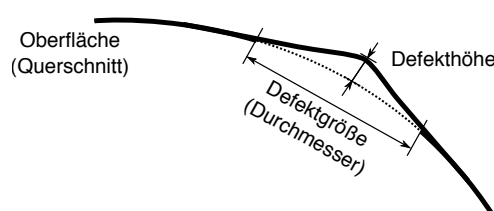
Abbildung 5.5: Überblick der verschiedenen Oberflächenrepräsentationen für Training und Abruf

### 5.2.3 Ergebnisse

**Überblick und Approximationsgüte.** Die Methode wird im Folgenden anhand des Datensatzes „Türgriffmulde“ validiert (vgl. Abbildungen 1.2 bzw. 4.7), der bereits in Abschnitt 4.3.3 vorgestellt wurde. Zunächst wurde die in Abschnitt 5.1 vorgestellte B-Spline-Flächenapproximation angewendet. Für die Anpassung wurde der Mittelwert aller (zuvor



**Abbildung 5.6:** Vergleich der Defektkarten für ein untrainiertes, fehlerfreies Teil. **Links:** herkömmlicher Assoziativspeicher, **rechts:** spline-basierter Assoziativspeicher



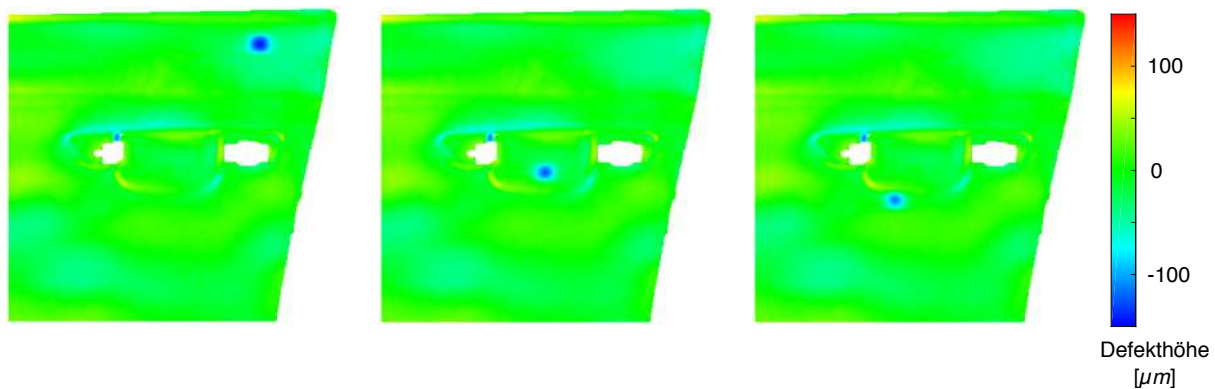
**Abbildung 5.7:** Simulation eines gaußförmigen Defekts auf eine Objekt Oberfläche. Die Defekthöhe liegt üblicherweise einige Größenordnungen unter der lateralen Defektausdehnung

registrierten) Messungen verwendet, um zufällige Fehler und Messrauschen in den Punktwolken zu minimieren. Als Parameter wurden eine absolute Fehlerschwelle von  $e_{abs} = 5 \mu\text{m}$  gewählt, kubische Spline ( $\kappa = \gamma = 3$ ) sowie eine relative Fehlerschwelle  $\alpha = 0.5$ .

Ergebnis der Flächenapproximation ist eine B-Spline-Fläche mit  $172 \times 192$  Kontrollpunkten, dies entspricht einer Datenreduktion um über 72% im Vergleich zur gemessenen Punktwolke. Der Restfehler des B-Spline-Modells liegt bei  $5 \mu\text{m}$  für den Maximalfehler und unter  $1 \mu\text{m}$  für den mittleren Fehler  $\sigma_z$ .

Der spline-basierte Assoziativspeicher wurde anhand von 80 zufällig gewählten Messungen trainiert. Anhand der nicht-trainierten fehlerfreien Messungen wurde die Grundapproximationsgüte untersucht und mit dem herkömmlichen Assoziativspeicher verglichen. Beide Assoziativspeicher wurden mit  $N_0 = 20$  Basisfunktionen trainiert. Das Ergebnis ist qualitativ in den Defektkarten in Abbildung 5.6 dargestellt. Insbesondere an den Kantenbereichen der Türgriffmulde zeigt sich eine deutlich bessere Approximation mit dem spline-basierten Assoziativspeicher. Die mittleren Approximationsfehler unterscheiden sich mit  $21.6 \mu\text{m}$  und  $19.1 \mu\text{m}$  nicht signifikant.

**Defekterkennung.** Die Defekterkennung wurde anhand eines simulierten gaußförmigen Fehlermodells (Höhe:  $150 \mu\text{m}$ , Ausdehnung:  $80 \text{mm} \times 20 \text{mm}$ , siehe Abbildung 5.7) in den untrainierten Messungen simuliert. Die Defektkarten sind für verschiedene De-



**Abbildung 5.8:** Defektkarten „Türgriffmulde“ mit simulierter Delle (Höhe  $20\ \mu\text{m}$ , Ausdehnung  $80\ \text{mm} \times 20\ \text{mm}$ ) für den spline-basierten Assoziativspeicher

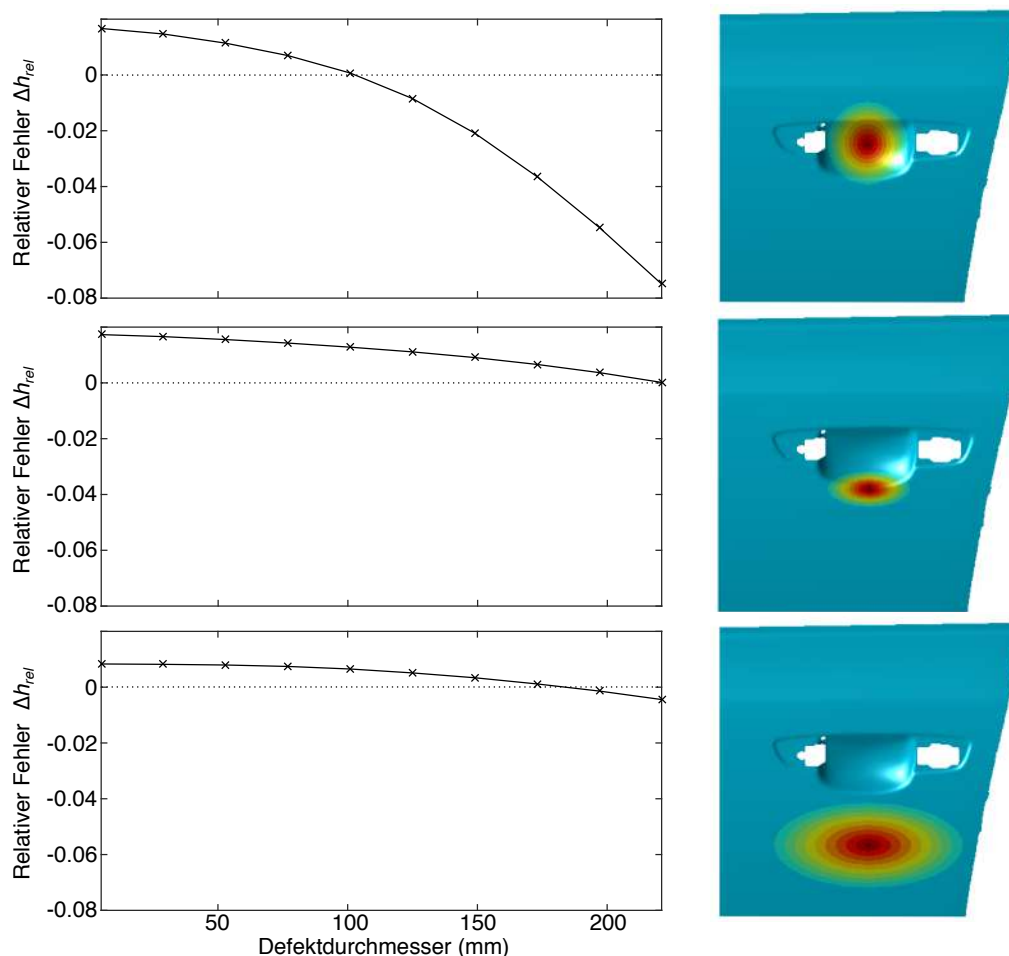
fektpositionen in Abbildung 5.8 dargestellt. Sie geben einen qualitativen Eindruck der Defekterkennung mit dem spline-basierten Assoziativspeicher.

Die simulierte Delle ist deutlich zu erkennen, insbesondere auch in den Gebieten nahe von Kanten bzw. starken Krümmungen der Oberfläche. Es zeigt sich jedoch, dass Defekte abhängig von ihrer Position auf der Oberfläche unterschiedlich gut zu erkennen sind. Die Sensitivität der Inspektion unterscheidet sich also ortsbezogen aufgrund der Oberflächenform und der Form des Defekts. Dies wurde in Abbildung 5.9 ausführlicher untersucht. Hier wurden verschieden geformte gauß'sche Defekte der Höhe  $h = 100\ \mu\text{m}$  an verschiedenen Stellen auf der Oberfläche simuliert und die resultierenden Defektkarten bestimmt.

Der relative Fehler  $\Delta h_{rel} = \frac{d_h - h}{h}$  sagt aus, wie stark die mittels Defektkarte ermittelte Höhe  $d_h$  von der tatsächlichen Defekthöhe  $h$  abweicht. Er ist in Abbildung 5.9, links systematisch für verschiedene Defektgrößen dargestellt. Generell zeigt sich, dass die Defekthöhe für kleine Defekte zu hoch ( $\Delta h_{rel} > 0$ ) und für größere Defekte zu niedrig ( $\Delta h_{rel} < 0$ ) erkannt wird. Befindet sich der Defekt auf einem großflächig glatten Bereich (Abbildung 5.9, unten), so liegt der relative Fehler für die Bestimmung der Defekthöhe bei unter 1%. An Kantenbereichen und insbesondere in der Türgriffmulde steigt der Fehler bei der Defekterkennung auf rund 8% für sehr großflächige Defekte.

### 5.3 Spline-basierte Registrierung von Oberflächen

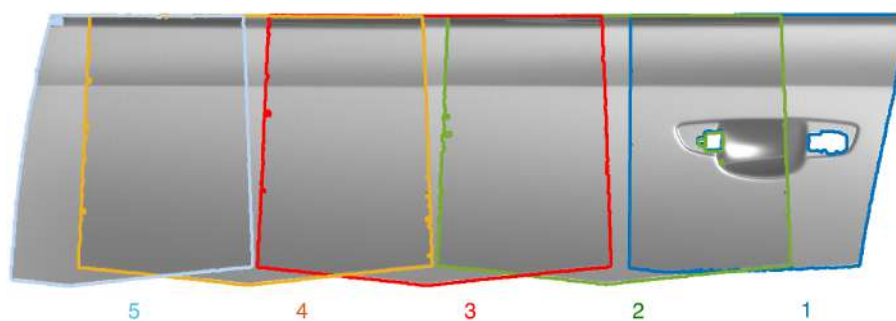
Großflächige Oberflächen können durch existierende Messsysteme bei vorgegebenen Genauigkeitsanforderungen nicht vollständig mit einer einzelnen Messung erfasst werden. Die Gründe dafür wurden bereits in Kapitel 3 analysiert: die Beziehung von Messbereich, Genauigkeit und Auflösung an photogrammetrischen Messsystemen (Abschnitt 3.2.1), sowie die spezifischen Anforderungen der Oberflächeninspektion an Auflösung und Genau-



**Abbildung 5.9:** Relativer Fehler für Defekthöhe einer simulierten gauß'schen Beule bei verschiedenen Defektgrößen und -positionen

igkeit (Abschnitt 3.2). Um dennoch großflächige Oberflächen mit photogrammetrischen Messsystemen zu erfassen, können die folgenden grundlegenden Ansätze genutzt werden:

- **Verwendung mehrerer Sensoren.** In der einfachsten Form können mehrere Sensoren (abgeschlossene Messsysteme) entlang der Oberfläche verteilt werden, die jeweils einen Teilbereich vermessen. Die Einzelmessungen werden anschließend miteinander verrechnet. Dabei kommt es jedoch zu Störungen zwischen den Sensoren, wenn jeweils eine eigene Projektion verwendet wird.
- **Erweiterung des Messsystems.** Um Effekte zwischen den Sensoren zu vermeiden, kann ein Messsystem um weitere Kameras und Projektoren ergänzt werden. Die Fläche wird dann beispielsweise durch gezielte paarweise Triangulation aus Kamerabildern sukzessive aufgebaut. Dies erfordert jedoch einen hohen Aufwand für die individuelle Auslegung und Kalibrierung des Systems.
- **Mehrfachmessung mit einem Sensor.** Wenn das zu vermessende Objekt oder



**Abbildung 5.10:** Visualisierung von 5 teilweise überlappenden Messungen einer Autotür

der Sensor gezielt verschoben werden können (z. B. Bauteile auf einem Fließband oder Sensor an einem Roboterarm), können nacheinander Mehrfachmessungen durchgeführt und miteinander verrechnet werden.

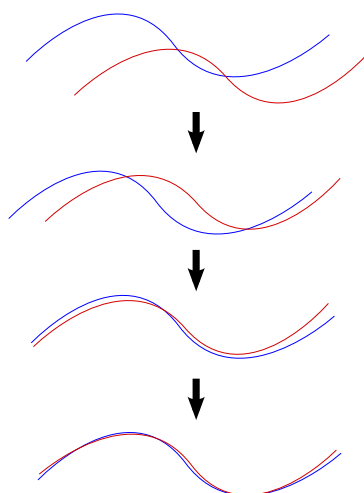
Die *Registrierung* ist der Kernalgorithmus, auf dem die Verrechnung mehrerer 3D-Punktwolken zu einem zusammenhängenden Datensatz aufbaut. Es existieren eine Reihe von Registrierungsverfahren, die jedoch an merkmalsarmen Flächen scheitern, die oft in der Oberflächeninspektion betrachtet werden (siehe Abschnitt 5.3.1). Abbildung 5.10 zeigt das Beispiel einer Autotür, die aus fünf überlappenden Teilmessungen des Blechteils zusammengesetzt wurde. Insbesondere die Teilmessungen 3, 4 und 5 der Abbildung sind merkmalsarm: Sie besitzen wenige charakteristische Oberflächenmerkmale, sodass eine eindeutige Registrierung anhand der Punktwolken nur schwer eindeutig zu bestimmen ist.

Daher wird in 5.3.2 ein Registrierungsverfahren für die Oberflächeninspektion entwickelt, das die Besonderheiten der photogrammetrischen Messung ausnutzt sowie auf der Spline-Oberflächenbeschreibung aufbaut. Ergebnisse werden abschließend in 5.3.3 dargestellt. Die Arbeiten wurden vom Autor erstmalig in [49] präsentiert.

### 5.3.1 Problemanalyse

**Verfahren zur Registrierung von 3D-Punktwolken.** Ziel der *rigiden* oder *starrten Registrierung* von Punktwolken ist es, die Translations- und Rotationsparameter zu bestimmen, sodass eine Ausgangspunktwolke (*source*) nach Transformation möglichst gut mit einer Zielpunktwolke (*target*) übereinstimmt. Es existiert eine Reihe von Verfahren und Anwendungsfälle, die in [64] zusammengestellt sind.

Sind mindestens drei nicht-kollineare, korrespondierende 3D-Punkte in Ausgangs- und Zielpunktwolke bekannt, kann die Transformation geschlossen berechnet werden [139]. Die Herausforderung besteht vor allem im Finden korrespondierender 3D-Punkte in den beiden Messungen. Diese können entweder mit Markern explizit auf der Oberfläche definiert werden oder durch Oberflächenmerkmale implizit beschrieben werden [28, 59].

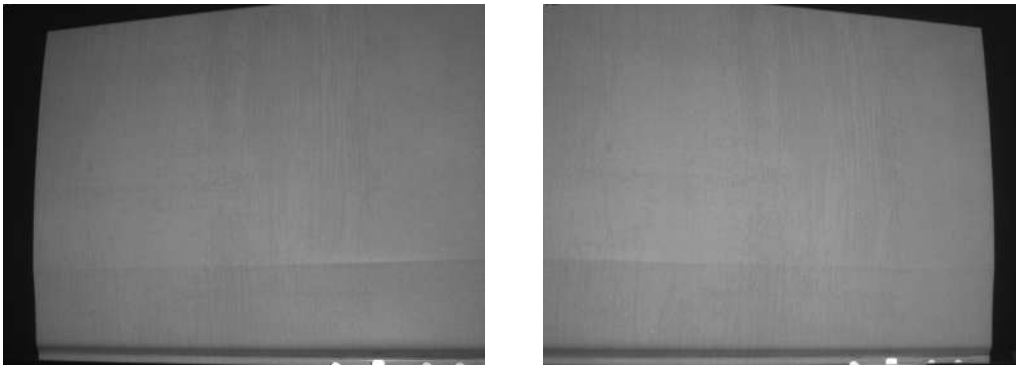


**Abbildung 5.11:** Prinzip des Iterative Closest Point Algorithmus. Die Ausgangs- (blau) und Zielpunkt- wolke (rot) sind schematisch als Flächenquerschnitt visualisiert

Oberflächenmerkmale sind Stellen, in deren Umgebung die Oberfläche eine charakteristische, eindeutige Ausprägung aufweist, zum Beispiel starke Krümmungen, Kanten oder Ecken. Diese können durch Punktdeskriptoren beschrieben werden (z. B. *intrinsic shape signature* [146]). Können korrespondierende Punkte nicht oder nur mit hoher Unsicherheit bestimmt werden weil z. B. keine eindeutigen Oberflächenmerkmale vorhanden sind oder keine Marker auf der Oberfläche angebracht werden dürfen, ist die geschlossene Berechnung nicht möglich.

Eine weitere Klasse von Methoden arbeitet daher mit vollständigen Punktwolken. Ein bedeutender Algorithmus ist *Iterative Closest Point* (ICP) nach BESL und MCKAY [15]. Die lokale Registrierung wird dabei iterativ verbessert; das Funktionsprinzip ist schematisch in Abbildung 5.11 dargestellt. Je Iterationsschritt werden die nächsten Nachbarpunkte zwischen Ausgangs- und Zielpunkt- wolke bestimmt. Anschließend wird eine Transformation der Ausgangspunkte ermittelt, sodass die Abstände der korrespondierenden Punkte nach Transformation minimal sind.

Dabei spielt die *Definition des Abstandsmaßes* (siehe Abschnitt 3.4.2) eine entscheidende Rolle: Die Verwendung des Punkt-Fläche-Abstandes verbessert die Genauigkeit der Registrierung entscheidend [108]; auch ist die Definition eines Flächen-Flächen-Abstandes möglich [122]. Eine weitere Verbesserung des ICP wird durch *Punktabweisung* von nächsten Nachbar-Punktpaaren erreicht. Dies kann z. B. basierend auf einem Distanzschwellenwert, auf einem Schwellenwert für die Normalenorientierung oder statistisch basiert (z. B. durch Abweisung eines gewissen Anteils an Punktpaaren) geschehen. Weiterhin können zusätzliche Informationen für jeden Punkt definiert werden, die die Punktkoordinaten ergänzen. Insbesondere können dies Textur- oder Farbinformationen der Oberfläche sein [77]. Die Suche nach nächsten Nachbarn erfolgt dann im kombinierten Koordinaten-Merkmal-Raum, sodass zu einer hochdimensionalen ICP verallgemeinert werden kann



**Abbildung 5.12:** Linke und rechte Ansicht des Stereo-Kamera-Paars für einen Ausschnitt der Autotür

[62, 71, 137]. In [114] sind verschiedene Verbesserungen des ICP-Algorithmus zusammengefasst.

**Herausforderungen der Registrierung in der Oberflächeninspektion.** Abbildung 5.10 zeigt eine zusammengesetzte Messung einer Autotür. Sie enthält sowohl Bereiche mit vielen Details und starken Krümmungsbereichen (z. B. im Bereich der Türgriffmulde), als auch glatte, ähnliche Bereiche mit geringer Krümmung. Letztere enthalten kaum Oberflächenmerkmale in der geometrischen Form und besitzen eine homogene Oberflächentextur (siehe Kamerabilder Abbildung 5.12). Solche *merkmalsarmen* Oberflächenbereiche sind typisch für die Oberflächeninspektion.

Registrierungsverfahren, die auf wenigen Punkten arbeiten, sind daher nicht einsetzbar. ICP-basierte Verfahren sind prinzipiell auch auf merkmalsarmen Oberflächen einsetzbar, da sie die gesamte Punktwolke betrachten. Der ICP-Algorithmus konvergiert jedoch vergleichsweise langsam und ist anfällig für lokale Minima und Messfehler. Daher ist zum einen eine hinreichend gute Näherung für die initiale Transformation notwendig. Zum anderen sollten systematische und zufällige Messfehler vermieden werden. Für die Punktabweisung von Ausreißern bzw. Messfehlern existieren Algorithmen wie *Random Sample Consensus* (RANSAC) [11, 55]. Weiterhin ist die Registrierung nur dann robust möglich, wenn die Ausgangspunktwolke eine Teilmenge der Zielpunktwolke ist [124].

Textur-basierte ICP-Algorithmen sind eine mögliche Lösung Punktwolken, die hinsichtlich ihrer Oberflächenform merkmalsarm sind, jedoch eine hinreichend eindeutige Textur aufweisen. Für photogrammetrische Verfahren ist das Mapping der Texturinformationen auf die 3D-Punktwolke nicht hinreichend präzise. Kleine Fehler in der Kamerakalibrierung pflanzen sich als Triangulationsfehler und letztendlich als Projektionsfehler zwischen Kamerabild und Punktwolke fort. Außerdem führt die Projektion zu einem Informationsverlust, da die Auflösung der 3D-Punktwolke üblicherweise geringer als die Auflösung der 2D Kamerabilder ist. Dies ist insbesondere dann kritisch, wenn die Oberflächentextur sehr fein ist und eine hochpräzise Ausrichtung der Punktwolken gefordert wird.

Für die meisten Anwendungen des ICP-Algorithmus ist eine präzise geometrische Rekonstruktion entscheidend. In der Oberflächeninspektion ist die geometrische Präzision weniger entscheidend. Vielmehr soll die zusammengesetzte Fläche keine lokalen Verwerfungen oder Sprünge enthalten, die die Erkennung von Oberflächendefekten erschwert.

### 5.3.2 Beschreibung des entwickelten Ansatzes

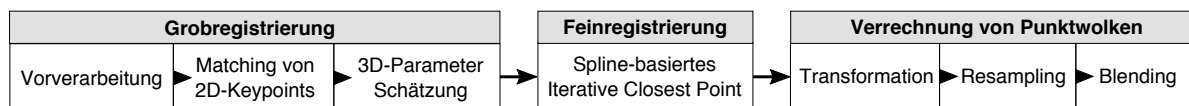


Abbildung 5.13: Verarbeitungskette und Teilalgorithmen

Die verschiedenen Herausforderungen bei der Registrierung von Punktwolken in der Oberflächeninspektion werden durch die in Abbildung 5.13 dargestellten Teilalgorithmen gelöst. Für die *Grobregistrierung* (Abschnitt 5.3.2.1) wurde aufgrund der schwachen Merkmalsausprägung in den 3D-Punktwolken auf die Texturdaten der 2D-Kamerabilder zurückgegriffen. Verarbeitete Bleche besitzen eine weitgehend homogene Oberflächentextur. Neben einer Mikrotextrur können jedoch Glanz- oder Schattenstellen sowie Unvollkommenheiten durch den Umformprozess zu leichten Inhomogenitäten in der Textur führen [57]. Bei Verwendung des Phase-Shift-Verfahrens können Texturbilder mit hoher Farb- bzw. Grauwertaufösung berechnet werden, in denen auch schwache Strukturen auf der Oberfläche sichtbar sind. Die *Feinregistrierung* (Abschnitt 5.3.2.2) basiert auf dem ICP Algorithmus. Eine Erhöhung der Genauigkeit wird erreicht durch die Beschneidung der Ausgangspunktwolke auf den durch die Grobregistrierung bekannten Überlappungsbereich, sowie durch die Nutzung einer B-Spline-Darstellung der Zielpunktwolke.

Mit der resultierenden Transformation kann die *Verrechnung der Punktwolken* durchgeführt werden (Abschnitt 5.3.2.3). Als erster Schritt müssen alle Punktwolken in ein gemeinsames Koordinatensystem transformiert werden. Im zweiten Schritt werden alle Messungen entsprechend des neuen Zielrasters abgetastet. Hierfür werden die B-Spline-Darstellung der Oberflächen genutzt. Im letzten Schritt kommt es zur eigentlichen Verrechnung der transformierten, neu abgetasteten Messungen. Dies geschieht durch ein Blending zusammengehöriger Tiefenwerte im Raster des neuen Messkoordinatensystems.

#### 5.3.2.1 Grobregistrierung

Die Grobregistrierung besteht aus drei Teilschritten: Zunächst werden die Kamerabilder vorverarbeitet, indem interessante Teilbereiche maskiert und in ein Texturbild überführt werden (*Vorverarbeitung*). Anschließend werden charakteristische 2D-Keypoints extrahiert und ihre Umgebungen pixelweise zwischen zwei unterschiedlichen Messungen verglichen, um Übereinstimmungen zu finden (*Matching*). Zuletzt werden die korrelierenden 2D-Punkte in den 3D-Raum projiziert, mittels RANSAC Verfahren auf robuste



3D-Punktpaare reudziert und daraus die Grobtransformation berechnet (*Parameterschätzung*).

*Eingangsgrößen* für den texturbasierten Registrierungsalgorithmus sind die Ausgangs- und Zielpunkt看ke sowie die zugehörigen Rohdaten eines Stereo-Kamerasystems: die Kamerakalibrierung (siehe Abschnitt 3.2.1) und Kamerabilder. Im folgenden werden Indizes  $\mathbf{M}$  und  $\mathbf{C}$  eingeführt, die die  $\mathbf{M}$ -te Messung ( $\mathbf{M} \in \{\mathbf{M}_0, \mathbf{M}_1\}$ ) sowie die  $\mathbf{C}$ -te Kamera des Stereopaars ( $\mathbf{C} \in \{\mathbf{C}_0, \mathbf{M}_1\}$ ) referenzieren:

- **Kamerakalibrierung**

$$(u^{\mathbf{C}}, v^{\mathbf{C}}) = f_{cam}^{\mathbf{C}}(x^{\mathbf{M}}, y^{\mathbf{M}}, z^{\mathbf{M}}) \quad (5.31)$$

- **Kamerabilder**

$$G_j^{\mathbf{C}, \mathbf{M}}(u^{\mathbf{C}, \mathbf{M}}, v^{\mathbf{C}, \mathbf{M}}) \quad (5.32)$$

- **3D-Punkt看ken**

$$X^{\mathbf{M}} = \{\mathbf{x}_i^{\mathbf{M}} = (x_i^{\mathbf{M}}, y_i^{\mathbf{M}}, z_i^{\mathbf{M}}) \mid i = 1, 2, \dots, N^{\mathbf{M}}\} \quad (5.33)$$

*Ausgabegrößen* des Algorithmus sind die Rotationsmatrix  $\mathbf{R}^{\mathbf{M}}$  und der Translationsvektor  $\mathbf{t}^{\mathbf{M}}$  je Messung, mit der die Punkt看ken aus dem Sensorkoordinatensystem in ein gemeinsames Weltkoordinatensystem überführt werden können:

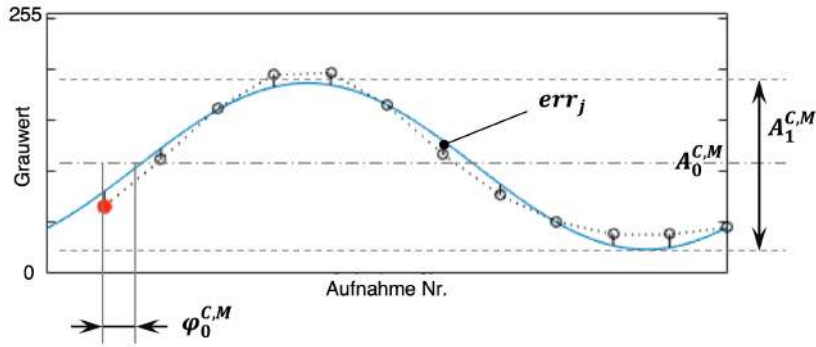
$$\tilde{X}^{\mathbf{M}} = \mathbf{R}^{\mathbf{M}} \cdot X^{\mathbf{M}} + \mathbf{t}^{\mathbf{M}} \quad (5.34)$$

**Vorverarbeitung.** Beim Phase-Shift-Verfahren wird eine Bildserie  $G_j^{\mathbf{C}, \mathbf{M}}$  mit wechselnden projizierten Streifenlichtmustern aufgenommen. Daraus ergibt sich ein sinusförmiger Helligkeitsverlauf für jedes Bildpixel, aus dem sich verschiedene Größen ableiten lassen. Die wichtigsten davon sind der mittlere Grauwert  $A_0^{\mathbf{C}, \mathbf{M}}$  oder *Hellbild* sowie die Amplitude des Phasenverlaufs  $A_1^{\mathbf{C}, \mathbf{M}}$  (siehe Abbildung 5.14).

Hieraus wird zunächst eine *Maske*  $\mathbf{m}^{\mathbf{C}, \mathbf{M}}$  berechnet. Ihre Aufgabe ist es, Bereiche im Bild zu maskieren, in denen die Textur voraussichtlich gut auswertbar ist. Diese Bereiche sollen im weiteren Verlauf weiterverarbeitet werden und besitzen gute Beleuchtungs- und Reflexionseigenschaften, z. B. keine Glanzstellen, Über- oder Unterbelichtung.

Dafür werden zunächst folgende Teilmasken  $\mathbf{m}_0^{\mathbf{C}, \mathbf{M}}$  und  $\mathbf{m}_1^{\mathbf{C}, \mathbf{M}}$  definiert, die jeweils mit spezifischen Anforderungen verknüpft sind:

$$\mathbf{m}_0^{\mathbf{C}, \mathbf{M}} = \left( A_0^{\mathbf{C}, \mathbf{M}} - \frac{255}{2} \right)^2 < (255 \cdot g_{reject})^2 \quad (5.35)$$



**Abbildung 5.14:** Wichtige Größen am Grauwertverlauf eines Bildpixels für das Phase-Shift-Verfahren

$$\mathbf{m}_1^{\text{C,M}} = A_1^{\text{C,M}} > \frac{255}{2} \cdot (0.5 - g_{\text{reject}}) \quad (5.36)$$

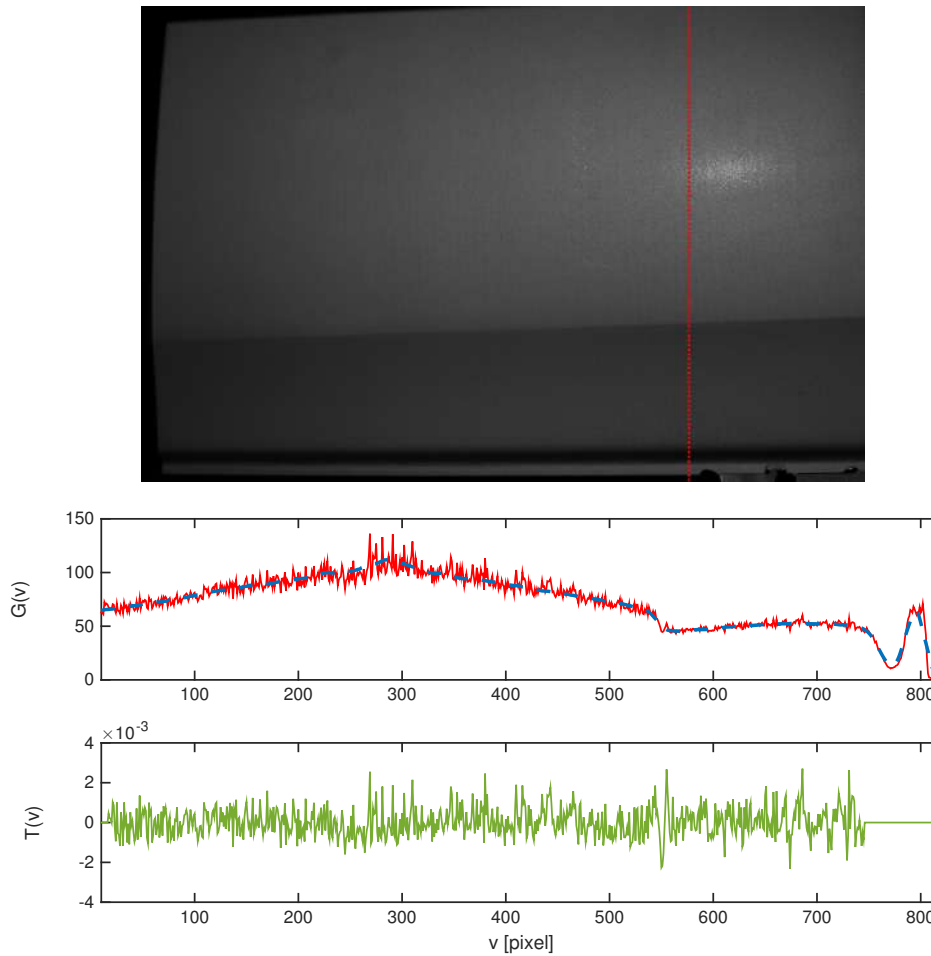
Dabei werden durch Formel (5.35) Pixel bevorzugt, die im mittleren Grauwertbereich liegen, also nicht Über- oder Unterbelichtet sind. Durch Formel (5.36) werden Bereiche mit hoher Amplitude bevorzugt, also Bereiche mit guter Ausleuchtung. Durch den Parameter  $g_{\text{reject}} \in [0 \dots 0.5]$  wird der Grad der Maskierung bestimmt. Für  $g_{\text{reject}} = 0$  findet keine Maskierung statt, alle Bildpunkte werden weiterverarbeitet; für  $g_{\text{reject}} = 0.5$  findet eine volle Maskierung statt, d.h. es werden keine Bereiche in den Bildern weiterverarbeitet.

Die Gesamtmaske ergibt sich aus Erosion der UND-verknüpften Teilmasken mit einem rechteckigen Strukturelement  $\mathbf{s}_m \in \mathbb{R}^{2 \cdot r_{\text{rem}} + 1}$ . Durch die Erosion werden Randbereiche effektiv von der Weiterverarbeitung ausgeschlossen.

$$\mathbf{m}^{\text{C,M}} = \left( \mathbf{m}_0^{\text{C,M}} \wedge \mathbf{m}_1^{\text{C,M}} \right) \ominus \mathbf{s}_m \quad (5.37)$$

Zusätzlich zur Maske kann aus dem Hellbild  $A_0^{\text{C,M}}$  ein *Texturbild* durch geeignete Filterung extrahiert werden. Der Vorgang ist in Abbildung 5.15 dargestellt. Hier ist oben exemplarisch ein Hellbild abgebildet. Die Metalloberfläche ist reflektierend und wird unter einer stark ungleichmäßig Beleuchtung beobachtet. Dies kann durch eine Tiefpassfilterung ausgeglichen werden. In Abbildung 5.15 (Mitte) ist das Ergebnis einer Tiefpassfilterung  $F^{\text{C,M}}$  zusammen mit dem Grauwertverlauf des Hellbildes  $A_0^{\text{C,M}}$  exemplarisch entlang einer Bildspalte dargestellt. Um ein Texturbild  $T^{\text{C,M}}$  zu erhalten, das möglichst unabhängig von Beleuchtungsbedingungen und Reflektionsverhalten der Oberfläche ist, wird die Differenz von Hellbild und gefiltertem Bild verwendet und auf  $F^{\text{C,M}}$  normiert. Dadurch werden sehr helle Bildbereiche gedämpft und dunkle Bildbereiche verstärkt:

$$\tilde{T}^{\text{C,M}}(u^{\text{C}}, v^{\text{C}}) = \left| \frac{G^{\text{C,M}} - F^{\text{C,M}}}{(F^{\text{C,M}})^2} \right| \quad (5.38)$$



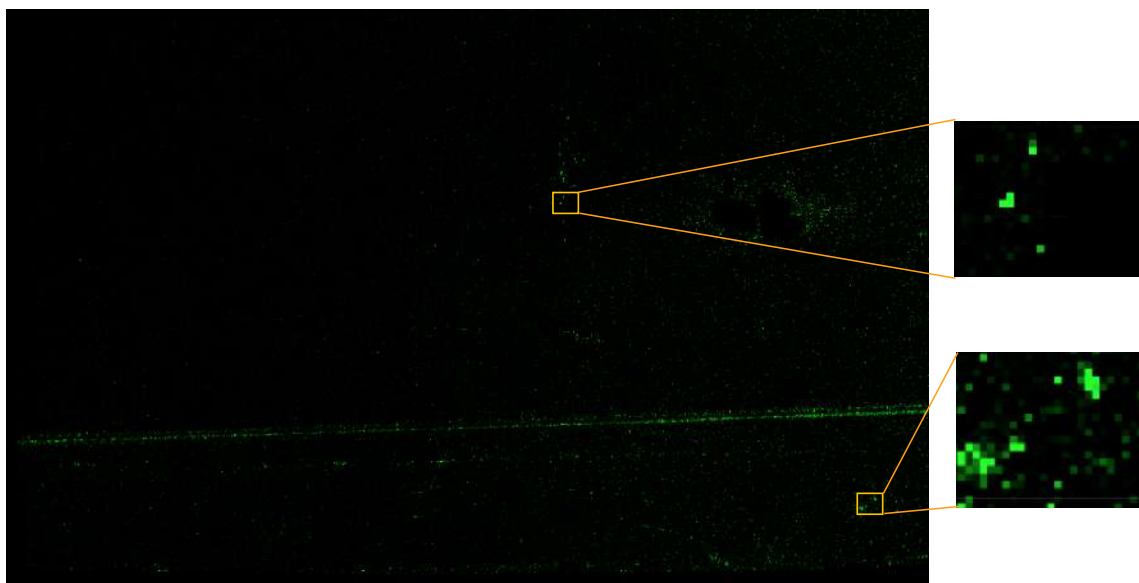
**Abbildung 5.15:** Texturextraktion aus dem Hellbild. **Oben:** Hellbild für einen Ausschnitt einer vermessenen Autotür mit markierter Bildspalte, **Mitte:** Grauwertverlauf des Hellbildes entlang der markierten Bildspalte (rot) sowie zugehöriger gefilterter Verlauf (blau gestrichelt), **Unten:** Grauwertverlauf des Texturbildes entlang der Bildspalte, nach Normalisierung und Maskierung

Um das finale Texturbild  $T^{c,M}$  zu erhalten, wird die zuvor bestimmte Maske durch elementweise Multiplikation angewendet:

$$T^{c,M}(u^c, v^c) = \tilde{T}^{c,M} \circ \mathbf{m}^c \quad (5.39)$$

Das Ergebnis ist in Abbildung 5.15 (unten) für eine Bildspalte und in 5.16 für das gesamte Bild dargestellt. Die extrahierte Textur  $T^{c,M}$  verläuft nun uniform innerhalb eines gleichmäßigen Bandes.

**Matching von 2D Keypoints.** In den extrahierten Texturbildern werden nun Bereiche mit charakteristischen Texturmerkmalen bestimmt und sogenannte *Keypoints* zugeordnet. Dabei liefert eine Schwellenwertbinarisierung und anschließende Detektion der größten zusammenhängenden Texturbereiche hinreichend sinnvolle Keypoints. Eine Merk-



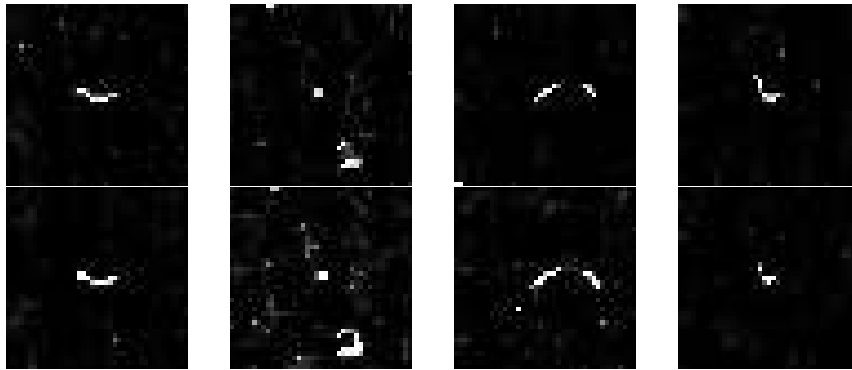
**Abbildung 5.16:** Resultierendes Texturbild mit einzelnen Texturdetails

malsextraktion ist aufgrund der schwachen Texturausprägung nicht sinnvoll. Stattdessen werden korrespondierende Punkte in den Kameras  $C$  durch Blockmatching bestimmt (siehe Abschnitt 3.2.2), es wird die normierte Kreuzkorrelationsfunktion nach Formel (3.7) verwendet.

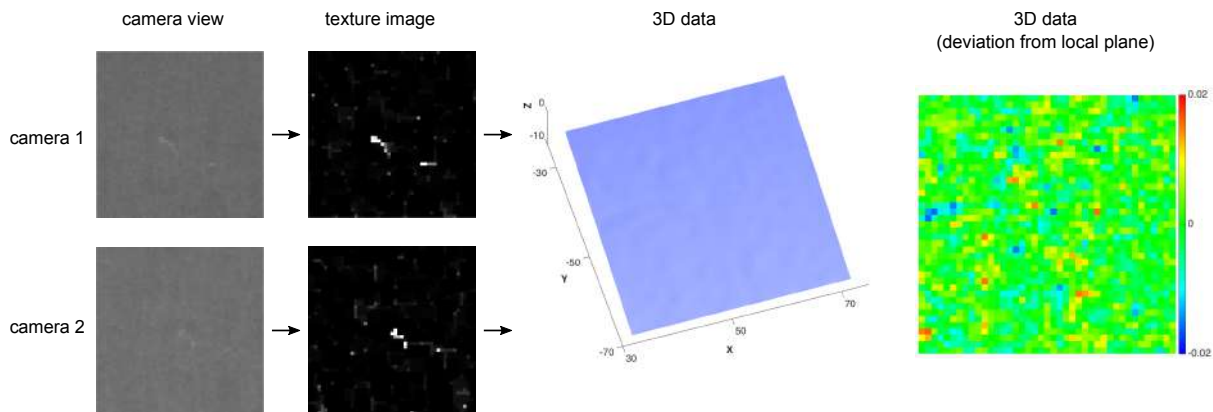
Die Kreuzkorrelation ist robust gegen Bildrauschen; durch die Normalisierung wird eine zusätzliche Robustheit gegen lokale Beleuchtungsvariationen erzielt [9, 140]. Die Blockgröße muss passend zur vorliegenden Oberfläche empirisch gewählt werden. Dazu kann die Auto-Korrelation von einigen Beispielblöcken beobachtet werden. Die Blockgröße sollte eine Mindestgröße erfüllen, damit keine Mehrdeutigkeiten in der Auto-Korrelation sichtbar sind. Gleichzeitig sollte die Blockgröße gering gehalten werden, um schnellere Rechenzeiten zu erreichen und robuster gegen systematische Fehler (z. B. durch unterschiedliche Blickwinkel oder fehlerhafte Kamerakalibrierung) zu sein.

In Abbildung 5.17 sind einige Beispiele für erfolgreich gematchte Texturblöcke der Größe  $41 \times 41$  pixel gegeben. In Abbildung 5.18 ist ein weiteres Beispiel für Texturblöcke im Kontext der verschiedenen Arbeitsschritte, beginnend vom Kamerabild bis zur zugehörigen Punktwolke, dargestellt.

Die Korrelation der Keypoint-Bereiche wird paarweise zwischen Ausgangs- und Zielpunktwolke ( $M_0, M_1$ ) und für alle möglichen Kamera-paare  $C_0, C_1$  wiederholt, um eine höhere Robustheit gegenüber Projektionsfehler zu erreichen. Es ergeben sich also insgesamt 4 mögliche Bildpaar-Kombinationen, für die alle Keypoint-Kombinationen per nKKF verglichen werden. Es werden nur die besten Matches behalten (d.h. maximale Werte der nKKF). Es verbleiben für typische Blechoberflächen nicht mehr als 100 Punkt-paare.



**Abbildung 5.17:** Beispiele für erfolgreich gematchte Texturblöcke der Größe  $41 \times 41$  pixel in Teilmessung 2 (oben) und 3 (unten).



**Abbildung 5.18:** Bereich um einen Keypoint in den Kamerabildern, Texturbildern und als Ausschnitt der 3D-Punktvolke. Die 3D-Daten sind zusätzlich als Defektkarte zu einer lokalen Ebene dargestellt und zeigen keine charakteristischen Merkmale

**3D-Parameterschätzung.** Die aus dem Matching resultierenden Punktpaare sind Kandidaten für korrelierende 2D-Bildpunkte. Durch Anwendung der inversen Kamerakalibrierung  $f_{cam}^{-1}$  können die Keypoints als Strahlen in den Raum rückprojiziert und mit der 3D-Oberfläche geschnitten werden. Dazu existieren eine Reihe von Standard-Algorithmen wie z. B. das Line Clipping [76]. Es ergeben sich 3D-Punkt Mengen  $Q^{M_0} = \{ \mathbf{q}_k^{M_0} \in \mathbb{R}^3 \mid k = 1 \dots \kappa \}$  und  $Q^{M_1} = \{ \mathbf{q}_k^{M_1} \in \mathbb{R}^3 \mid k = 1 \dots \kappa \}$  für Messung  $M_0$  und  $M_1$ , deren Elemente paarweise Kandidaten für korrelierende 3D Keypoints in den lokalen Sensorkoordinatensystemen sind.

Ziel der Parameterschätzung ist es, für die Punktpaare  $(\mathbf{q}_k^{M_0}, \mathbf{q}_k^{M_1})$  robuste Transformationen  $\mathbf{R}^{M_1}, \mathbf{t}^{M_1}$  zu finden, mit denen die Punkte der Ausgangsmessung  $M_1$  bei möglichst geringem Fehler mit den zugehörigen Punkten der Zielmessung  $M_0$  übereinstimmen. Dies führt zu einem Minimierungsproblem der kleinsten Fehlerquadrate, also den Entfernungen

der Ausgangs- zu den Zielkeypoints:

$$E(\mathbf{R}^{\mathbf{M}1}, \mathbf{t}^{\mathbf{M}1}) = \sum_{k=1}^{\kappa} [\text{err}_k^{\mathbf{M}0, \mathbf{M}1}(\mathbf{R}^{\mathbf{M}1}, \mathbf{t}^{\mathbf{M}1})]^2 \rightarrow \min_{\mathbf{R}^{\mathbf{M}1}, \mathbf{t}^{\mathbf{M}1}} \quad (5.40)$$

$$\text{mit} \quad \text{err}_k^{\mathbf{M}0, \mathbf{M}1}(\mathbf{R}^{\mathbf{M}1}, \mathbf{t}^{\mathbf{M}1}) = \mathbf{q}_k^{\mathbf{M}0} - (\mathbf{R}^{\mathbf{M}1} \cdot \mathbf{q}_k^{\mathbf{M}1} + \mathbf{t}^{\mathbf{M}1}) \quad (5.41)$$

Die Homogenität der Oberfläche führt dazu, dass eine Vielzahl der Kandidaten keine tatsächlich korrespondierenden Punkte oder mehrdeutige Korrespondenzen sind. Für die Daten der Autotür sind etwa 70% der  $\kappa$  Kandidaten für Keypoints fehlerhaft. Um dennoch eine robuste Auswahl von korrespondierenden Punkten zu erhalten, wurde das *RANSAC Verfahren* angepasst und angewendet. Die Grundidee des Verfahrens ist es, die Transformation für eine zufällige Auswahl von wenigen (mindestens drei) Punktpaaren aus den Kandidaten zu bestimmen (Formel (5.40)) und auf die Gesamtmenge anzuwenden. Anschließend wird die Anzahl der Punktpaare ermittelt, für die der Betrag des resultierenden Fehlers (5.41) kleiner als ein Schwellenwert  $\tau_{\text{err}}$  ist (sog. *Consensus-Menge*). Dieser Vorgang wird mehrmals mit einer neuen zufälligen Auswahl wiederholt. Nach Abschluss der Iterationen wird die Transformation aus der größten ermittelten Consensus-Menge bestimmt und zurückgegeben. Der Gesamtalgorithmus ist in Tabelle 5.3 dargestellt.

### 5.3.2.2 Feinregistrierung

Ziel des Verfahrens ist es, ausgehend von einer Näherung (Grobregistrierung) eine Transformation zu finden, für die der Abstand der Ausgangs- zur Zielpunktswolke minimal ist. Als Distanzmaß wird meist der mittlere euklidische Punktabstand verwendet. Das bestimmen zusammengehöriger Punktpaare stellt dabei das größte Problem dar. Beim ursprünglichen ICP wird dafür der nächste Nachbarpunkt angenommen und iterativ eine lokale Lösung gefunden. Bei einer hinreichend genauen Näherung kann das Verfahren zur globalen Lösung konvergieren.

Die Verwendung des Punkt-Punkt-Abstandes hat jedoch mehrere Nachteile: Das ICP Verfahren konvergiert langsam, da die Bewegungsrichtung in den Iterationen nicht orthogonal zur Oberfläche steht. Außerdem ist prinzipiell mit einem Restfehler zu rechnen, der durch den Abstand der Punkte bzw. die Größe des Punktraster vorgegeben wird. Daher wurde die Spline-Darstellung  $\mathbf{x}_s^{\mathbf{M}0}$  der Zielpunktswolke berechnet (vgl. Abschnitt 5.1.1) und stattdessen der Punkt-Spline Abstand verwendet (vgl. Abschnitt 3.4.2). Das Gesamtverfahren ist in Tabelle 5.4 dargestellt. Die Hauptiteration wird abgebrochen, wenn eines der folgenden *Konvergenzkriterien* erreicht ist:

- (K1) Die Änderung des Punkt-Spline-Abstandes liegt unterhalb eines Schwellenwertes.
- (K2) Die Änderung der Translation und Rotation liegt unterhalb eines Schwellenwertes.

**Tabelle 5.3:** Algorithmus: RANSAC-Verfahren zum robusten Finden der Transformation für Punktpaare

<b>Input</b>	: Überlappende Messungen $M_0, M_1$ mit Kandidaten für paarweise korrespondierende 3D Keypoints $Q^{M_0}, Q^{M_1}$ , Fehlerschwelle $\tau_{err}$
<b>Output:</b>	Transformationsparameter $\mathbf{R}^{M_1}, \mathbf{t}^{M_1}$
1	$n_{cons,max} \leftarrow 0$ // Größte Consensus-Menge
2	<b>for</b> $I = 1 \dots I_{max}$ <b>do</b> // Maximale Iterationszahl
3	$n_{cons} \leftarrow 0$ // Aktuelle Consensus-Menge
4	Wähle zufällige Keypoint-Indizes $\{k_1, k_2, k_3\} \subset \{1, \dots, \kappa\}$
5	$\tilde{Q}^{M_0} \leftarrow \{\mathbf{q}_{k_1}^{M_0}, \mathbf{q}_{k_2}^{M_0}, \mathbf{q}_{k_3}^{M_0}\} \subset Q^{M_0}$
6	$\tilde{Q}^{M_1} \leftarrow \{\mathbf{q}_{k_1}^{M_1}, \mathbf{q}_{k_2}^{M_1}, \mathbf{q}_{k_3}^{M_1}\} \subset Q^{M_1}$
7	$(\mathbf{R}_{cons}, \mathbf{t}_{cons}) \leftarrow$ Formel (5.40) für $\tilde{Q}^{M_0}, \tilde{Q}^{M_1}$
8	<b>foreach</b> $\mathbf{q}_k^{M_0} \in Q^{M_0}$ <b>und</b> $\mathbf{q}_k^{M_1} \in Q^{M_1}$ <b>do</b> // für alle Keypoints
9	<b>if</b> $[\text{err}_k^{M_0, M_1}(\mathbf{R}_{cons}, \mathbf{t}_{cons})]^2 < \tau_{err}^2$ <b>then</b> // Fehlerberechnung nach (5.41)
10	$n_{cons} \leftarrow n_{cons} + 1$ // gehört zu Consens-Menge
11	<b>end</b>
12	<b>end</b>
13	<b>if</b> $n_{cons} > n_{cons,max}$ <b>then</b> // neue „beste“ Transformation
14	$n_{cons,max} \leftarrow n_{cons}$
15	$\mathbf{R}^{M_1} \leftarrow \mathbf{R}_{cons}$
16	$\mathbf{t}^{M_1} \leftarrow \mathbf{t}_{cons}$
17	<b>end</b>
18	<b>end</b>

**(K3)** Die maximale Anzahl der Iterationen ist erreicht.

Die Genauigkeit des Verfahrens konnte weiter verbessert werden, indem die Ausgangspunktwolke auf den Überlappungsbereich entsprechend der Grobregistrierung beschnitten wird. Da diese fehlerbehaftet ist, bietet es sich an, die ermittelte Überlappungsfläche weiter um einige Randpunkte zu verkleinern. Weiterhin können Punktpaare von der Bestimmung der Transformation ausgeschlossen werden, die eine maximale Distanzschwelle überschreiten.

**Tabelle 5.4:** Algorithmus: Spline-basiertes Iterative Closest Point Verfahren

```

Input : Ausgangspunktwolke  $X^{M1} = \{\mathbf{x}_i^{M1} \in \mathbb{R}^3\}$  aus Messung  $M_1$ ,
          Ziel-Splinefläche  $\mathbf{x}_s^{M0}(u, v)$  aus Messung  $M_0$ 
Output: Transformationsparameter  $\mathbf{R}^{M1}, \mathbf{t}^{M1}$ 

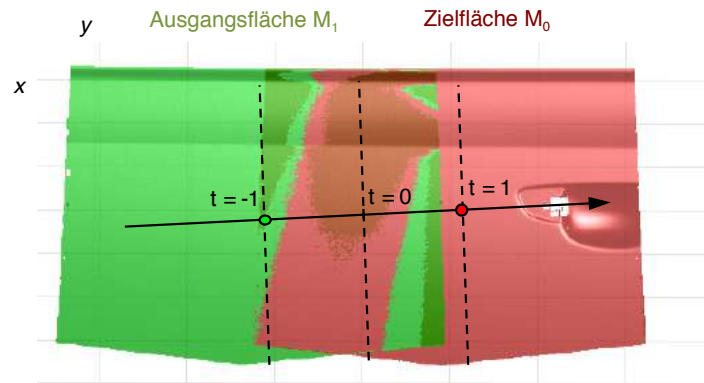
1  $\mathbf{R}^{M1} \leftarrow$  Einheitsmatrix  $3 \times 3$ 
2  $\mathbf{t}^{M1} \leftarrow (0, 0, 0)$ 
3  $\{\tilde{\mathbf{x}}_i^{M1}\} \leftarrow X^{M1}$ 
4 for  $I = 1 \dots I_{max}$  do // Maximale Iterationszahl (K3)
5   foreach  $\tilde{\mathbf{x}}_i^{M1}$  do // nächster Nachbar  $\forall i$ 
6      $\tilde{\mathbf{x}}_i^{M0} \leftarrow$  nächster Spline-Punkt von  $\tilde{\mathbf{x}}_i^{M1}$  zu  $\mathbf{x}_s^{M0}(u, v)$ 
7   end
8    $E = \sum_{\forall i} \|(\mathbf{R}_I^{M1} \cdot \tilde{\mathbf{x}}_i^{M1} + \mathbf{t}_I^{M1}) - \tilde{\mathbf{x}}_i^{M0}\|^2 \rightarrow \min_{\mathbf{R}^{M1}, \mathbf{t}^{M1}}$  // Parameterschätzung
9   foreach  $\tilde{\mathbf{x}}_i^{M1}$  do // Transformation  $\forall i$ 
10     $\tilde{\mathbf{x}}_i^{M1} \leftarrow \mathbf{R}_I^{M1} \cdot \tilde{\mathbf{x}}_i^{M1} + \mathbf{t}_I^{M1}$ 
11  end
12   $\mathbf{R}^{M1} \leftarrow \mathbf{R}_I^{M1} \cdot \mathbf{R}^{M1}$  // Parameter-Update
13   $\mathbf{t}^{M1} \leftarrow \mathbf{R}_I^{M1} \cdot \mathbf{t}^{M1} + \mathbf{t}_I^{M1}$ 
14  if  $\Delta E < E_{thresh}$  then // Konvergenzkriterium (K1)
15    exit
16  end
17  if  $\|\mathbf{R}_I^{M1}\| < R_{thresh}$  und  $\|\mathbf{t}_I^{M1}\| < t_{thresh}$  then // Konvergenzkriterium (K2)
18    exit
19  end
20 end
    
```

### 5.3.2.3 Verrechnung der Punktwolken

Die zuvor ermittelten Transformationen können auf die Messungen angewendet werden, sodass sich diese nun in einem einheitlichen Koordinatensystem befinden. Die transformierten Oberflächen müssen nun zu einer Gesamtfläche zusammengeführt werden. Im folgenden wird die Weiterverarbeitung von Distanzkarten angenommen, dazu müssen die transformierten Oberflächen zunächst im neuen  $x$ - $y$ -Raster abgetastet werden. Da sie ohnehin als B-Spline-Fläche vorliegen, kann die parametrische Darstellung für die Abtastung genutzt werden. Es ergibt sich für jede Messung eine Tiefenkarte  $\mathbf{z}^M$ .

In den Überlappungsbereichen müssen nun die  $z$ -Werte miteinander verrechnet werden.





**Abbildung 5.19:** Blending-Gerade zur Verrechnung von Ausgangs- und Zielfläche

Dazu wird eine Übergangsgerade oder *Blending*-Gerade definiert. Sie verbindet die geometrischen Schwerpunkte zweier Punktwolken miteinander. An den Schwerpunkten wird der Geradenparameter  $t = -1$  bzw.  $t = 1$  gewählt, dies ist in Abbildung 5.19 dargestellt.

Ausschlaggebend für die Verrechnung der Tiefenwerte ist der Parameter  $t$ , der sich durch Projektion des zugehörigen 3D-Punktes auf die Blending-Gerade ergibt. Die Gewichtung der Tiefenwerte erfolgt über eine Gewichtungsfunktion  $w(t)$ , die exemplarisch in Abbildung 5.20 dargestellt ist. Für den verrechneten Wert  $z^{M_0, M_1}$  ergibt sich dann als:

$$z^{M_0, M_1}(x, y) = w(t) \cdot z^{M_0}(x, y) + [1 - w(t)] \cdot z^{M_1}(x, y) \quad (5.42)$$

$$\text{mit } w(t) = 0.5 + \frac{\tanh(\alpha_{blend} \cdot t)}{2} \quad (5.43)$$

Dabei stellt der Parameter  $\alpha_{blend}$  die Steilheit der Gewichtungsfunktion und somit die Breite des Übergangsbereiches ein. Dadurch wird ein glatter Übergang zwischen den Teilflächen erreicht, der die weitere Auswertung mit Methoden der Oberflächeninspektion möglich macht.

### 5.3.3 Ergebnisse

Für die Validierung des Algorithmus wurden erneut die 5 teilweise überlappenden Messungen einer Autotür verwendet. Das Ergebnis der Registrierung ist in Abbildung 5.10 dargestellt, hier sind die verschiedenen Teilmessungen farblich umrandet. Die Messungen entstammen wieder einem photogrammetrisches Messsystem mit dem Phase-Shift Verfahren, einem Messbereich von ca.  $300 \text{ mm} \times 400 \text{ mm}$  und einer Tiefenauflösung von etwa  $\Delta z = 10 \mu\text{m}$  auf einem regelmäßigen  $1 \text{ mm} \times 1 \text{ mm}$ -Raster.

Für die Verwendung in der Oberflächeninspektion dürfen vor allem keine lokalen Verwer-

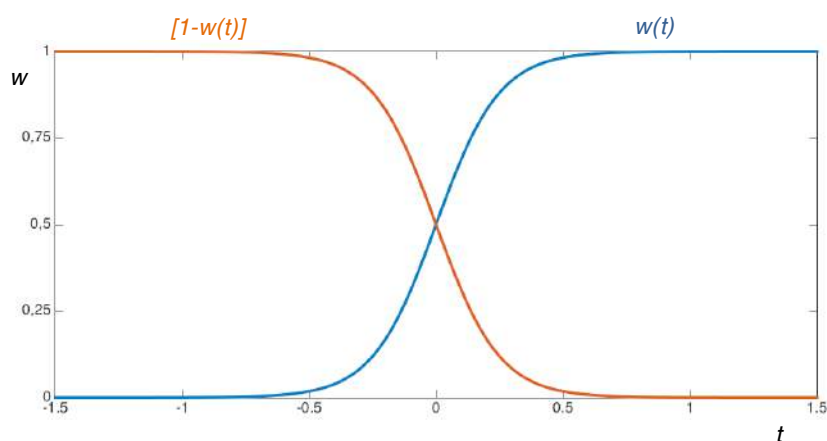


Abbildung 5.20: Gewichtsfunktion für das Blending von Tiefenkarten

funken oder Sprünge in die Daten eingebracht werden. Um dies zu bewerten, werden im Folgenden die Defektkarten der Überlappungsbereiche zur Bewertung herangezogen.

Tabelle 5.5: Ergebnisse der Grobregistrierung. Anzahl der Punktpaare aus dem Blockmatching (Kandidaten), Anzahl Punkte der Consensus-Menge, punktwiser Fehler im Überlappungsbereich (Minimum-, Maximum- und mittlerer quadratischer Fehler)

Mess.	Anzahl Punkte		Fehler Überlappungsbereich		
	Kandidaten	Consensus	Min [mm]	Max [mm]	MSE [mm]
$M_1, M_2$	344	82	-0.457	0.333	0.105
$M_2, M_3$	194	23	-0.549	0.701	0.096
$M_3, M_4$	152	26	-0.599	0.417	0.139

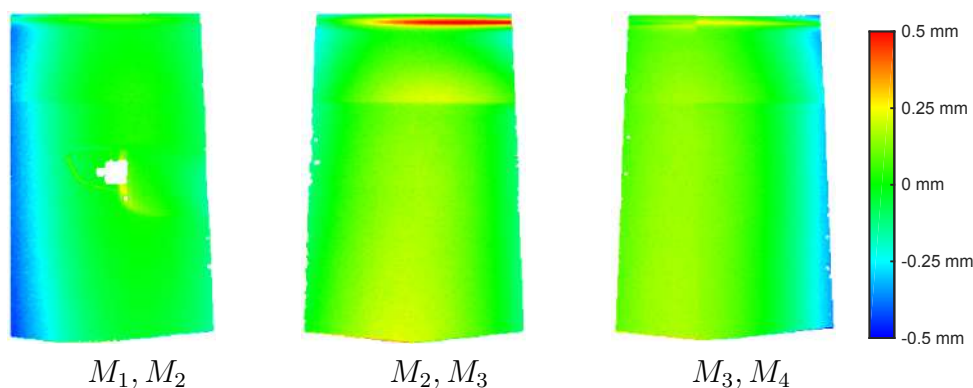


Abbildung 5.21: Defektkarten der Überlappungsbereiche zweier Teilmessungen nach Grobregistrierung

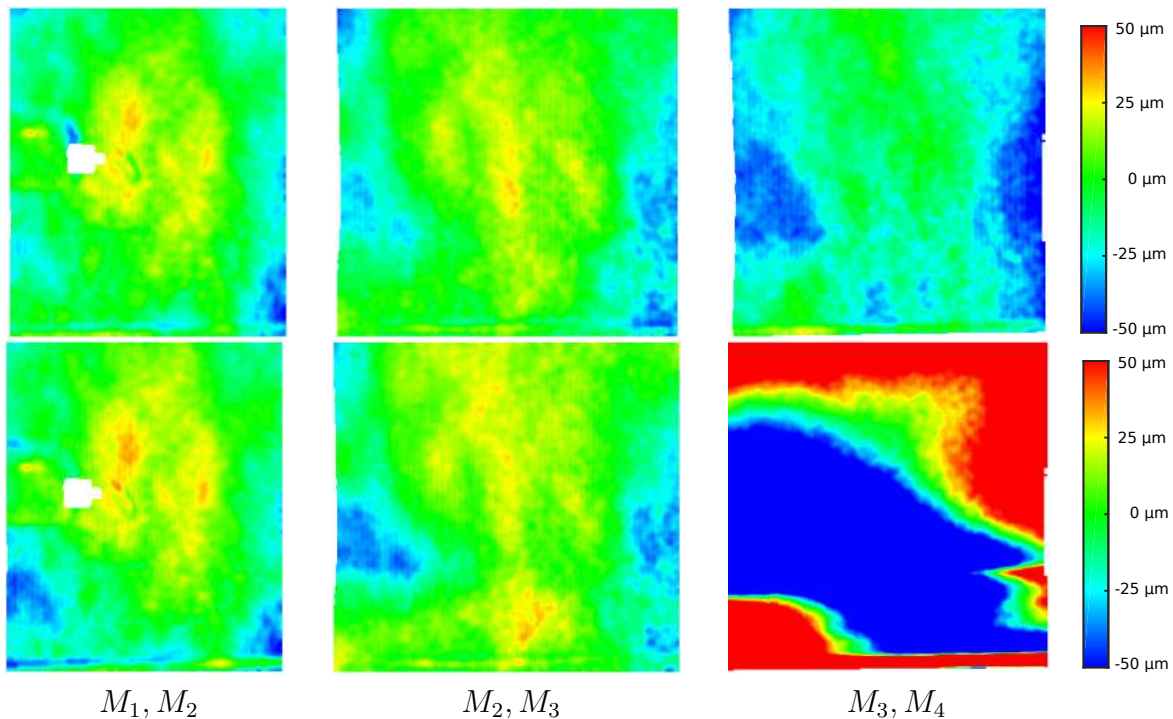
**Grobregistrierung.** Zunächst wurde die zu erwartende Genauigkeit der Grobregistrierung untersucht. Dazu sind in Abbildung 5.21 die Defektkarten dargestellt. Die größten

Flächenbereiche weisen einen Fehler deutlich unter  $\pm 100 \mu\text{m}$  auf. An den Randbereichen steigt der Fehler, liegt aber insgesamt deutlich unter 1 mm.

In Tabelle 5.5 sind die Fehler quantitativ dargestellt. Zunächst fällt auf, dass in den Teilmessungen 1 und 2 die meisten Keypoints gefunden wurden. Beide Messungen enthalten Teile der Türgriffmulde und somit ausgeprägte Merkmale in der Oberflächenform, was sich auch auf die Textur auswirkt. Die weiteren Teilmessungen enthalten deutlich weniger Formmerkmale, sodass die Texturmerkmale entscheidend sind. Mit knapp über 20 eindeutige Keypoints konnten bereits gute Ergebnisse in der Grobregistrierung erzielt werden, mit einer Maximalen Abweichung im Überlappungsbereich von ca. 0.7 mm.

**Tabelle 5.6:** Ergebnisse der Feinregistrierung. Punktweiser Fehler im Überlappungsbereich (Minimum-, Maximum- und mittlerer quadratischer Fehler) für die Spline-basierte Registrierung (SplineReg) sowie die PCL-Implementierung von ICP (PCL-ICP)

Mess.	Fehler SplineReg [ $\mu\text{m}$ ]			Fehler PCL-ICP [ $\mu\text{m}$ ]		
	Min	Max	MSE	Min	Max	MSE
$M_1, M_2$	-57.7	32.1	14.2	-58.7	38.7	15.9
$M_2, M_3$	-46.0	31.2	14.7	-43.3	34.3	15.8
$M_3, M_4$	-127.2	28.1	12.2	-502.8	698.9	163.4



**Abbildung 5.22:** Defektkarten der Überlappungsbereiche zweier Teilmessungen nach Feinregistrierung. **Oben:** Spline-basierte Registrierung, **unten:** PCL-Implementierung von ICP

**Feinregistrierung.** Für die Feinregistrierung wurde als Vergleich der Registrierungsalgorithmus aus der *Point Cloud Library* [67] herangezogen (im folgenden: PCL-ICP). Die Defektkarten der Überlappungsbereiche sind in Abbildung 5.22 dargestellt. Im Vergleich zur Grobregistrierung konnte der Fehler im Überlappungsbereich um eine ganze Größenordnung gesenkt werden (Skalierung der Farbachse bei  $50\ \mu\text{m}$ ). Die Ergebnisse für unsere Methode und die PCL-ICP sind qualitativ ähnlich, was für die korrekte Funktionsweise spricht. Auffällig ist, dass die PCL-ICP für die Messungen  $M_3$  und  $M_4$  ein deutlich schlechteres Registrierungsergebnis erzeugt; das Verfahren ist aufgrund der geringen Merkmalsausprägung nicht konvergiert.

Dies zeigt sich auch in den quantitativen Ergebnissen (Tabelle 5.6) für Messung  $M_3$  und  $M_4$ . Insgesamt können Maximalfehler auch durch die Spline-Registrierung nicht reduziert werden, da sie durch Abweichungen der gemessenen Oberflächen aufgrund systematischer Messfehler nicht exakt übereinstimmen. Der mittlere Fehler konnte jedoch für  $M_1$  und  $M_2$  um 11% (von  $15.9\ \mu\text{m}$  auf  $14.2\ \mu\text{m}$ ) und für  $M_2$  und  $M_3$  um 7% (von  $15.8\ \mu\text{m}$  auf  $14.7\ \mu\text{m}$ ) reduziert werden.

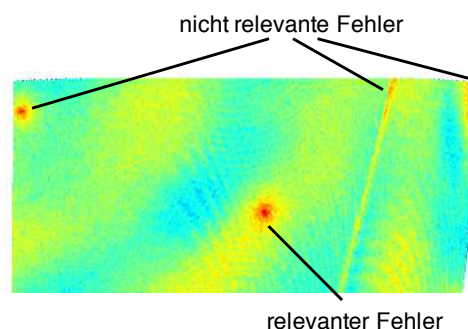
---

## Kapitel 6

# Weiterverarbeitung von Defektkarten

---

Mit den in Abschnitt 4.3 und 5.2 vorgestellten Verbesserungen des Assoziativspeichers kann die Fehlerschwelle des Modellvergleichs effektiv herabgesenkt werden. Dies ermöglicht die Detektion von Oberflächendefekten mit kleinerer Höhe. Dennoch ist nach wie vor keine perfekte Approximation der Oberfläche möglich. Für die zuverlässige Fehlerdetektion ist daher eine Klassifizierung der Fehler notwendig, sodass nicht relevante Modellabweichungen von tatsächlichen Oberflächenfehlern unterschieden werden können (Abbildung 6.1). Weiterhin können Defekte aufgrund ihrer Eigenschaften (Typ und Schwere des Fehlers) bewertet werden – dies ist Ziel weiterer Arbeiten [42, 44].



**Abbildung 6.1:** Defektkarte mit nicht relevanten, systematischen Modellfehlern und relevanten Oberflächendefekten

Abbildung 6.2 zeigt die Verarbeitungskette der Weiterverarbeitung von Defektkarten. Die aus dem Modellvergleich resultierende Defektkarte wird zunächst über eine einfache *Schwellenwert-Binarisierung* in eine Maske möglicher Fehlerbereiche überführt. In Abschnitt 6.1 wird speziell eine Mehr-Schwellen-Segmentierung vorgestellt. Durch Anwendung *morphologischer Operationen* werden anschließend Bereiche geeignet gefiltert. Es werden zusammenhängende Bereiche identifiziert (*Segmentierung*) und *Merkmale* berechnet, die die Fehlerbereiche aussagekräftig beschreiben. Jeder segmentierte Bereich stellt dann einen Punkt in einem hochdimensionalen Merkmalsraum dar. Aufgabe der *Klassifizierung* ist es, Gebieten im Merkmalsraum die Klasse „relevanter Fehler“ bzw. „nicht

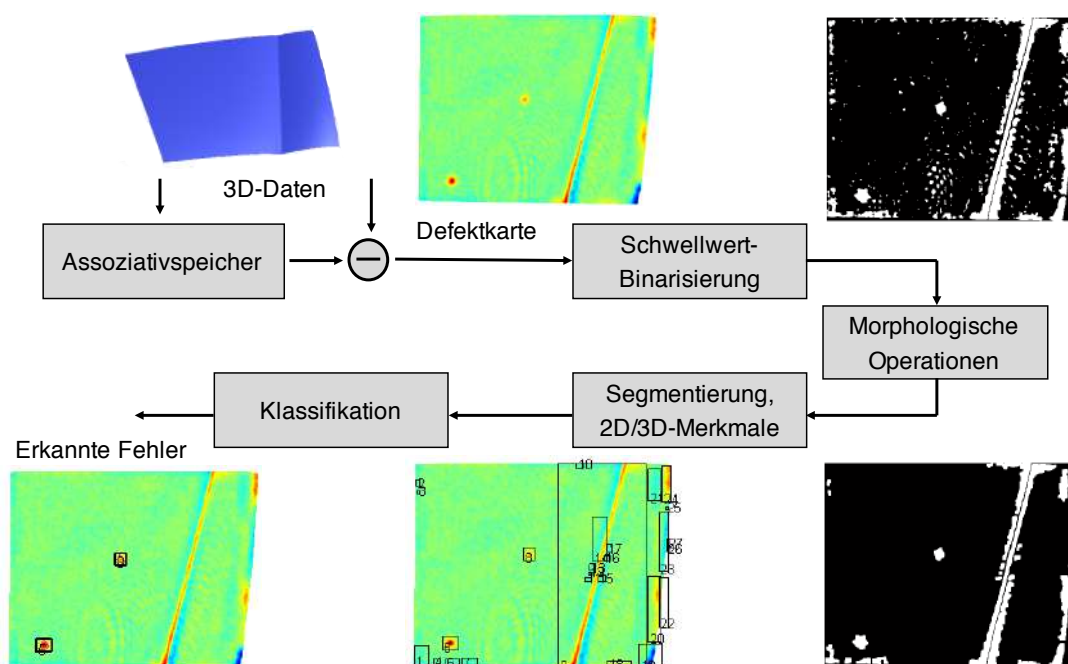


Abbildung 6.2: Überblick zur Weiterverarbeitung von Tiefenkarten

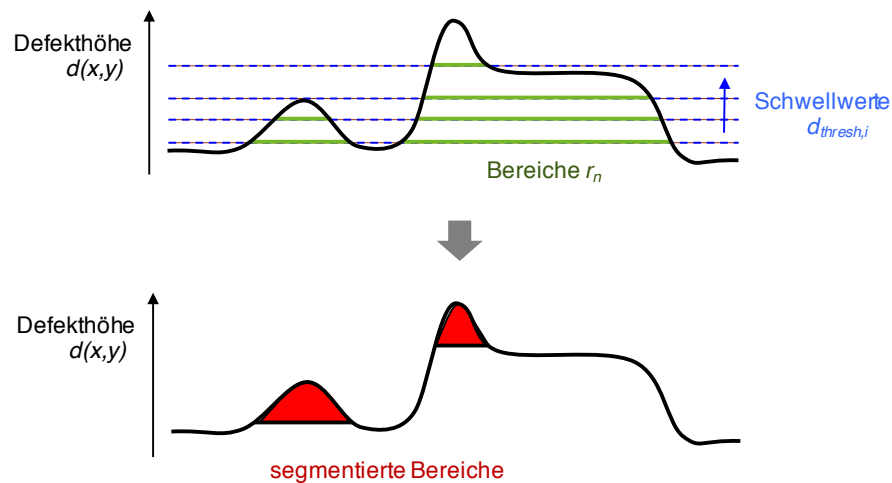
relevanter Fehler“ zuzuweisen. Die auf diese Weise als relevant klassifizierten Bereiche werden als erkannte Fehler zurückgegeben. Die Überlegungen zur Definition geeigneter Merkmale sowie zur Klassifikation ist Abschnitt 6.2 zu entnehmen. Validierungsergebnisse der entwickelten Verfahren werden abschließend in Abschnitt 6.3 dargestellt.

## 6.1 Mehr-Schwellen-Segmentierung

Eine einfache Segmentierung der Defektkarte  $\mathbf{d}(x, y)$  wird durch die Definition eines Schwellenwertes  $d_{thresh}$  für die Defekthöhe erreicht. Es ergibt sich eine Maskierung der Defektkarte:

$$\mathbf{m}_{seg}(x, y) = \mathbf{d}(x, y) > d_{thresh} \quad (6.1)$$

Die Wahl des Schwellenwertes  $d_{thresh}$  ist dabei entscheidend für die korrekte Segmentierung. Ein kleiner Schwellenwert kann notwendig sein um Defekte sehr geringer Höhe zu erkennen, führt aber gleichzeitig zur Übersegmentierung nicht relevanter Bereiche. An Designkanten, an Bereichen hoher Krümmung oder in Überlappungsbereichen zusammengesetzter Flächen treten meist stärkere Modellfehler auf, da diese Bereiche schlechter approximiert werden. In diesen Bereichen muss der Schwellenwert  $d_{thresh}$  höher gewählt werden, um Oberflächendefekte von den systematischen Modellfehlern der Oberfläche unterscheiden zu können.



**Abbildung 6.3:** Visualisierung der Funktionsweise der Mehr-Schwellen-Segmentierung

Der *Mehr-Schwellen-Segmentierung* liegt der Gedanke zugrunde, dass unterschiedliche Bereiche in der Defektkarte jeweils unterschiedliche, angepasste Schwellenwerte  $d_{thresh,1} < d_{thresh,2} < \dots < d_{thresh,\ell}$  erfordern. Zu jeder Schwelle gehört je ein Binärbild  $\mathbf{m}_{seg,i}$ . Zusammenhängende Bereiche bilden über diese Binärbilder hinweg eine Hierarchie. Ein neuer Bereich  $r_n$  auf einer Schwelle  $d_{thresh,i}$  wird dann segmentiert, wenn sein Flächeninhalt sich signifikant zu dem Flächeninhalt des zugehörigen, überlappenden Bereiches der vorherigen Schwelle  $d_{thresh,i-1}$  ändert. Dafür wird ein maximal zulässiger relativer Flächenschwellenwert  $\alpha_{max}$  definiert.

Der Gesamtalgorithmus ist in Tabelle 6.1 dargestellt und kann anhand des Querschnitts einer Defektkarte in Abbildung 6.3 erklärt werden. In dem Höhenprofil (oben) sind Bereiche auf den verschiedenen Schwellen  $d_{thresh,i}$  als grüne horizontale Geraden visualisiert. Im Höhenprofil Abbildung 6.3, unten ist das Resultat der Mehr-Schwellen-Segmentierung dargestellt. Der entwickelte Algorithmus ist angelehnt an die hierarchische Wasserschneidentransformation [16], mit einem zusätzlichen Flächenschwellenwert-Kriterium. Dabei ist ein relativer Flächenschwellenwert  $\alpha_{max} = 0$  gleichbedeutend mit einer Binarisierung auf der ersten Stufe  $d_{thresh,1}$ ; wählt man den Parameter  $\alpha_{max} = 1$ , so wird die Binarisierung auf der Schwelle  $d_{thresh,\ell}$  bevorzugt. Experimentell hat sich ein Wert von  $\alpha_{max} = 0.5$ , sowie  $\ell = 2 \dots 3$  verschiedene Schwellen als sinnvoll erwiesen. Die Schwellenwertparameter  $d_{thresh,i}$  müssen empirisch, abhängig von der Oberflächenform und typischen Defektcharakteristiken gewählt werden.

## 6.2 Merkmale und Klassifikatoren

Für die segmentierten Bereiche können nun verschiedene Merkmale aufgestellt werden, die sich aus den morphologischen Eigenschaften der 2D-Bereiche (Binärbild), aus den

**Tabelle 6.1:** Algorithmus: Mehr-Schwellen-Segmentierung

<p><b>Input</b> : Defektkarte <math>\mathbf{d}(x, y)</math>,                  Anzahl der Schwellenwerte <math>\ell</math>,                  Schwellenwerte der Defekthöhe <math>d_{thresh,1}, \dots, d_{thresh,\ell}</math>,                  relativer Flächenschwellenwert <math>\alpha_{max}</math></p> <p><b>Output:</b> Liste segmentierter Bereiche <math>R^\ell</math></p> <pre style="margin: 0;"> 1 <math>R_0^0 \leftarrow \{(x, y) \mid \forall x, y\}</math> // Initialisiere als Gesamtbild 2 <b>for</b> <math>L = 1 \dots \ell</math> <b>do</b> 3   <math>\mathbf{m}_{seg}(x, y) \leftarrow \begin{cases} 0 &amp; \mathbf{d}(x, y) &lt; d_{thresh,L} \\ 1 &amp; \mathbf{d}(x, y) \geq d_{thresh,L} \end{cases}</math> // Binarisierung 4   <math>\{r_1, r_2, \dots, r_{M_L}\} \leftarrow \text{Bereiche}(\mathbf{m}_{seg})</math> 5   <b>for</b> <math>n = 1 \dots M_L</math> und <math>m = 1 \dots M_{L-1}</math> <b>do</b> // Test auf Überlappung... 6     <b>if</b> <math>r_n \subset R_m^{L-1}</math> <b>then</b> // ...<math>r_n</math> ist Teilmenge 7       <b>if</b> Fläche(<math>r_n</math>)/Fläche(<math>R_m^{L-1}</math>) &lt; <math>\alpha_{max}</math> <b>then</b> 8         Füge <math>r_n</math> zur Bereichsliste <math>R^L</math> hinzu 9       <b>else</b> 10        Füge <math>R_m^{L-1}</math> zur Bereichsliste <math>R^L</math> hinzu 11      <b>end</b> 12    <b>end</b> 13  <b>end</b> 14 <b>end</b> 15 <b>return</b> <math>R^\ell</math>                 </pre>
--

zugehörigen 3D-Daten, aus statistischen oder sonstigen Merkmalen ergeben. Diese bilden einen Merkmalsvektor. Ein einfaches Defektmodell wird durch eine zweidimensionale GAUSS-Funktion gegeben:

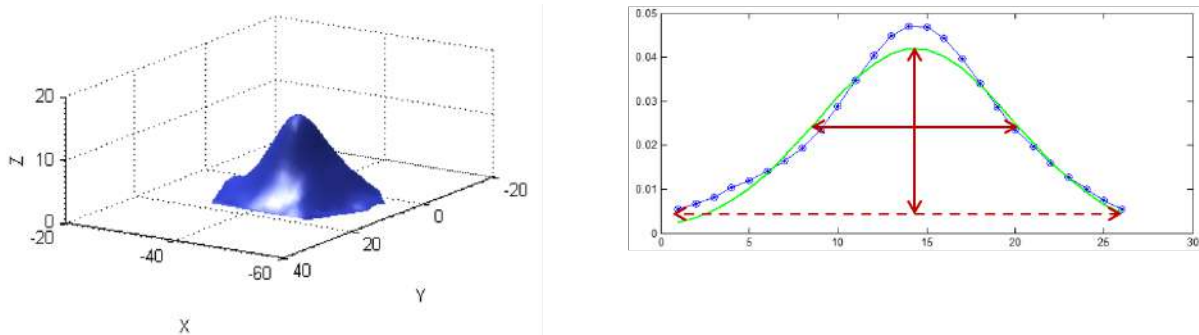
$$e(x, y) = h \cdot \exp\left(-\frac{(x - x_0)^2}{2\sigma_x^2} - \frac{(y - y_0)^2}{2\sigma_y^2}\right) \quad (6.2)$$

Dieses Modell wurde bereits zuvor für die Simulation von Defekten genutzt (vgl. Abbildung 5.7). Es kann an die 3D-Daten der Fehlerbereiche angepasst werden. Als Merkmale können dann die Mittelposition  $(x_0, y_0)$  sowie die Streuungen  $\sigma_x, \sigma_y$  relativ zur Bereichsbreite genutzt werden (siehe Abbildung 6.4). Auch der quadratische Fehler bei der Anpassung des Fehlermodells an die Daten gibt ein aussagekräftiges Merkmal. Insgesamt wurden 18 Merkmale aufgestellt, die in Tabelle 6.2 zusammengefasst sind.



**Tabelle 6.2:** Merkmale zur Charakterisierung von Oberflächendefekten

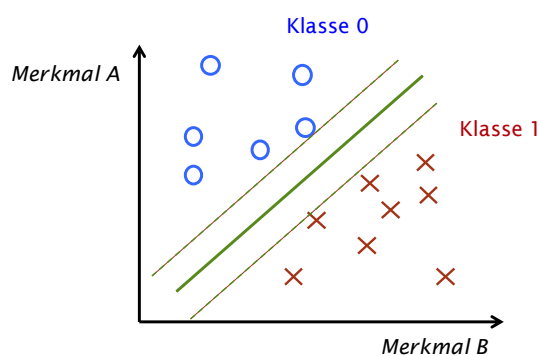
2D-Merkmale	3D-Merkmale
<ul style="list-style-type: none"> <li>• Flächeninhalt</li> <li>• Länge der Haupt- und Nebenachse</li> <li>• Umfang</li> <li>• Circularity, Squareness</li> </ul>	<ul style="list-style-type: none"> <li>• Höhe</li> <li>• relative Mittelposition <math>x,y</math></li> <li>• relative Breite <math>x,y</math></li> <li>• Mittlerer Fitting-Fehler</li> </ul>
Statistische $z$ -Merkmale	Sonstige Merkmale
<ul style="list-style-type: none"> <li>• Absoluter Maximalwert <math>z</math></li> <li>• Mittelwert <math>z</math></li> <li>• Standardabweichung <math>z</math></li> </ul>	<ul style="list-style-type: none"> <li>• Abstand zum Rand</li> <li>• Binarisierungsschwelle</li> </ul>

**Abbildung 6.4:** Darstellung eines Oberflächendefekts und des Defektmodells. **Links:** Beule,  $z$ -Achse 300-fach skaliert, **rechts:** angepasste GAUSS'sches Fehlermodell (grün) mit den Parametern Mittelposition und Streuung bezogen auf die Bereichsbreite (rot) und den zugehörigen 3D-Daten aus der Defektkarte (blau)

Für die Klassifizierung wird ein *überwachtes Lernverfahren* eingesetzt. Dies bedeutet, das für einen Beispielsatz an Oberflächenabweichungen die Merkmalsvektoren und die zugehörige Soll-Klassifikation (relevant/nicht relevant) bekannt sind und für das Training des Klassifikators genutzt werden.

Als Klassifikator wurde eine lineare *Support Vector Machine* (SVM) verwendet; die Funktionsweise ist schematisch in Abbildung 6.5 dargestellt. Aus den Trainingsdaten wird eine lineare Funktion (Ebene bzw. Hyper-Ebene) bestimmt, die einen größtmöglichen Abstandsbereich zwischen den Klassen erzeugt. Die Hyperebene wird dann durch eine Anzahl an Merkmalsvektoren bzw. Stützvektoren (*support vectors*) beschrieben.

Als weiterer Klassifikator wurde eine SVM mit GAUSS'scher radialer Basisfunktion als Kernelfunktion verwendet. Die Kernelfunktion wird genutzt, um den Merkmalsraum in einen höherdimensionalen Raum zu transformieren, in welchem die lineare SVM angewendet wird. Eine lineare Funktion im transformierten Raum führt zu einer nicht-linearen



**Abbildung 6.5:** Funktionsprinzip einer linearen Support Vector Machine zur Trennung zweier Klassen in einem zweidimensionalen Merkmalsraum

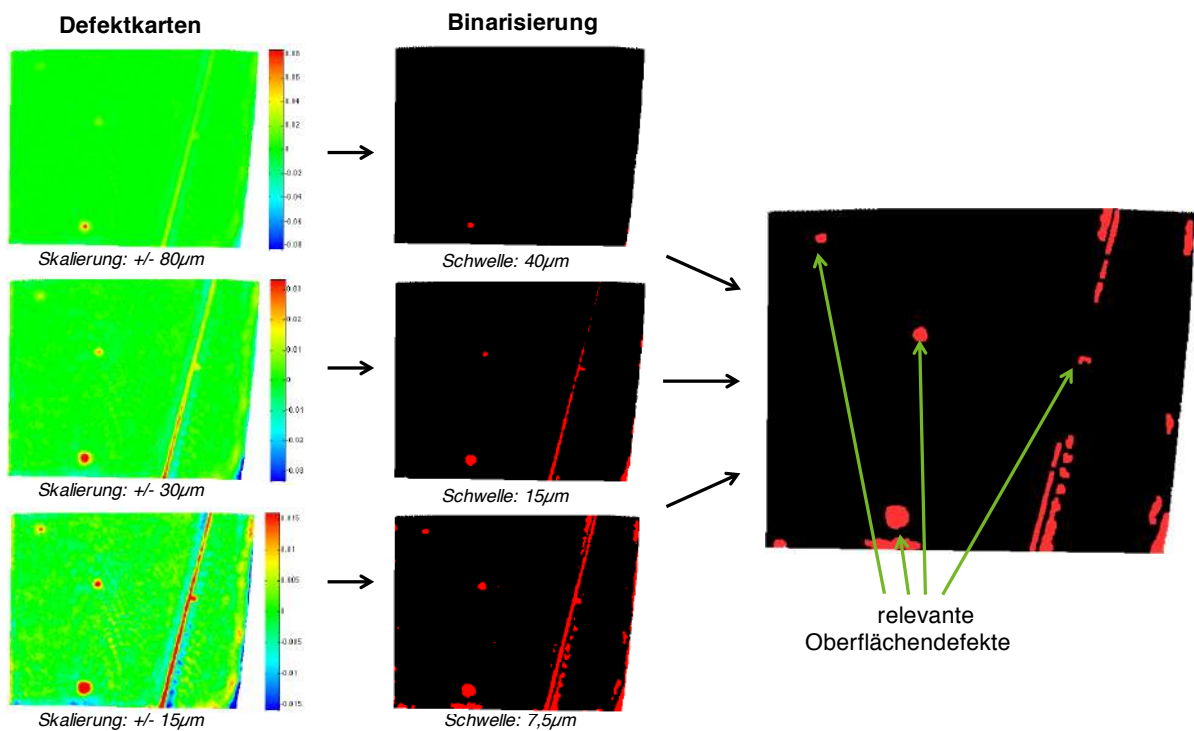
Trennung im ursprünglichen Merkmalsraum, somit lassen sich die Klassengrenzen besser anpassen und Klassen besser voneinander trennen.

Der Merkmalsvektor wird vor Eingabe in den Klassifikator normiert, so dass die einzelnen Merkmale ähnliche Wertebereiche abdecken (z.B. Normierung auf gleiche Minimum-/Maximumwerte). Ein Problem ist die Ungleichverteilung von Klassen. Typischerweise ist es deutlich aufwändiger korrekt gelabelte Positivbeispiele (relevante Fehler) zu erhalten, als Negativbeispiele (nicht-relevante Fehler). Das Training führt zur Bevorzugung der überrepräsentierten Klasse. Um dies zu vermeiden, können Kostenfunktion eingeführt werden, um die Klassen entsprechend ihrer Beispielzahl höher oder niedriger zu gewichten. Eine einfachere Methode ist es, durch Over- bzw. Undersampling der Beispiele eine Klassenbalance herzustellen. Prinzipiell ist ein Lernverfahren bei ungleichverteilten Klassenmengen anfälliger für Überanpassung.

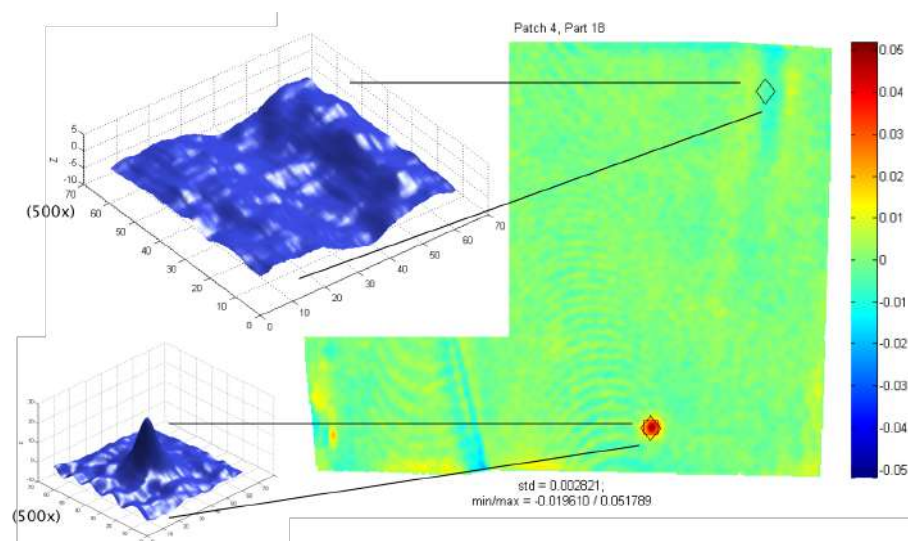
## 6.3 Ergebnisse

Zur Validierung wurde das Teil „Frontklappe“ verwendet. In einem Ringvergleich wurden gezielt 21 Defektteile manuell durch Auditoren bewertet. Dabei wurde eine Datenbasis von 129 gelabelten Oberflächendefekten verschiedener Art und Höhe erzeugt. Für die automatisierte Inspektion wurde ein Assoziativspeicher mit  $N_0 = 20$  Basisfunktionen genutzt.

Abbildung 6.6 zeigt das Qualitative Ergebnis der Mehr-Schwelensegmentierung. Es lässt sich erkennen, dass die untersuchten Defekte bei unterschiedlichen Fehlerschwellen  $d_{thresh}$  verschieden stark in Erscheinung treten. Der Fehler im Bereich der Designkante geht bei höheren Schwellen im Modellfehler unter und ist bei niedrigen Schwellen eindeutig zu segmentieren. Andere Defekte hingegen sind bei niedrigen Schwellen nicht segmentierbar und bedürfen einer höheren Schwelle. Mit der Mehr-Schwelensegmentierung (Abbildung 6.6, rechts) ist die Segmentierung aller vier relevanten Oberflächendefekte möglich.



**Abbildung 6.6:** Qualitatives Ergebnis der Mehr-Schwellen-Segmentierung für einen Ausschnitt der „Frontklappe“. **Links:** Defektkarten mit verschiedener Skalierung der Farbachsen, **Mitte:** Binarisierung auf verschiedenen Schwellen, **Rechts:** Ergebnis der Mehr-Schwellen-Segmentierung mit Markierung der relevanten Oberflächendefekte



**Abbildung 6.7:** Defektkarte für einen Ausschnitt des Teils „Frontklappe“, sowie 3D-Daten zweier durch Auditor bewerteter Fehlerbereiche ( $z$ -Achse 500-fach skaliert). **Defekt oben:** falsch klassifiziert (nicht erkannt); **Defekt unten:** korrekt klassifiziert

**Tabelle 6.3:** Grundklassifikationsgüte (Recall aller Trainingdaten) der linearen SVM und RBF-Kernel SVM

<b>Lineare SVM, Erkennungsrate: 98.91%</b>		
	korrekte Erkennung	inkorrekte Erkennung
relevanter Defekt	126	73
nicht-relevanter Defekt	6669	3

<b>SVM mit RBF Kernel, Erkennungsrate: 99.96%</b>		
	korrekte Erkennung	inkorrekte Erkennung
relevanter Defekt	126	0
nicht-relevanter Defekt	6742	3

**Tabelle 6.4:** Ergebnisse der Klassifikation nach 10-facher Kreuzvalidierung

	lineare SVM	RBF-Kernel SVM
Erkennungsrate	97.60%	99.00%
positiver Vorhersagewert	55.45%	98.17%
negativer Vorhersagewert	99.93%	99.02%

Um die grundsätzlichen Eigenschaften der Klassifikatoren und der Trainingsdaten einzuschätzen, wurden eine lineare SVM sowie eine RBF Kernel SVM mit allen 129 Defekten trainiert und wieder aufgerufen. Die Ergebnisse sind in Tabelle 6.3 dargestellt. Es zeigt sich zunächst, dass eine ganze Reihe von Modellfehlern und weiteren nicht-relevanten Defekten segmentiert wurden. Eine Klassifikation ist daher für eine korrekte Inspektion zwingend notwendig. Weiterhin sind drei gelabelte Oberflächendefekte nicht erkannt worden. Nach näherer Untersuchung wurden diese Bereiche nicht segmentiert und waren daher nicht für die Klassifikation verfügbar. Es handelt sich entweder um Defekte, die durch das Messprinzip grundsätzlich nicht messbar sind oder ein falsches Labeling der Daten.

Die Klassifikatoren wurden weiterhin mit 10-facher Kreuzvalidierung bewertet (Tabelle 6.4). Für die praktische Anwendung ist vor allem die falsch negative Klassifikation kritisch, da in diesem Fall ein Defekt als nicht-relevant bewertet und nicht entdeckt wird. Hier zeigt die lineare SVM beim negativen Vorhersagewert eine leicht bessere Erkennung. Allerdings zeigt die RBF-Kernel SVM eine deutlich bessere Ergebnis hinsichtlich falsch positiver Erkennung und somit eine insgesamt bessere Erkennungsrate von 99.0%.

---

## Kapitel 7

# Zusammenfassung und Ausblick

---

Die vorliegende Arbeit befasst sich mit der Oberflächenforminspektion industriell gefertigter, verformbarer Freiformflächen. Es wird ein modellbasiertes Oberflächeninspektionssystem beschrieben und weiterentwickelt, das auf einem optischen 3D-Messsystem mit Musterprojektion aufbaut. Für die Validierung der neu entwickelten Methoden werden Karosserieoberflächen aus der Automobilindustrie betrachtet.

Bei der Oberflächenforminspektion werden im Gegensatz zur Geometrieprüfung die lokalen, qualitativen Eigenschaften einer Oberfläche untersucht. Ziel ist die Erkennung von Oberflächenunvollkommenheiten, also Defekten wie Beulen oder Dellen, die die optische Wahrnehmung der Oberfläche negativ beeinflussen. Diese stellen sich als Abweichung der gemessenen Oberfläche von einer gedachten Bezugsoberfläche dar. In der Fertigung von Karosserieteilen finden derzeit punktuelle Stichprobenprüfungen statt, die zum größten Teil manuell von Auditoren durchgeführt werden. Insbesondere auf dem unbearbeiteten Blech nach dem Pressen ist eine automatisierte Einschätzung nicht ohne Weiteres möglich.

Die Messwertaufnahme des Systems basiert auf einem Stereo-Kamerasystem mit Streifenlichtprojektion. Durch Projektion einer Bildserie von Streifenmustern nach dem Gray-Code- und Phase-Shift-Verfahren können übereinstimmende Bildpunkte in zwei kalibrierten Kameras im Subpixelbereich gefunden werden. Durch Triangulation ergeben sich hochaufgelöste 3D-Koordinaten. Die resultierende 3D-Oberflächeninformation muss geeignet abgebildet werden. Dazu werden neben den verbreiteten Punktwolken oder Tiefenkarten in dieser Arbeit B-Spline-Oberflächen vorgeschlagen.

Die Auswertung der gemessenen Oberflächengeometrie geschieht im vorgestellten Ansatz modellbasiert. Dazu werden auto-assoziative neuronale Netze oder Assoziativspeicher als datengetriebenes Oberflächenmodell betrachtet. Sie lernen anhand von Beispielmessungen die durchschnittliche Oberflächenform und typische systematische Abweichungen davon. Auf diese Weise können Toleranzbereiche und Verformungen der realen, gefertigten Oberfläche modelliert werden. Für den Assoziativspeicher wurde ein effizienteres Berechnungsverfahren vorgestellt, das auf der dualen Eigenwertzerlegung der Kovarianzmatrix

basiert. Es wurde gezeigt, dass sich die realen Rechenzeiten für das Modelltraining so um über 90% reduzieren. Dies erlaubt die Anwendung des Modells für sehr große Punktwolken.

Für die Erkennung sehr kleiner Oberflächenfehler mit einer Höhe von wenigen  $10\ \mu\text{m}$  ist die Qualität des Modells ausschlaggebend, das Bauteilvarianzen von einigen Millimetern abbilden soll. Die Erhöhung der Approximationsgüte und somit der Modellgenauigkeit ist entscheidend für die erfolgreiche Fehlererkennung. Mit einem hierarchischen Ansatz des Assoziativspeichers kann der maximale Approximationsfehler um rund 24% reduziert werden. Dies betrifft vor allem komplex geformte Oberflächenbereiche. Der Ansatz basiert auf der Erweiterung des Assoziativspeichers nach dem Multi-Skalen-Prinzip in einen Orts-Varianz-Raum. Der dafür entwickelte Trainingsalgorithmus findet Bereiche hoher Varianz und erhöht lokal die Modellvariabilität.

Neben der Modellqualität kann auch der Modellvergleich verbessert werden. B-Spline-Oberflächendarstellungen ermöglichen einen Vergleich in lokalen, oberflächenbezogenen Koordinaten. Es wurden neue Verfahren zur Lösung von Teilproblemen entwickelt, die sich bei ihrer Anwendung in der Oberflächeninspektion ergeben. Dies ist zum einen ein Ansatz zur Vorverarbeitung von Tiefenkarten, mit dem Löcher und Randbereiche aufgefüllt werden können. Dies ist Voraussetzung für die Anwendung des effizienten Lofting-Ansatzes für die Flächenapproximation. Weiterhin wurde ein neues Verfahren zur Bestimmung optimaler Knotenvektoren dargestellt.

Um nun die Vorteile von B-Spline-Modellen und Assoziativspeichern zu kombinieren, wurde ein spline-basierter Assoziativspeicher entwickelt. Dazu wurde die Karhunen-Löève-Transformation für B-Spline-Basen hergeleitet und umgesetzt. Der resultierende spline-basierte Assoziativspeicher führt aufgrund der Abstandsmessung im oberflächenbezogenen Koordinatensystem zu weniger Modellfehlern in starken Krümmungsbereichen. Dies äußert sich in einer qualitativ besseren Defektkarte nach Modellvergleich und erleichtert somit die anschließende Segmentierung und Klassifikation.

Großflächige Bauteile können bei definierten Genauigkeitsanforderungen aufgrund des begrenzten Messfeldes optischer Messsysteme nicht mit einer einzelnen Messung erfasst werden. Das vorgestellte Registrierungsverfahren kann Teilmessungen auch für merkmalsarme Oberflächen kombinieren und ermöglicht so die Inspektion des gesamten Bauteils. Die Grobregistrierung nutzt Texturmerkmale aus den 2D-Kamerabildern und findet auch bei schwach ausgeprägten Texturen und Geometrie eine robuste Schätzung für die Transformation der Teilmessungen. Die spline-basierte Feinregistrierung ermöglicht eine präzise Ausrichtung durch Minimierung des Fehlers im Überlappungsbereich. Am Beispiel einer Autotür wurde gezeigt, dass auch merkmalsarme Oberflächenbereiche mit mittleren Fehlern von  $12\text{...}15\ \mu\text{m}$  registriert werden können, die im Vergleich zum ICP-Verfahren nicht zur Konvergenz führten. Mit einem neuen Blending-Ansatz können letztendlich Punktwolken ohne lokale Verwerfungen oder Sprünge kombiniert werden.

Abschließend müssen die aus dem Vergleich zwischen Messung und Modell entstandene

---

Defektkarte ausgewertet werden. Aufgrund von Modellfehlern können einige Defekte nicht segmentiert werden. Eine neu entwickelte Mehr-Schwellen-Segmentierung löst dieses Problem. Sie greift den Gedanken der hierarchischen Wasserscheidentransformation auf und erweitert ihn um ein Flächenschwellenwert-Kriterium. Aufgrund der hohen Sensitivität des Systems wird eine Reihe nicht-relevanter Bereiche in der Defektkarte segmentiert. Eine Klassifikation basierend auf Support Vector Machines mit radialen Basisfunktionen als Kernel kann tatsächliche Oberflächendefekte identifizieren. Für einen Datensatz einer Kfz-Frontklappe konnte eine Defekt-Erkennungsrate von 99.0% nachgewiesen werden.

**Ausblick.** In der vorliegenden Arbeit wurde eine Reihe von neuen Verfahren präsentiert, die im Sinne eines *Proof of Concepts* validiert wurden. Die Integration der Teilalgorithmen wirft weitere Fragestellungen auf, die zum großen Teil detaillierte Entwicklungs- bzw. Umsetzungsfragen sind. Dazu gehört vor allem eine Validierung in größerem Maßstab mit weiteren Oberflächen.

Ein Problem für die vorgestellte B-Spline-Flächenapproximation stellen Oberflächenbereiche dar, die nicht flächig ausgeprägt sind wie z. B. Stege oder dünne Verbindungsstellen. Da als Grundlage für die Spline-Approximation rechteckige  $x$ - $y$ -Bereiche angenommen werden, kommt es zu sehr dünn besetzten Tiefenkarten, die mit der beschriebenen Methode nicht effizient als B-Spline-Fläche approximierbar sind. Eine „freie“ Anpassung von B-Spline-Flächen ist zwar möglich, jedoch basiert die anschließende Auswertung wiederum auf rechteckigen  $x$ - $y$ -Bereichen. Die Untersuchung von oberflächenbezogenen Defektkarten und ihre Auswertung bietet daher Potential für weitere Arbeiten.

Es existieren zahlreiche B-Spline-Varianten, die in dieser Arbeit nicht betrachtet werden. NURBS bieten potentiell eine weitere Verbesserung der Modellqualität durch Einbringung eines rationalen Terms in die analytische Beschreibung. Dies erlaubt theoretisch eine bessere Abbildung einiger geometrischer Formen; in der praktischen Anwendung kann der Approximationsfehler durch Erweiterung des Knotenvektors jedoch auf sehr einfache Weise hinreichend gesenkt werden. Die nichtlineare mathematische Beschreibung erschwert hingegen die Umsetzung von erweiterten NURBS-Verfahren (z.B. Anwendung des Assoziativspeichers). Ein Problem von Tensorprodukt-B-Spline-Flächen bei ihrer Anwendung auf Oberflächen aus Tiefenkarten sind die gleichbleibenden Knotenvektoren in  $u$ - und  $v$ -Richtung. Ein Oberflächendetail in der Mitte einer glatten Fläche führt zu einer lokal hohen Kontrollpunktdichte in  $u$  und  $v$ , auch wenn dies auf dem Großteil der Fläche nicht notwendig ist. Hierarchische B-Splines oder T-Splines sind interessante Ansätze für weitere Untersuchungen.

Die vorgestellte Fehlerklassifikation dient der Verbesserung der Erkennungsleistung des Inspektionssystems. Eine erweiterte Betrachtung zur Klassifikation hinsichtlich Art und Schwere von Defekten zeigt ebenfalls vielversprechende Ergebnisse [42, 44]. Es existiert eine große Menge an Klassifikationsverfahren aus dem Machine Learning, die getestet werden können. Das Problem ist hier jedoch die – im Vergleich zu verbreiteten Klassifikationsaufgaben aus der Bildverarbeitung – extrem kleine Datenbasis (wenige hunderte

Beispiele) und geringe Datenqualität. Gelabelte Referenzdaten ergeben sich aus der manuellen Prüfung. Die Erweiterung dieser Prüfung um eine gezielte geometrische Lokalisierung und Vermessung sowie Dokumentation für den Aufbau einer Trainingsdatenbasis ist aufwändig und fehleranfällig.



---

# Abbildungsverzeichnis

---

1.1	Reflexionsmuster einer Delle an einer Kfz-Hecktür . . . . .	3
1.2	3D-Messung einer Türgriffmulde und Querschnitt mit Defekt innerhalb des Toleranzbereiches . . . . .	4
2.1	Zusammenfassung der Oberflächenbegriffe nach DIN 4760 und ISO 8785 . . . . .	7
2.2	Arten von Gestaltabweichungen einer Istoberfläche . . . . .	8
2.3	Überblick verschiedener Prüfaufgaben und Einordnung der Oberflächeninspektion . . . . .	9
2.4	Beispiele für Oberflächendefekte . . . . .	10
2.5	Prüfaufgaben der Oberflächeninspektion in der Produktentstehung . . . . .	10
2.6	Mechanisches Tiefziehen eines Blechformteils . . . . .	11
2.7	Relative Kosten für Nacharbeiten in verschiedenen Stadien der Produktion . . . . .	13
2.8	Optisch-zerstörungsfreie manuelle Inspektion in einem Lichttunnel . . . . .	15
2.9	Optisch-zerstörende manuelle Inspektion mit einem Abziehstein . . . . .	15
2.10	Kratzererkennung durch Beobachtung der Oberflächentextur bei verschiedenen Beleuchtungsrichtungen . . . . .	17
2.11	Prinzipieller Aufbau eines Deflektometriesensors . . . . .	18
3.1	Schematische Darstellung einer klassischen 2D-Bildverarbeitungskette . . . . .	24
3.2	Schematische Darstellung der 3D-Datenverarbeitungskette für ein System zur Defekterkennung . . . . .	25
3.3	Prinzip der Triangulation mit zwei Messkameras und Musterprojektion . . . . .	27
3.4	Welt-, Kamera- und Bildkoordinatensystem zur Beschreibung der Kalibrierung . . . . .	28
3.5	Zentralprojektive Abbildung . . . . .	29
3.6	Messvolumen bei Verwendung zweier Kameras . . . . .	31
3.7	Beispiele für strukturierte Muster . . . . .	33
3.8	Musterprojektion beim Phasenshiftverfahren und resultierende Phase . . . . .	35
3.9	Triangulation eines Objektpunktes aus zwei Kameras im Stereonormalfall . . . . .	36
3.10	Rektifizierung von Kamerabildern zur Vereinfachung der Korrespondenzsuche . . . . .	38
3.11	Nutzung der Epipolargeometrie für die bildbasierte Punktsuche . . . . .	38

---

3.12	Objektbasierte Suche in beiden Kamerabildern für das Phase-Shift-Verfahren	38
3.13	Visualisierung von 3D-Punkten, Vertices und Polygonen eines Polygonnetzes mit Vierecksflächen . . . . .	40
3.14	Beispiel für baryzentrische Koordinaten . . . . .	40
3.15	Rendering und Falschfarbendarstellung der Tiefenkarte einer dreidimensional vermessenen Oberfläche . . . . .	42
3.16	Oberfläche mit isoparametrischen Linien . . . . .	45
3.17	Begleitendes Dreibein nach FRENET . . . . .	46
3.18	Bindefunktionen einfacher kubischer Splines . . . . .	49
3.19	Kubische BÉZIER-Kurve und ihre umschreibenden Kontrollpunkte . . . . .	50
3.20	Prinzip „Blossoming“ an drei Kontrollpunkten . . . . .	50
3.21	Bindefunktionen von kubischen BÉZIER-Kurven . . . . .	51
3.22	Quadratischer B-Spline . . . . .	52
3.23	Visualisierung des DE BOOR-Algorithmus für $n = \kappa = 4$ . . . . .	53
3.24	Beispiel für B-Spline-Fläche und zugehörige Kontrollpunkte . . . . .	54
3.25	2D-Basisfunktionen auf äquidistanten Einheits-Knotenvektoren für kubische Splines . . . . .	55
3.26	Vergleich quadratischer B-Spline-Basisfunktionen auf äquidistantem Knotenvektor und mit dichterem Knotenvektor . . . . .	55
3.27	Anwendung des Assoziativspeichers als variables Oberflächenmodell für die Oberflächeninspektion . . . . .	59
3.28	Funktionsweise eines Auto-Assoziativspeichers zur Rekonstruktion am Beispiel einer 2D-Kurve . . . . .	60
3.29	Abstandsberechnung zwischen zwei Punktwolken . . . . .	65
3.30	Abstandsberechnung zwischen zwei Tiefenkarten . . . . .	65
3.31	Abstandsberechnung zwischen parametrischer Fläche und Punktwolke . . . . .	65
3.32	Anwendung eines Tiefpassfilters auf eine Tiefenkarte . . . . .	65
3.33	Anwendung der Dilation auf eine Tiefenkarte . . . . .	66
4.1	Topologie eines Auto-Assoziativen künstlichen neuronalen Netzes . . . . .	69
4.2	Beispiel einer Hauptachsentransformation . . . . .	70
4.3	Zusammenhang zwischen (reduzierter) Eigenwertzerlegung EVD, dualer Eigenwertzerlegung DEVD und Singulärwertzerlegung SVD . . . . .	76
4.4	Gesamtrechenzeit für SVD und DEVD-Ansatz . . . . .	78
4.5	Pyramidale Zerlegung von Eingangsdaten durch iterative Dezimierung und Interpolation . . . . .	79
4.6	Prinzip der Multiskalen-Unterteilung . . . . .	80
4.7	3D-Visualisierungen der untersuchten Oberflächen „Türgriffmulde“ und „Frontklappe“ . . . . .	85
4.8	Ergebnis der automatischen Multiskalen-Unterteilung . . . . .	86
4.9	Ergebnisse der Multiskalen-Unterteilung für verschiedene Algorithmusparameter . . . . .	87

---

4.10	Defektkarte „Türgriffmulde“ mit simulierter Delle nach Anwendung des Assoziativspeichers . . . . .	89
4.11	Defektkarte „Türgriffmulde“ mit zwei simulierten Dellen nach Anwendung einer Tiefpassfilterung . . . . .	89
4.12	Defektkarte „Frontklappe“ mit echten Oberflächendefekten nach Anwendung des Assoziativspeichers . . . . .	89
5.1	Wahl des Parameters $u_k$ als Lotfußpunkt von $\mathbf{x}_k$ auf $\mathbf{x}_c$ . . . . .	94
5.2	Algorithmus zum Füllen von Löchern in Tiefenkarten . . . . .	99
5.3	Iterationsschritte des Algorithmus zur Bestimmung optimaler Knotenvektoren . . . . .	100
5.4	Messung in einem globalen Sensorkoordinatensystem und in einem angepassten lokalen Koordinatensystem . . . . .	102
5.5	Überblick der verschiedenen Oberflächenrepräsentationen für Training und Abruf . . . . .	105
5.6	Vergleich der Defektkarten für ein untrainiertes, fehlerfreies Teil für den herkömmlichen und spline-basierten Assoziativspeicher . . . . .	106
5.7	Simulation eines gaußförmigen Defekts auf eine Objekt Oberfläche. Die Defekthöhe liegt üblicherweise einige Größenordnungen unter der lateralen Defektausdehnung . . . . .	106
5.8	Defektkarten „Türgriffmulde“ mit simulierter Delle für den spline-basierten Assoziativspeicher . . . . .	107
5.9	Relativer Fehler für Defekthöhe einer simulierten gauß'schen Beule bei verschiedenen Defektgrößen und -positionen . . . . .	108
5.10	Visualisierung von 5 teilweise überlappenden Messungen einer Autotür . . . . .	109
5.11	Prinzip des Iterative Closest Point Algorithmus . . . . .	110
5.12	Linke und rechte Ansicht des Stereo-Kamera-Paars für einen Ausschnitt der Autotür . . . . .	111
5.13	Verarbeitungskette und Teilalgorithmen . . . . .	112
5.14	Wichtige Größen am Grauwertverlauf eines Bildpixels für das Phase-Shift-Verfahren . . . . .	114
5.15	Texturextraktion aus dem Hellbild . . . . .	115
5.16	Resultierendes Texturbild mit einzelnen Texturdetails . . . . .	116
5.17	Beispiele für erfolgreich gematchte Texturblöcke . . . . .	117
5.18	Keypoint-Bereich in Kamerabildern, Texturbildern und als Ausschnitt der 3D-Punktwolke . . . . .	117
5.19	Blending-Gerade zur Verrechnung von Ausgangs- und Zielfläche . . . . .	121
5.20	Gewichtsfunktion für das Blending von Tiefenkarten . . . . .	122
5.21	Defektkarten der Überlappungsbereiche zweier Teilmessungen nach Grobregistrierung . . . . .	122
5.22	Defektkarten der Überlappungsbereiche zweier Teilmessungen nach Feinregistrierung . . . . .	123

6.1	Defektkarte mit nicht relevanten, systematischen Modellfehlern und relevanten Oberflächendefekten . . . . .	125
6.2	Überblick zur Weiterverarbeitung von Tiefenkarten . . . . .	126
6.3	Visualisierung der Funktionsweise der Mehr-Schwellen-Segmentierung . .	127
6.4	Darstellung eines Oberflächendefekts und des Defektmodells . . . . .	129
6.5	Funktionsprinzip einer linearen Support Vector Machine . . . . .	130
6.6	Qualiatives Ergebnis der Mehr-Schwellen-Segmentierung für das Teil „Frontklappe“ . . . . .	131
6.7	Defektkarte für das Teil „Frontklappe“ und 3D-Daten zweier Fehlerbereiche	131

---

# Tabellenverzeichnis

---

3.1	Dreiecksschema für kubische BÉZIER-Kurve . . . . .	51
3.2	DE BOOR-Algorithmus zur Konstruktion von B-Splines . . . . .	53
4.1	Rechenzeiten für die Methoden EVD, SVD und DEVD . . . . .	77
4.2	Algorithmus: Training eines hierarchischen Oberflächenmodells mit gegebenener Multiskalen-Unterteilung . . . . .	82
4.3	Algorithmus: Abruf eines trainierten hierarchischen Oberflächenmodells . . . . .	82
4.4	Algorithmus: Training eines hierarchischen Oberflächenmodells mit automatischer Wahl der Anzahl an Eigenvektoren und der Multiskalen-Unterteilung . . . . .	84
4.5	Spitzenwerte und Standardabweichung der Defektkarten für verschiedene Assoziativspeicher-Ansätze . . . . .	86
5.1	Algorithmus: Füllen von Löchern in Tiefenkarten . . . . .	98
5.2	Algorithmus: Bestimmung optimaler Knotenvektoren . . . . .	101
5.3	Algorithmus: RANSAC-Verfahren zum robusten Finden der Transformation für Punktpaare . . . . .	119
5.4	Algorithmus: Spline-basiertes Iterative Closest Point Verfahren . . . . .	120
5.5	Ergebnisse der Grobregistrierung . . . . .	122
5.6	Ergebnisse der Feinregistrierung . . . . .	123
6.1	Algorithmus: Mehr-Schwellen-Segmentierung . . . . .	128
6.2	Merkmale zur Charakterisierung von Oberflächendefekten . . . . .	129
6.3	Grundklassifikationsgüte (Recall aller Trainingdaten) der linearen SVM und RBF-Kernel SVM . . . . .	132
6.4	Ergebnisse der Klassifikation nach 10-facher Kreuzvalidierung . . . . .	132



---

# Veröffentlichungen

---

## Journal-Publikationen

**Sebastian von Enzberg and Ayoub Al-Hamadi:** „A Multiresolution Approach to Model-Based 3D Surface Quality Inspection“. IEEE Transactions on Industrial Informatics 12(4), 2016.

**Sebastian von Enzberg, Ayoub Al-Hamadi and Ahmed Ghoneim:** „Registration of Feature-Poor 3D Measurements from Fringe Projection“. Sensors 16(3), 2016.

## Konferenzartikel

**Sebastian von Enzberg and Ayoub Al-Hamadi:** „Improvement of Data-Driven 3-D Surface Quality Inspection by Deformation Simulation“. IEEE International Conference on Signal and Image Processing Applications, 2019 (submitted).

**Felix Reinhart, Sebastian von Enzberg, Arno Kühn and Roman Dumitrescu:** „Machine Learning for Process-X: A Taxonomy“. Machine Learning for Cyber Physical Systems, Technologies for Intelligent Automation, Vol 11., Springer, 2019.

**Robert Joppen, Sebastian von Enzberg, Arno Kühn and Roman Dumitrescu:** „Data Map – Method for the Specification of Data Flows“. CIRP Conference on Intelligent Computation in Manufacturing Engineering, 2018.

**Arno Kühn, Robert Joppen, Felix Reinhart, Daniel Röltgen, Sebastian von Enzberg and Roman Dumitrescu:** „Analytics Canvas – A Framework for the Design and Specification of Data Analytics Projects“. CIRP Design Conference, 2018.

**Sebastian von Enzberg and Ayoub Al-Hamadi:** „A defect recognition system for automated inspection of non-rigid surfaces“. International Conference on Pattern Recognition, 2014.

**Michal Rapczynski, Erik Lilienblum, Sebastian von Enzberg and Ayoub Al-Hamadi:** „Simultaneous Multi-Camera Calibration Based on Phase-Shift Measurements

on Planar Surfaces“. IEEE International Instrumentation and Measurement Technology Conference, 2014.

**Sebastian von Enzberg and Ayoub Al-Hamadi:** „Variables B-Spline-Modell zur 3D-Oberflächenprüfung“. Proceedings of Workshop 3D-NordOst, 2013.

**Sebastian von Enzberg and Bernd Michaelis:** „Surface Quality Inspection of Deformable Parts with Variable B-Spline Surfaces“. In Proceedings of the joint 34th DAGM and 36th OAGM Symposium, Lecture Notes in Computer Science, Springer, 2012.

**Sebastian von Enzberg, Erik Lilienblum and Bernd Michaelis:** „A physical simulation approach for active photogrammetric 3D measurement systems“. IEEE Instrumentation and Measurement Technology Conference (I2MTC), 2011.

**Sebastian von Enzberg and Bernd Michaelis:** „Approximation of 3D data with piecewise series expansions for surface inspection“. Proceedings of IADIS International Conference CGVCVIP 2011.

## Patentanmeldungen

**Bernd Michaelis, Tilo Lilienblum, Sebastian von Enzberg:** „Method and device for detecting deviations of an object surface“. International Patent Application WIPO PCT, WO 2014 032661 A1, 2014.



---

# Literaturverzeichnis

---

- [1] H. Abdi. A neural network primer. *Journal of Biological Systems*, Jan. 1994.
- [2] G. N. Abenhaim, A. Desrochers, and A. Tahan. Nonrigid parts' specification and inspection methods: notions, challenges, and recent advancements. *International Journal of Advanced Manufacturing Technology*, 63(5-8):741–752, Nov. 2012.
- [3] G. N. Abenhaim, A. S. Tahan, A. Desrochers, and R. Maranzana. A Novel Approach for the Inspection of Flexible Parts Without the Use of Special Fixtures. *Journal of Manufacturing Science and Engineering*, 133(1):011009, Jan. 2011.
- [4] W. Adam and B. Nickolay. Bildverarbeitungsverfahren zur Erkennung von Oberflächenfehlern. *Metalloberfläche*, 43(2):69–72, 1989.
- [5] A. Aidibe, A. Tahan, and G. N. Abenhaim. Dimensioning control of non-rigid parts using the iterative displacement inspection with the maximum normed residual test. *International conference on theoretical and applied mechanics*, Jan. 2011.
- [6] P. Albrecht and B. Michaelis. Stereo photogrammetry with improved spatial resolution. In *Proceedings of the 14th International Conference on Pattern Recognition*, volume 1, pages 845–849. IEEE, 1998.
- [7] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, Aug. 2014.
- [8] L. Armesto, J. Tornero, and A. Herraiez. Inspection system based on artificial vision for paint defects detection on cars bodies. In *ICRA Communications*. IEEE, 2011.
- [9] P. F. Aschwanden. *Experimenteller Vergleich von Korrelationskriterien in der Bildanalyse*. PhD thesis, ETH Zürich, 1993.
- [10] G. Aumann and K. Spitzmüller. *Computerorientierte Geometrie*. BI Wissenschaftsverlag, 1993.
- [11] K.-H. Bae and D. D. Lichti. A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1):36–54, Jan. 2008.

- [12] N. Bauer. *Leitfaden zur Wärmefluss-Thermographie*. zerstörungsfreie Prüfung mit Bildverarbeitung. Fraunhofer-Allianz Vision, 2005.
- [13] R. C. Beach. *An introduction to the curves and surfaces of computer-aided design*. Van Nostrand Reinhold Co., 1991.
- [14] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.
- [15] P. J. Besl and N. D. McKay. IEEE Transactions on pattern analysis and machine intelligence. *Robotics-DL tentative*, 14(2), 1992.
- [16] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In J. Serra and P. Soille, editors, *Mathematical Morphology and Its Applications to Image Processing*, pages 69–76, Dordrecht, 1994. Springer Netherlands.
- [17] A. Birkert, S. Haage, and M. Straub. *Umformtechnische Herstellung komplexer Karosserieteile*. Auslegung von Ziehanlagen. Springer-Verlag, Oct. 2013.
- [18] E. Bispo and R. Fisher. Free-Form Surface Matching for Surface Inspection. In *Proceedings of the 6th IMA Conference on the Mathematics of Surfaces*. Clarendon Press, 1994.
- [19] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1):231–243, Jan. 2004.
- [20] F. Boinski. Auslegung von Ziehteilen und Preßwerkzeugen mit elementaren Methoden unter besonderer Berücksichtigung der Rückfederung. In *Ziehteil- und Preßwerkzeugauslegung*. VDI-Verlag, 1996.
- [21] V. Borsu, A. Yogeswaran, and P. Payeur. Automated Surface Deformations Detection and Marking on Automotive Body Panels. In *IEEE Conference on Automation Science and Engineering*, pages 1–6, July 2010.
- [22] T. Bothe, W. Li, C. von Kopylow, and W. P. O. Juptner. High-resolution 3d shape measurement on specular surfaces by fringe reflection. *Proceedings Optical Metrology in Production Engineering*, 5457:5457 – 5457 – 12, 2004.
- [23] M. Botsch, M. Pauly, L. Kobbelt, P. Alliez, B. Lévy, S. Bischoff, and C. Rössl. Geometric modeling based on polygonal meshes video files associated with this course are available from the citation page. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.
- [24] H.-H. Braess, M. Wawzyniak, L. Teske, H. Goßmann, H. Timm, L. Hamm, V. Peitz, W. Pecho, R. Lachmayer, R. Stauber, R. Konorsa, K. W. Thomer, T. Herpel, P.-O. Wagner, E. P. Neukirchner, G. Laukart, and T. Vorberg. Aufbau - Karosseriebauweisen. In *Vieweg Handbuch Kraftfahrzeugtechnik*, pages 497–630. Springer Fachmedien Wiesbaden, Wiesbaden, 2013.

- 
- [25] C. Bräuer-Burchardt, C. Munkelt, M. Heinze, P. Kühmstedt, and G. Notni. Phase Unwrapping in Fringe Projection Systems Using Epipolar Geometry. *Advanced Concepts for Intelligent Vision Systems*, 5259:422–432, 2008.
- [26] G. B. Broggiato, F. Campana, and S. Gerbino. Shape deviation analysis on sheet-metal parts through reverse engineering techniques. In *ADM International Conference*, Bologna, Sept. 2001. ADM International Conference.
- [27] R. J. Campbell and P. J. Flynn. A Survey Of Free-Form Object Representation and Recognition Techniques. *Computer Vision and Image Understanding*, 81(2):166 – 210, Feb. 2001.
- [28] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.
- [29] W.-J. Chung, Y.-C. Chung, and W.-S. Kim. A robust prediction method of surface deflection using stoning simulation and curvature analysis. *steel research international*, 86(8):886–893, Jan. 2015.
- [30] W.-J. Chung, J.-H. Kim, and Y.-C. Chung. An integrated approach for prediction of surface deflection by using stoning simulation and curvature analysis. *AIP Conference Proceedings*, 1567(1):433–438, 2013.
- [31] T. A. Clarke and J. G. Fryer. The development of camera calibration methods and models. *The Photogrammetric Record*, 1998.
- [32] W. Cuypers, N. van Gestel, A. Voet, J. P. Kruth, J. Mingneau, and P. Bleys. Optical measurement techniques for mobile and large-scale dimensional metrology. *Optics and Lasers in Engineering*, 47(3-4):292–300, Mar. 2009.
- [33] J. Danzberg, M. Engeli, P. Hora, S. Schmäzle, and D. Sourlier. Approximation und Flächenrückführung für Probleme aus der Umformtechnik. In *Optimierung der Produkt- und Prozessentwicklung: Beiträge aus dem Zentrum für integrierte Produktionssysteme*, pages 67–78. vdf Hochschulverlag AG, Zürich, 1999.
- [34] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars. *Computational Geometry: Introduction*. Springer, 2008.
- [35] J. W. Demmel, O. A. Marques, B. N. Parlett, and C. Vömel. Performance and accuracy of LAPACK’s symmetric tridiagonal eigensolvers. *SIAM Journal on Scientific Computing*, 30(3):1508–1526, 2008.
- [36] B. Denkena, H. Ahlers, F. Berg, T. Wolf, and H. K. Tönshoff. Fast inspection of larger sized curved surfaces by stripe projection. *CIRP Annals - Manufacturing Technology*, 51(1):499–502, 2002.
- [37] B. Denkena, F. Berg, and W. Acker. Surface Inspection System for Large Sheet Metal Parts. *Advanced Materials Research*, 6-8:559–564, 2005.

- [38] P. Dierckx. *Curve and Surface Fitting with Splines*. Oxford University Press, Oxford, UK, 1995.
- [39] DIN 4760. Gestaltabweichungen, 1982.
- [40] DIN EN ISO 8785. Geometric Product Specification (GPS) - Surface imperfections - Terms, definitions and parameters, 1998.
- [41] C. Doring, A. Eichhorn, and D. Girimonte. Improving surface detection for quality assessment of car body panels. *Mathware & Soft Computing*, 2004.
- [42] C. Doring, A. Eichhorn, and X. Wang. Improved classification of surface defects for quality control of car body panels. *Fuzzy Systems*, 2006.
- [43] A. Eichhorn, D. Girimonte, A. Klose, and R. Kruse. Surface Quality Analysis with Soft Computing. In *Proceedings of the 10th Zittau Fuzzy Colloquium*, pages 292–299, June 2002.
- [44] A. Eichhorn, C. Döring, A. Klose, and R. Kruse. Classification of surface form deviations for Quality Analysis. *Proc. of the European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems*, pages 121–129, 2003.
- [45] S. von Enzberg and A. Al-Hamadi. Variables B-Spline-Modell zur 3D-Oberflächenprüfung. In *3D NordOst*, pages 1–10, Dec. 2013.
- [46] S. von Enzberg and A. Al-Hamadi. A defect recognition system for automated inspection of non-rigid surfaces. In *International Conference on Pattern Recognition*, pages 1812–1816, Aug. 2014.
- [47] S. von Enzberg and A. Al-Hamadi. A Multiresolution Approach to Model-Based 3-D Surface Quality Inspection. *IEEE Transactions on Industrial Informatics*, 12(4):1498–1507, 2016.
- [48] S. von Enzberg and A. Al-Hamadi. Improvement of data-driven 3-d surface quality inspection by deformation simulation. *IEEE International Conference on Signal and Image Processing Applications, 2019 (submitted)*, (submitted).
- [49] S. von Enzberg, A. Al-Hamadi, and A. Ghoneim. Registration of Feature-Poor 3D Measurements from Fringe Projection. *Sensors*, 16(3):283, Mar. 2016.
- [50] S. von Enzberg, E. Lilienblum, and A. Al-Hamadi. A physical simulation approach for active photogrammetric 3D measurement systems. (*IEEE International Instrumentation and Measurement Technology Conference*), Jan. 2011.
- [51] S. von Enzberg and B. Michaelis. Approximation of 3D data with Piecewise Series Expansions for Surface Inspection. In *Proceedings of the International Conference Computer Graphics, Visualization, Computer Vision and Image Processing*, Rome, Italy, July 2011.

- 
- [52] S. von Enzberg and B. Michaelis. Surface Quality Inspection of Deformable Parts with Variable B-Spline Surfaces. In *Lecture Notes in Computer Science: Pattern Recognition*, pages 175–184. Springer Berlin Heidelberg, 2012.
- [53] E. Ezra, M. Sharir, and A. Efrat. On the ICP algorithm. In *Proceedings of the Annual Symposium on Computational Geometry*, pages 95–104, Sedona, Arizona, USA, 2006. New York, NY, USA, ACM.
- [54] G. Farin. *Curves and Surfaces for CAGD - A Practical Guide*. Morgan Kaufmann, 2001.
- [55] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.
- [56] J. G. Fryer and D. C. Brown. Lens distortion for close-range photogrammetry. *Photogrammetric engineering and remote sensing*, 52(1):51–58, 1986.
- [57] E. de la Fuente López and F. M. Trespaderne. Inspection of stamped sheet metal car parts using a multiresolution image fusion technique. *Computer vision systems*, pages 345–353, 2009.
- [58] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Oct. 2013.
- [59] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust Global Registration. In *Eurographics Symposium on Geometry Processing*, pages 1–10, 2005.
- [60] J. J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern Recognition*, 1981.
- [61] M. Germani, F. Mandorli, M. Mengoni, and R. Raffaeli. CAD-based environment to bridge the gap between product design and tolerance control. *Precision Engineering*, 34(1):7–15, Jan. 2010.
- [62] G. Godin, D. Laurendeau, and R. Bergevin. A method for the registration of attributed range images. *3-D Digital Imaging and Modeling*, 2001.
- [63] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2008.
- [64] A. A. Goshtasby. Three-dimensional model construction from multiview range images: survey with new results. *Pattern Recognition*, 31(11):1705–1714, 1998.
- [65] G. Heber. *Mathematische Hilfsmittel der Physik*. Akademie Verlag, Berlin, 1966.
- [66] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106–1112. IEEE, 1997.

- [67] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. Registration with the point cloud library: A modular framework for aligning in 3-d. *IEEE Robotics & Automation Magazine*, 22(4):110–124, 2015.
- [68] B. Jähne. *Digitale Bildverarbeitung*. Springer-Verlag, 6. edition, 2005.
- [69] B. Jähne, R. Massen, B. Nickolay, and H. Scharfenberg. *Technische Bildverarbeitung - Maschinelles Sehen*. Springer Verlag, Aug. 1995.
- [70] A. E. Jaramillo, P. Boulanger, and F. Prieto. On-line 3-D system for the inspection of deformable parts. *International Journal of Advanced Manufacturing Technology*, 57(9-12):1053–1063, Dec. 2011.
- [71] A. E. Johnson and S. B. Kang. Registration and integration of textured 3D data. *Image and vision computing*, 1999.
- [72] S. Kammel and J. Horbach. Topography reconstruction of specular surfaces. *Electronic Imaging 2005*, 2005.
- [73] S. Karbacher, J. Babst, G. Häusler, and X. Laboureaux. Visualization and Detection of Small Defects on Car-Bodies. In *Proceedings of Vision, Modeling and Visualization*, 1999.
- [74] B. Klotzek. *Einführung in die Differentialgeometrie*. Verlag Harri Deutsch, 1997.
- [75] K. R. Koch. Identity of simultaneous estimates of control points and of their estimates by the lofting method for NURBS surface fitting. *International Journal of Advanced Manufacturing Technology*, 44(11-12):1175–1180, Oct. 2009.
- [76] I. Kolingerová. 3D-line clipping algorithms — a comparative study. *The Visual Computer*, 11(2):96–104, Feb. 1994.
- [77] M. Korn, M. Holzkothen, and J. Pauli. Color Supported Generalized-ICP. In *international conference on Computer Vision Theory and Applications (VISAPP)*, 2014.
- [78] M. A. Kramer. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE Journal*, 37(2), 1991.
- [79] T.-T. Le, M. Ziebarth, T. Greiner, and M. Heizmann. Inspection of specular surfaces using optimized M-channel wavelets. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 2405–2409, 2013.
- [80] P. Leinweber. *Taschenbuch der Längenmeßtechnik*. Für Konstruktion / Werkstatt / Meßraum und Kontrolle. Springer-Verlag, Mar. 2013.
- [81] Y. Li and P. Gu. Free-form surface inspection techniques state of the art review. *Computer-Aided Design*, 36(13):1395–1417, Nov. 2004.

- 
- [82] B. Lifshitz and A. Fischer. Verification of scanned engineering parts with CAD models based on discrete curvature estimation. *Shape Modeling Applications*, pages 333–336, 2004.
- [83] E. Lilienblum, S. Handrich, and A. Al-Hamadi. Low cost calibration of stereo line scan camera systems. In *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*, pages 322–325. IEEE, 2017.
- [84] E. Lilienblum and B. Michaelis. Optical 3D Surface Reconstruction by a Multi-Period Phase Shift Method. *Journal of Computers*, 2:73–83, Jan. 2007.
- [85] T. Lilienblum. *Aufbereitung von 3D-Messwerten unter Nutzung künstlicher neuronaler Netze*. PhD thesis, Universität Magdeburg, 1999.
- [86] T. Lilienblum, P. Albrecht, and R. Calow. Dent detection in car bodies. In *International Conference on Pattern Recognition*, pages 775–778, Jan. 2000.
- [87] T. Lilienblum, P. Albrecht, and B. Michaelis. Improvement of 3-D Data by Neural Networks. *Neural Networks for Identification, Control, and Robotics, International Workshop*, 0:0203, 1996.
- [88] H. Loferer. Automatic Painted Surface Inspection and Defect Detection. In *SENSOR+TEST Conferences*, 2011.
- [89] T. Luhmann. *Nahbereichsphotogrammetrie*. VDE Verlag, 2018.
- [90] L. Mackey. Deflation methods for sparse PCA. *Advances in neural information processing systems*, 2009.
- [91] E. N. Malamas, E. Petrakis, M. Zervakis, and L. Petit. A survey on industrial vision systems, applications and tools. *Image and vision computing*, 2003.
- [92] B. Michaelis. *Zusammengesetzte Messgrößen und ihre Anwendung*. PhD thesis, TH Magdeburg, July 1980.
- [93] B. Michaelis, T. Lilienblum, and S. von Enzberg. Method and device for detecting deviations of an object surface. WIPO PCT.
- [94] J. Molleda, R. Usamentiaga, D. F. García, F. G. Bulnes, A. Espina, B. Dieye, and L. N. Smith. An improved 3D imaging system for dimensional quality inspection of rolled products in the metal industry. *Computers in Industry*, 64(9):1186–1200, Dec. 2013.
- [95] N. Neogi, D. K. Mohanta, and P. K. Dutta. Review of vision-based steel surface inspection systems. *EURASIP Journal on Image and Video Processing*, 2014(1):50, 2014.
- [96] T. S. Newman and A. K. Jain. A survey of automated visual inspection. *Computer vision and image understanding*, 61(2):231–262, 1995.

- [97] B. Nickolay and T. Wieland. Texturanalyse an leicht gekrümmten Oberflächen. *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 87(10):600–602, 1992.
- [98] N. J. Nilsson. *The Quest for Artificial Intelligence*. Cambridge University Press, Oct. 2009.
- [99] A. Ohl. *Modellierung von Gestaltmerkmalen für die industrielle Oberflächeninspektion*. PhD thesis, Universität Siegen, 2004.
- [100] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag New York, 2003.
- [101] M. Özkul. *Qualitätsansprüche bezüglich des äußeren Erscheinungsbildes von Automobilen der Premiumklasse*. PhD thesis, TU München, 2009.
- [102] M. Ozkul and T. Becker. Qualitätssichernde Messverfahren zur Detektion von Oberflächenmerkmalen in der Automobilindustrie. *Allgemeine Vermessungs-Nachrichten (AVN)*, 2009.
- [103] H. Park and K. Kim. Smooth surface approximation to serial cross-sections. *Computer-Aided Design*, 28(12):995–1005, 1996.
- [104] H. Park, K. Kim, and S.-C. Lee. A method for approximate nurbs curve compatibility based on multiple curve refitting. *Computer-Aided Design*, 32(4):237–252, 2000.
- [105] J. Peng, V. Strela, and D. Zorin. A simple algorithm for surface denoising. In *Proceedings IEEE Computer Society Conference on Visualization*, pages 107–112, San Diego, California, 2001. Washington, DC, USA, IEEE Computer Society.
- [106] L. Piegl and W. Tiller. Algorithm for approximate nurbs skinning. *Computer-Aided Design*, 28(9):699–706, 1996.
- [107] L. Piegl and W. Tiller. *The NURBS Book*. Springer Science & Business Media, Dec. 2012.
- [108] M. Potmesil. Generating Models of Solid Objects by Matching 3D Surface Segments. In *IJCAI*, pages 1089–1093, 1983.
- [109] K. Qin. General matrix representations for B-splines. *The Visual Computer*, 16(3):177–186, 2000.
- [110] L. Ramshaw. Blossoming: a connect-the-dots approach to splines. *Digital Equipment Corporation*, pages 1–184, July 1998.
- [111] M. Rapczynski, E. Lilienblum, S. von Enzberg, and A. Al-Hamadi. Simultaneous multi-camera calibration based on phase-shift measurements on planar surfaces. In *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, pages 175–180. IEEE, 2014.



- 
- [112] R.-J. Recknagel. *Defekterkennung an Oberflächen mittels Waveletmethoden*. PhD thesis, Universität Jena, 2000.
- [113] A. Redd. A comment on the orthogonalization of B-spline basis functions and their derivatives. *Statistics and Computing*, 22(1):251–257, Jan. 2012.
- [114] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152, 2001.
- [115] V. Sabri. *Dimensional metrology of non-rigid parts without specialized inspection fixtures*. PhD thesis, École de technologie supérieure, 2017.
- [116] J. Salvi, X. Armangué, and J. Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 2002.
- [117] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43(8):2666–2680, Aug. 2010.
- [118] J. Salvi, J. Pagès, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, Apr. 2004.
- [119] G. Sansoni, S. Corini, S. Lazzari, R. Rodella, and F. Docchio. Three-dimensional imaging based on Gray-code light projection: characterization of the measuring algorithm and development of a measuring system for industrial applications. *Applied optics*, 36(19):4463–4472, 1997.
- [120] G. Sansoni, M. Trebeschi, and F. Docchio. State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation. *Sensors*, 9(1):568–601, Jan. 2009.
- [121] Z. Sárosi, W. Knapp, A. Kunz, K. Wegener, A. Kunz, A. Kunz, K. Wegener, and K. Wegener. Detection of surface defects on sheet metal parts by using one-shot deflectometry in the infrared range. *Inframation*, 2010.
- [122] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. *Robotics: Science and Systems*, 2(4), 2009.
- [123] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *Computer Vision and Pattern Recognition*, 1:519–528, 2006.
- [124] Q. Shi, N. Xi, Y. Chen, and W. Sheng. Registration of point clouds for 3D shape inspection. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 235–240, 2006.
- [125] K. Siegert. Tiefziehen. In K. Siegert, editor, *Blechumformung: Verfahren, Werkzeuge und Maschinen*. Springer-Verlag, Berlin Heidelberg, Apr. 2015.

- [126] S. Steeb. *Zerstörungsfreie Werkstück-und Werkstoffprüfung: die gebräuchlichsten Verfahren im Überblick*. Expert-Verlag, 1993.
- [127] B. Thielbeer. Glänzende Ansichten - Reflexionsmesstechnik im Fahrzeugbau. In T. Luhmann and C. Müller, editors, *Photogrammetrie - Laserscanning - Optische 3D-Messtechnik: Beiträge der Oldenburger 3D-Tage 2010*, pages 302–308. Wichmann, 2010.
- [128] J. Tornero, L. Armesto, N. Montés, M. C. Mora, and A. Herráez. Quality Control of Car-Bodies Based on Artificial Vision. In *European Automotive Congress 2011*, Valencia, Spain, June 2011.
- [129] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *Robotics and Automation, IEEE Journal of*, 3(4):323–344, 1987.
- [130] VDI/VDE 2634. Optische 3D-Messsysteme, May 2002.
- [131] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*, volume 87. Prentice Hall PTR, Englewood Cliffs, New Jersey, Feb. 1995.
- [132] J. Weber. *Automotive Development Processes*. Processes for Successful Customer Oriented Vehicle Development. Springer Science & Business Media, June 2009.
- [133] A. Weckenmann. *Koordinatenmesstechnik*. Flexible Meßstrategien für Maß, Form und Lage. Carl Hanser Verlag GmbH Co KG, June 2012.
- [134] A. Weckenmann and J. Weickmann. Optical inspection of formed sheet metal parts applying fringe projection systems and virtual fixation. *Metrology and Measurement Systems*, 13(4):321–334, 2006.
- [135] G. Q. Wei and S. D. Ma. Two plane camera calibration: a unified model. . . . *1991 Proceedings CVPR'91*, 1991.
- [136] G. Q. Wei and S. D. Ma. A complete two-plane camera calibration method and experimental comparisons. *Computer Vision*, 1993.
- [137] A. Wendt and C. Heipke. Simultaneous orientation of brightness, range and intensity images. *ISPRS Archives*, 2006.
- [138] S. B. Werling. *Deflektometrie zur automatischen Sichtprüfung und Rekonstruktion spiegelnder Oberflächen*. PhD thesis, KIT Scientific Publishing, 2011.
- [139] J. Williams and M. Bennamoun. Simultaneous Registration of Multiple Corresponding Point Sets. *Computer vision and image understanding*, 81(1):117–142, Jan. 2001.
- [140] C. Wöhler. 3D computer vision: efficient methods and applications, 2012.

- [141] X. Xie. A review of recent advances in surface defect detection using texture analysis techniques. *Electronic Letters on Computer Vision and Image Analysis*, 7(3):1–22, 2008.
- [142] B. G. P. Y.Y. Hung, L. Lin. Practical 3-d computer vision techniques for full-field surface measurement. *Optical Engineering*, 39:39 – 39 – 7, 2000.
- [143] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jan. 2000.
- [144] Z. H. Zhang. Review of single-shot 3D shape measurement by phase calculation-based fringe projection techniques. *Optics and Lasers in Engineering*, 2012.
- [145] H.-K. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. *IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 194–201, 2001.
- [146] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3D object recognition. *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696, 2009.
- [147] H. Zollner and R. Sablatnig. Comparison of methods for geometric camera calibration using planar calibration targets. In *Proceedings of the 28th Workshop of the Austrian Association for Pattern Recognition*, Jan. 2004.