

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Betriebswissenschaften und Montagetechnik am  
Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*)

### **3D-Image-Stitching für roboterbasierte Messsysteme**

**Marco Ulrich**

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktor-Ingenieurs (Dr.-Ing.)**

genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. Michael Zäh

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Gunther Reinhart
2. Prof. Dr. rer. nat. Dominik Henrich

Die Dissertation wurde am 28.06.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 05.02.2018 angenommen.



## Geleitwort der Herausgeber

Die Produktionstechnik ist für die Weiterentwicklung unserer Industriegesellschaft von zentraler Bedeutung, denn die Leistungsfähigkeit eines Industriebetriebes hängt entscheidend von den eingesetzten Produktionsmitteln, den angewandten Produktionsverfahren und der eingeführten Produktionsorganisation ab. Erst das optimale Zusammenspiel von Mensch, Organisation und Technik erlaubt es, alle Potentiale für den Unternehmenserfolg auszuschöpfen.

Um in dem Spannungsfeld Komplexität, Kosten, Zeit und Qualität bestehen zu können, müssen Produktionsstrukturen ständig neu überdacht und weiterentwickelt werden. Dabei ist es notwendig, die Komplexität von Produkten, Produktionsabläufen und -systemen einerseits zu verringern und andererseits besser zu beherrschen.

Ziel der Forschungsarbeiten des *iwb* ist die ständige Verbesserung von Produktentwicklungs- und Planungssystemen, von Herstellverfahren sowie von Produktionsanlagen, Betriebsorganisation, Produktions- und Arbeitsstrukturen sowie Systeme zur Auftragsabwicklung werden unter besonderer Berücksichtigung mitarbeiterorientierter Anforderungen entwickelt. Die dabei notwendige Steigerung des Automatisierungsgrades darf jedoch nicht zu einer Verfestigung arbeitsteiliger Strukturen führen. Fragen der optimalen Einbindung des Menschen in den Produktentstehungsprozess spielen deshalb eine sehr wichtige Rolle.

Die im Rahmen dieser Buchreihe erscheinenden Bände stammen thematisch aus den Forschungsbereichen des *iwb*. Diese reichen von der Entwicklung von Produktionssystemen über deren Planung bis hin zu den eingesetzten Technologien in den Bereichen Fertigung und Montage. Steuerung und Betrieb von Produktionssystemen, Qualitätssicherung, Verfügbarkeit und Autonomie sind Querschnittsthemen hierfür. In den *iwb* Forschungsberichten werden neue Ergebnisse und Erkenntnisse aus der praxisnahen Forschung des *iwb* veröffentlicht. Diese Buchreihe soll dazu beitragen, den Wissenstransfer zwischen dem Hochschulbereich und dem Anwender in der Praxis zu verbessern.



## **Vorwort**

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Werkzeugmaschinen und Betriebswissenschaften (iwb) der Technischen Universität München.

Herrn Prof. Dr.-Ing. Gunther Reinhart danke ich für das entgegengebrachte Vertrauen und die wohlwollende Betreuung der Dissertation sowie Herrn Prof. Dr.-Ing. Michael Zäh für die Übernahme des Vorsitzes der Prüfungskommission. Weiterhin gilt mein Dank Herrn Prof. Dr. Dominik Henrich, dem Leiter des Lehrstuhls Angewandte Informatik III (Robotik und Eingebettete Systeme) an der Universität Bayreuth, für die Übernahme des Koreferates und die aufmerksame Durchsicht der Arbeit.

Darüber hinaus bedanke ich mich herzlich bei allen Mitarbeiterinnen und Mitarbeitern des Institutes für die angenehme Atmosphäre und gute Zusammenarbeit. Mein spezieller Dank gilt Gregor Lux-Gruenberg und Veit Hammerstingl, die mich bei der Erstellung dieser Arbeit mit Rat und Tat unterstützt, das Manuskript kritisch begutachtet und wertvolle Anregungen gegeben haben.

Weiterhin möchte ich mich bei allen Studierenden bedanken, die mich in der Umsetzung engagiert und zuverlässig unterstützt haben. Besonders hervorheben möchte ich dabei Andreas Forstner, Patrick Haslinger und Sebastian Kutter.

Mein Dank gilt außerdem meiner Familie und meinen Freunden, die immer für mich da waren und mir stets den Rücken gestärkt haben. Besonderen Anteil am Gelingen dieser Arbeit hatte meine Frau Ramona: Wenn es schwierig wurde, motivierte sie mich – wenn ich schwierig wurde, zeigte sie Verständnis.



# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>Verzeichnis der Formelzeichen</b>	<b>VII</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Grundlagen roboterbasierter Messtechnik</b>	<b>5</b>
2.1 Optische 3D-Koordinatenmesstechnik . . . . .	6
2.1.1 Sensoren für die 3D-Bildgewinnung . . . . .	6
2.1.2 Datenverarbeitung und Interpretation . . . . .	8
2.1.3 Registrierung von Punktwolken . . . . .	9
2.2 Vertikal-Knickarmroboter . . . . .	10
2.2.1 Aufbau und Funktionsweise . . . . .	10
2.2.2 Programmierung . . . . .	14
2.2.3 Industrieller Einsatz . . . . .	16
<b>3 Zielstellung und Vorgehensweise</b>	<b>19</b>
3.1 Defizite und Grenzen roboterbasierter Messsysteme . . . . .	19
3.2 Zielstellung und Lösungsansatz . . . . .	21
<b>4 Stand der Forschung und Technik</b>	<b>25</b>
4.1 Datengetriebene Registrierung von Punktwolken . . . . .	25
4.1.1 Registrierung durch bekannte Korrespondenzen . . . . .	26
4.1.2 Der „Iterative Closest Point“-Algorithmus . . . . .	28
4.1.3 Anwendung in der Produktionstechnik . . . . .	35
4.2 Posen- und Bahnplanung von Robotern . . . . .	35
4.2.1 Beschreibung im Konfigurationsraum . . . . .	35
4.2.2 Kollisionsdetektion als Flaschenhals . . . . .	37
4.2.3 Modellierung mittels Graphentheorie . . . . .	38
4.2.4 Stichprobenbasierte Verfahren . . . . .	44

4.2.5	Anwendung in der Fertigungsmesstechnik . . . . .	51
4.3	Fazit und Handlungsbedarf . . . . .	52
<b>5</b>	<b>Registrierung durch Verknüpfung von Punktwolken</b>	<b>55</b>
5.1	Anforderungen und Rahmenbedingungen . . . . .	55
5.2	Konzept zur Registrierung durch Verknüpfung . . . . .	57
5.3	Anreicherung homogener Oberflächen durch Projektion . . . . .	61
5.3.1	Systemtechnik für die Muster-Projektion . . . . .	62
5.3.2	Resultierende Anforderungen . . . . .	63
5.4	Algorithmus zur paarweisen Verknüpfung . . . . .	65
5.4.1	Startwert und Zuschnitt . . . . .	67
5.4.2	Umwandlung der Projektion in räumliche Strukturen . . . . .	68
5.4.3	Ausrichtung mittels ICP-Algorithmus . . . . .	72
5.5	Einfluss von Abbildungsfehlern . . . . .	76
5.5.1	Fehler der paarweisen Verknüpfung . . . . .	76
5.5.2	Fehlerfortpflanzung bei inkrementeller Registrierung . . . . .	77
5.5.3	Verringerung des Messfehlers . . . . .	78
5.6	Zusammenfassung der Methode zur Registrierung . . . . .	80
<b>6</b>	<b>Automatisierte Bahnplanung</b>	<b>81</b>
6.1	Anforderungen und Rahmenbedingungen . . . . .	82
6.2	Konzept zur Programmerstellung . . . . .	83
6.3	Analyse des Prüfbjekts . . . . .	86
6.4	Generierung von Aufnahmeposes . . . . .	87
6.4.1	Anforderungen des Sensors . . . . .	89
6.4.2	Anforderungen des handhabenden Roboters . . . . .	90
6.4.3	Berechnung des Sichtbereichs und externe Projektion . . . . .	91
6.4.4	Ablauf zur Generierung von Aufnahmeposes . . . . .	93
6.5	Erstellung des Registrierungsbaums . . . . .	93
6.5.1	Modellierung der Bahnplanungsaufgabe . . . . .	95
6.5.2	Spannender Baum mit minimalem Messfehler . . . . .	96
6.6	Generierung einer kollisionsfreien Roboterbahn . . . . .	98
6.6.1	Auswahl der Bahnplanungsmethode . . . . .	98
6.6.2	Aufbau der Roadmap . . . . .	100
6.6.3	Bestimmung der optimalen Reihenfolge . . . . .	104
6.6.4	Ablauf zur Generierung der Roboterbahn . . . . .	106

6.7	Anpassung auf den konkreten Anwendungsfall . . . . .	106
6.8	Zusammenfassung der Bahnplanung . . . . .	109
<b>7</b>	<b>Validierung und Evaluation</b>	<b>111</b>
7.1	Prototypische Implementierung . . . . .	111
7.1.1	Mechanischer und elektrischer Aufbau . . . . .	111
7.1.2	Softwaretechnische Realisierung . . . . .	114
7.1.3	Bedienkonzept und Prüfablauf . . . . .	118
7.2	Parameterstudie am Prototyp . . . . .	119
7.2.1	Variablen des Algorithmus zur Verknüpfung . . . . .	121
7.2.2	Einfluss des Überlagerungsbereichs . . . . .	124
7.2.3	Variablen der automatischen Bahnplanung . . . . .	126
7.3	Experiment am Anwendungsbeispiel . . . . .	129
7.3.1	Versuchsaufbau und Durchführung . . . . .	129
7.3.2	Auswertung der Messdaten . . . . .	130
7.3.3	Bewertung nach technischen Aspekten . . . . .	133
7.4	Fallstudie zur Bewertung der Wirtschaftlichkeit . . . . .	135
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>141</b>
	<b>Literatur</b>	<b>145</b>
	<b>Anhang</b>	<b>171</b>



## Abkürzungsverzeichnis

<b>CAD</b>	„Computer Aided Design“
<b>COM</b>	„Component Object Model“
<b>ICP</b>	„Iterative Closest Point“
<b>PRM</b>	„Probabilistic Roadmap Method“
<b>RANSAC</b>	„Random Sample Consensus“
<b>PTP</b>	„Point-to-Point“
<b>RRT</b>	„Rapidly-Exploring Random Trees“
<b>SCARA</b>	„Selective Compliance Assembly Robot Arm“
<b>STL</b>	„Standard Tessellation Language“
<b>TCP</b>	„Tool Center Point“
<b>TCP/IP</b>	„Transmission Control Protocol/Internet Protocol“
<b>TSP</b>	„Travelling Salesman Problem“



## Verzeichnis der Formelzeichen

Bezeichner	Einheit	Beschreibung
$a, b, c$	–	Ortsvektoren zu Eckpunkten eines Dreiecks
$A, H$	–	Hilfsmatrix
$\Delta A$	$\text{mm}^2$	Differenz aus geschätzter und realer Fläche
$A_{real}$	$\text{mm}^2$	Fläche im Sichtbereich des 3D-Sensors
$A_{ij}$	–	Element von Matrix $A$ in Zeile $i$ und Spalte $j$
$\mathcal{C}()$	–	Operator für die Korrespondenzsuche
$d$	mm	Abstand des 3D-Sensors vom Prüfobjekt
$D_{ICP}$	–	Abstandsmetrik des ICP-Algorithmus
$D_{PRM}$	–	Abstandsmetrik der PRM
$e$	–	Ecke eines Graphen
$E$	–	Menge der Ecken eines Graphen
$f$	–	Freiheitsgrad
$G, R$	–	Graph aus Ecken und Kanten
$G_{reg}$	–	Graph mit sämtlichen Aufnahmeposes
$G_{tree}$	–	Spannender Baum notwendiger Posen
$G_{rob}$	–	Roadmap zur Planung der Roboterbahn
$G_{sample}$	–	Abtastung des freien Konfigurationsraums $\mathcal{Q}_{free}$
$h_0, h_1, h_2, h_3$	–	Elemente einer Quaternion
$\mathbf{h}_R$	–	Hilfsvektor zur Darstellung einer Quaternion
$\mathcal{H}()$	–	Operator für die Fehlerminimierung
$H$	–	Menge der Quaternionen
$i, j, n, m$	–	Laufvariable

## Verzeichnis der Formelzeichen

---

Bezeichner	Einheit	Beschreibung
$I_{3 \times 3}$	–	$(3 \times 3)$ -Einheitsmatrix
$k$	–	Kante eines Graphen
$k_{gauss}$	Pixel	Größe der Filtermaske bei der Glättung
$k_{thresh}$	–	Faktor bei der Kantenerkennung
$K$	–	Menge der Kanten eines Graphen
$K_M$	€	Belegungskosten der Maschine
$K_{M,rate}$	€	Kostensatz der Maschine
$K_P$	€	Personalkosten
$K_{P,rate}$	€	Kostensatz für Personal
$K_{task}$	€	Kosten für einen Prüfauftrag
$\mathcal{M}$	–	Geometrisches Prüfmerkmal
$\mathbf{n}$	–	Normalenvektor
$n_{pose}$	–	Anzahl Versuche zur Generierung einer Pose
$n_{neighbors}$	–	Anzahl betrachteter Nachbarn
$n_{add}$	–	Anzahl der Pixel, die generiert werden
$\mathcal{O}$	–	Hindernis im Arbeitsraum
$p$	–	Rasterpunkt auf dem Prüfobjekt
$p_{suc}$	%	Erfolgsquote bei der Suche nach gültigen Posen
$p_M$	–	Mittelpunkt des Sichtbereichs
$P_{detect}$	–	Erfassbare Rasterpunkte im Sichtbereich
$P_{proj}$	–	Rasterpunkte mit projiziertem Muster
$P_{reg}$	–	Für Ausrichtung nutzbare Rasterpunkte
$P_{sight}$	–	Rasterpunkte im Sichtbereich des 3D-Sensors
$\mathbf{q}$	–	Roboter-Konfiguration aus sechs Achswinkeln
$\mathcal{Q}$	–	Konfigurationsraum
$\mathcal{Q}_{free}$	–	Freier Konfigurationsraum
$\mathcal{Q}_{pert}$	–	Menge der relevanten Posen
$\mathcal{Q}_P$	–	Menge der Prüfposen

---

Bezeichner	Einheit	Beschreibung
$\mathcal{Q}_V$	–	Menge der Verknüpfungsposen
$r$	mm	Abstand der Rasterpunkte
$\mathbf{R}$	–	$(3 \times 3)$ -Rotationsmatrix
$R_{ij}$	–	Element von Matrix $\mathbf{R}$ in Zeile $i$ und Spalte $j$
$\mathbb{R}$	–	Menge der reellen Zahlen
$s$	m	Schrittweite für das RRT-Verfahren
$s_x, s_y$	–	Schwerpunkt einer Punktwolke
$\mathbf{t}$	–	$(3 \times 1)$ -Translationsvektor
$t_{c,discr}$	s	Berechnungszeit für die Diskretisierung
$t_{c,pose}$	s	Berechnungszeit für die Aufnahmeposes
$t_{c,path}$	s	Berechnungszeit für die Roboterbahn
$t_m$	s	Messzeit pro Merkmal
$t_{pre}$	s	Vorbereitungszeit
$t_{p,0}$	s	Berechnungszeit für CAD-Analyse
$t_{p,man}$	s	Programmierzeit pro Merkmal
$t_{p,auto}$	s	Berechnungszeit pro Merkmal
$t_{PTP}$	s	Fahrzeit für eine PTP-Bewegung
$t_{robot}$	s	Fahrzeit des Roboters für einen Prüfzyklus
$t_x, t_y, t_z$	–	Element des Vektors $\mathbf{t}$
$\mathbf{T}$	–	$(4 \times 4)$ -Transformationsmatrix
$\mathbf{T}_{ICP}$	–	Transformationsmatrix aus ICP-Algorithmus
$\mathbf{T}_{ref}$	–	Transformationsmatrix aus Referenzmessung
$\mathbf{T}_{robot}$	–	Transformationsmatrix aus Robotersteuerung
$u_{ref}$	mm	Standardmessunsicherheit der Ausrichtung
$\mathcal{W}$	–	Kartesischer Arbeitsraum
$\mathbf{x}, \mathbf{y}$	–	Bildpunkt der Punktwolke $\{X, Y\}$
$x, y, z$	mm	Kartesische Koordinate
$X, Y$	–	Punktwolke als Menge von Bildpunkten

## Verzeichnis der Formelzeichen

---

Bezeichner	Einheit	Beschreibung
$X_c$	–	Menge von Korrespondenzen
$\alpha$	°	Winkel zwischen 3D-Sensor und Prüfobjekt
$\Delta$	–	Hilfsvektor
$\varepsilon$	mm	Restfehler bei datengetriebener Ausrichtung
$\varepsilon_{abs}$	mm	Absoluter Fehler bei Kugelabstandsmessung
$\varepsilon_{pair}$	mm	Fehler bei paarweiser Verknüpfung
$\varepsilon_{path}$	mm	Fehler bei inkrementeller Registrierung
$\varepsilon_{ref}$	mm	Abweichung laut Referenzmessung
$\varepsilon_{rel}$	mm	Relativer Fehler bei Kugelabstandsmessung
$\varepsilon_{robot}$	mm	Erwartete Abweichung des Roboters
$\theta$	–	Gewicht einer Kante
$\Theta$	–	Distanz eines Weges
$\kappa$	mm <sup>2</sup>	Überlagerungsbereich zwischen Aufnahmen
$\Lambda()$	–	Operator für den lokalen Planer
$\nu$	–	Exakter Abstand zur Zielecke
$\hat{\nu}$	–	Geschätzter Abstand zur Zielecke
$\Sigma_{XY}$	–	Kovarianzmatrix
$\tau$	mm	Abbruchkriterium des ICP-Algorithmus
$\tau_{ICP}$	mm	Abstieg des Fehlermaßes
$\tau_{ref}$	mm	Abstieg der gemessenen Abweichung
$\phi$	°	Gelenkwinkel eines Roboters
$\psi$	–	Pfad im Konfigurationsraum
$\Psi$	–	Weg in einem Graph
$\Psi_{HK}$	–	Hamilton-Kreis
$\Psi_{reg}$	–	Pfad aus benachbarten Aufnahmen
$\Psi_{TSP}$	–	Taktzeitoptimierte Roboterbahn

---

## 1 Einleitung

Produzierende Unternehmen nutzen moderne Methoden des Qualitätsmanagements, um der zunehmenden Bedeutung einer hohen Qualität als produktprägende Eigenschaft gerecht zu werden (BRÜGGEMANN & BREMER 2012). Die Fertigungsmesstechnik liefert hierzu die datentechnische Grundlage für deren erfolgreichen Einsatz. Als prominentes Beispiel ist das Prüfen der geometrischen Maßhaltigkeit von Bauteilen im Automobil- und Luftfahrtbereich zu nennen. Die Qualität entspricht dabei dem Erfüllungsgrad der quantifizierten Anforderungen, welche an das Produkt gestellt werden (WEIDNER 2014, S. 12).

Neben der Qualität sind die Zielgrößen der Produktion Herstellkosten, Geschwindigkeit und Wandelbarkeit (ABELE & REINHART 2011, S. 34). Aus der Herausforderung diese Größen zu optimieren, ergibt sich unter anderem der Wunsch, mehr, öfter und schneller zu messen. Die verfügbare Messtechnik gerät dadurch unter einen erhöhten Kosten- und Innovationsdruck. Um die sich ergebenden Anforderungen zu konkretisieren, lassen sich die genannten Zielgrößen auf die Eigenschaften der eingesetzten Systemtechnik in der Fertigungsmesstechnik abbilden. Die Qualität selbst entspricht als quantifizierter Parameter der Genauigkeit des Messmittels und definiert die prüfbare Toleranz der Bauteilmerkmale. Die Kosten werden beeinflusst vom notwendigen Investitionsaufwand für die Systemtechnik sowie von den laufenden Kosten für die konkrete Messaufgabe. Die Zielgröße der Geschwindigkeit kann durch eine Verkürzung der Prüfzeit verbessert werden und die Wandelbarkeit im Kontext der Messtechnik beschreibt schließlich die Fähigkeit, kurzfristig auf wechselnde Prüfobjekte umzurüsten. Mit diesen Relationen lassen sich Messmittel in Bezug auf deren Eignung für den produktiven Einsatz unter aktuell gültigen Rahmenbedingungen bewerten.

## 1 Einleitung

---

In der produktionstechnischen Praxis lassen sich zwei wesentliche, in Abbildung 1.1 dargestellte, Anwendungsgebiete der Fertigungsmesstechnik identifizieren. Im Ersteren werden Bauteile direkt in der Fertigungslinie geprüft. Diese sogenannte *Inline*-Messtechnik ist dabei auf eine vorgegebene Taktzeit beschränkt, das Bauteil und damit die Prüfaufgabe bleibt über den gesamten Produktlebenszyklus konstant. Dadurch gewinnt die Eigenschaft der Geschwindigkeit gegenüber der Messgenauigkeit sowie Wandelbarkeit an Bedeutung. Vorteilhaft ist die Prüfung sämtlicher produzierten Werkstücke (100 %-Prüfung) und die damit einhergehende, sehr breite Datengrundlage für nachfolgende Qualitätssicherungsmethoden. Etablierte Messmittel für die *Inline*-Prüfung sind zum einen Mehrstellenmesseinrichtungen. Diese bestehen aus einer Vielzahl an stationären Sensoren, wodurch sich eine sehr hohe Geschwindigkeit der Prüfung ergibt, da alle Sensoren parallel Daten aufnehmen. Die Wandelbarkeit ist beschränkt, da das System hohen manuellen Aufwand zur Anpassung an wechselnde Prüfobjekte erfordert. Zum anderen werden roboterbasierte Messsysteme eingesetzt, die aus einem Industrieroboter und einem zumeist optisch arbeitenden Sensor bestehen. Diese Systeme weisen im Vergleich eine höhere Wandelbarkeit und Prüfgeschwindigkeit, jedoch verminderte Messgenauigkeit auf. (LETTENBAUER 2002; LUDWIG 2011)

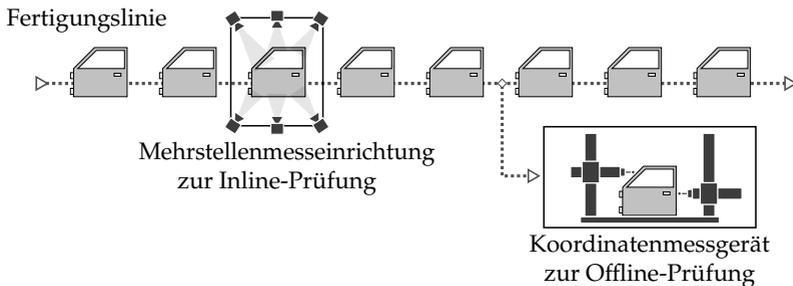


Abbildung 1.1: *Inline*-Messtechnik prüft jedes Bauteil partiell, *Offline*-Messtechnik hingegen fokussiert die vollumfängliche Prüfung einzelner Bauteile.

Im zweiten Anwendungsfall werden in regelmäßigen Abständen Stichproben der Bauteile separiert und gesondert geprüft. Bei dieser *Offline*-Prüfung gilt es, das Werkstück möglichst umfanglich und mit kleinstmöglichem Messfehler zu erfassen (JASTROW 2014). Eingesetzt werden dazu üblicherweise Koordinaten-

---

messgeräte. Eine hohe Messgenauigkeit als prägende Eigenschaft steht hier einem hohen Zeitaufwand zur Inbetriebnahme sowie Durchführung der Messung gegenüber. Die begrenzte Wandelbarkeit und Geschwindigkeit dieser Geräte begrenzt demnach die Leistungsfähigkeit der Offline-Prüfung, welche durch oft wechselnde Prüfobjekte und Messaufgaben charakterisiert ist. Nachteilig ist außerdem der hohe Investitionsaufwand für derartige Präzisionsmaschinen im Vergleich zu alternativen Systemen wie beispielsweise roboterbasierten Messsystemen. Zusätzlich werden handgeführte Messmittel eingesetzt. Durch die fehlende Möglichkeit zur Automatisierung und die direkte Abhängigkeit der Güte der Messdaten von Qualifikation und Konzentrationszustand des Personals werden handgeführte Messmittel nur verwendet, wenn die beschränkte Wandelbarkeit und Geschwindigkeit von Koordinatenmessgeräten für die aktuelle Prüfaufgabe nicht ausreicht (PFEIFER & SCHMITT 2010, S. 314).

Als Alternative für die Offline-Prüfung bieten sich roboterbasierte Messsysteme durch die frei programmierbare Kinematik an. Diese Systeme besitzen das Potenzial, sich gegenüber herkömmlich eingesetzten Messgeräten zu einer zeitsparenden und kosteneffizienten Alternative zu entwickeln (SCHMITT 2000). Bisher wird der Einsatz in der Praxis jedoch durch die geringe erreichbare Messgenauigkeit gehemmt. Die zu prüfenden Toleranzen in der Automobil- und Luftfahrtindustrie liegen momentan im Bereich von Zehntelmillimetern, die durch Industrieroboter erreichten Pose-Genauigkeiten hingegen im Millimeterbereich (ULRICH et al. 2014).



## 2 Grundlagen roboterbasierter Messtechnik

Die adressierten roboterbasierten Messsysteme nutzen zur Datenerfassung ein optisches Messmittel mit flächenhafter Antastung. Dieses wird durch einen Vertikal-Knickarmroboter, eine häufig genutzte Bauart der Industrieroboter, im Raum positioniert. Abbildung 2.1 zeigt den Aufbau des Systems, dessen Komponenten im Folgenden beschrieben werden.

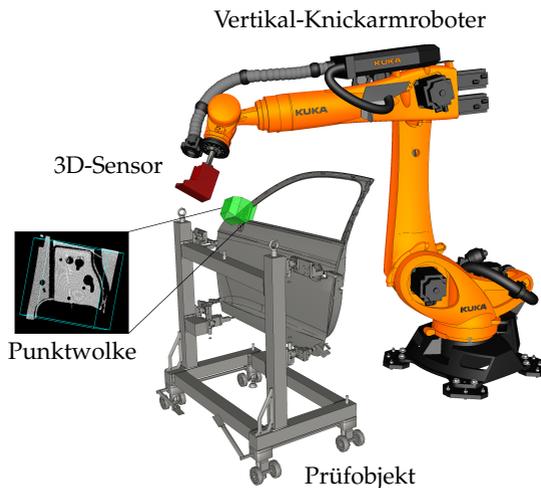


Abbildung 2.1: Aufbau eines roboterbasierten Messsystems. Das optische Messmittel (3D-Sensor) wird durch den Vertikal-Knickarmroboter im Raum positioniert und erzeugt eine digitale Abbildung (Punktwolke) des Prüfobjekts.

### 2.1 Optische 3D-Koordinatenmesstechnik

#### 2.1.1 Sensoren für die 3D-Bildgewinnung

Das Ziel optischer Prüfmittel in der 3D-Koordinatenmesstechnik ist die realitätsgetreue Abbildung einer betrachteten Szene in Form einer *Punktwolke* (KEFERSTEIN & MARXER 2015, S. 233). Bei der Bildaufnahme werden dreidimensionale Geometrien auf eine zweidimensionale Bildebene projiziert. Die Tiefe der Szene geht dabei verloren und muss durch geeignete Verfahren rekonstruiert werden. Die zur Erfassung genutzten Messsysteme werden als *3D-Sensoren* bezeichnet und existieren in verschiedenen Ausprägungen, die sich im physikalischen Grundprinzip zur Generierung der Tiefeninformation unterscheiden.

Basierend auf der bekannten und konstanten Ausbreitungsgeschwindigkeit des Lichts kann die Tiefe aus der Laufzeit des gesendeten Signals bestimmt werden, die sich proportional zur Strecke zwischen 3D-Sensor und aufzunehmender Oberfläche verhält. Das Verfahren ist schematisch in Abbildung 2.2.a visualisiert. Ein Spezialfall stellt die Methode der Interferometrie dar, bei der zusätzlich die Phasenlage gemessen wird. Die Genauigkeit erhöht sich dadurch bei gleichzeitig stark verringertem Messbereich. Voraussetzung für die Anwendung der Interferometrie ist die Verwendung von kohärentem Licht. (JÄHNE 2005, S. 239)

Alternativ kann die Tiefe durch Triangulation berechnet werden. Dafür wird die Szene aus zwei verschiedenen Positionen betrachtet. Es ergibt sich eine Verschiebung innerhalb der Bildebene, welche als Disparität bezeichnet wird. Ist der Abstand zwischen den Aufnahmepositionen bekannt, kann mit Hilfe der Disparität die Tiefe ermittelt werden. Wird an beiden Positionen die Szene durch Kameras aufgenommen, handelt es sich um ein passives Triangulationsverfahren (Abbildung 2.2.b). Dabei ist unerheblich, ob zwei Kameras genutzt werden oder eine Kamera während der Aufnahme die Position wechselt. (JÄHNE 2005, S. 231)

Bei aktiven Triangulationsverfahren wie in Abbildung 2.2.c hingegen wird eine einzelne, während der Aufnahme fixierte Kamera in Verbindung mit einer

Lichtquelle in definiertem Abstand genutzt. Aus der Beleuchtung folgt eine zusätzliche Bildinformation, aus der die Tiefeninformation ermittelt wird. Dabei ist eine Kodierung notwendig, um die zugehörigen Projektionen eindeutig zuordnen zu können. In der Praxis wird entweder eine Sequenz von Streifenmustern aufgebracht, oder ein fokussierter Laserstrahl bewegt sich rasterförmig über die Szene. Wird zu jedem Pixel des Kamerabildes die Tiefe berechnet, ergibt sich eine flächige Abbildung in Form der geforderten Punktwolke. (KEFERSTEIN & MARXER 2015, S. 235; PFEIFER & SCHMITT 2010, S. 215)

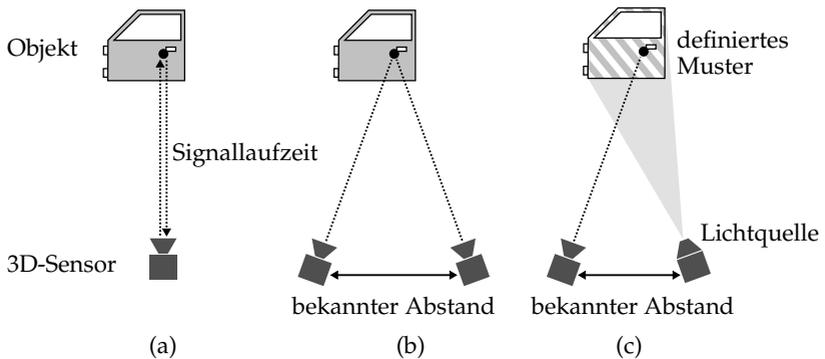


Abbildung 2.2: Prinzipien zur Berechnung von Tiefe aus Laufzeit (a), durch passive Triangulation (b) und aktive Triangulation (c).

In der Fertigungsmesstechnik haben sich 3D-Sensoren nach dem Prinzip der aktiven Triangulation durchgesetzt (KEFERSTEIN & MARXER 2015, S. 234; PFEIFER & SCHMITT 2010, S. 214). Diese in sich geschlossenen Systeme geben als Ergebnis eine rektifizierte, dreidimensionale Punktwolke aus. Abbildungsfehler (z. B. Verzerrungen) sind dabei bereits im Auslieferungszustand auf ein Minimum reduziert. Durch die Verwendung von Kameras als Teilkomponente zur Bildaufnahme können neben den geometrischen Dimensionen auch Intensitäts- und Farbinformationen abgebildet werden. Herausforderungen in der produktionstechnischen Praxis stellen sich vor allem durch nicht konstante Beleuchtungsbedingungen und stark reflektierende Oberflächen (PFEIFER & SCHMITT 2010, S. 224; KEFERSTEIN & MARXER 2015, S. 237).

### 2.1.2 Datenverarbeitung und Interpretation

Ziel einer digitalen Abbildung mit Hilfe von 3D-Sensoren ist die *Messung* im Sinne einer quantitativen Erfassung maßlich unbekannter Geometrien. Die anschließend folgende *Prüfung* wiederum ist die Herleitung einer Entscheidung, ob die gemessenen Werte innerhalb der geforderten Toleranz liegen. Ein zu prüfendes Bauteil wird als *Prüfobjekt* bezeichnet, die zu messenden geometrischen Größen als *Prüfmerkmale*. Diese umfassen für Funktion und Gestalt des Prüfobjekts relevante Längen- und Winkelmaße sowie Form- und Lagetoleranzen. (WECKENMANN 2012, S. 25; PFEIFER & SCHMITT 2010, S. 112)

Die Gesamtheit aller Prüfmerkmale und deren Eigenschaften werden in einem *Prüfplan* bereitgestellt. Dieser umfasst Nennwerte für die geometrischen Charakteristika der Merkmale sowie deren Toleranzwerte, Messstrategien sowie die Art der Dokumentation. Zusätzlich liegt meist ein ideales, virtuelles Volumenmodell des Prüfobjekts als „*Computer Aided Design*“ (CAD)-Datei vor. (WECKENMANN 2012, S. 254; KEFERSTEIN & MARXER 2015, S. 269)

Im Gegensatz zu taktilen Verfahren wird das Prüfmerkmal mit optischen 3D-Sensoren nicht direkt erfasst. Das Raster der aufgenommenen Punkte orientiert sich an den Kamerapixeln und nicht an der Struktur des Prüfobjekts. Dadurch bilden die erfassten Punkte vielmehr die unmittelbare Umgebung des Prüfmerkmals ab, nicht aber die zu messenden Werte. Die Punktwolke wird daher nachbearbeitet und algorithmisch interpretiert. (KEFERSTEIN & MARXER 2015, S. 238)

Zunächst werden Ausreißer entfernt und die Punktwolke geglättet, um stochastische Fehler des 3D-Sensors weitestgehend zu kompensieren. Es folgen die Flächenrückführung und Segmentierung, wodurch eine parametrische Beschreibung der Oberfläche des Prüfobjektes vorliegt und das Prüfmerkmal identifiziert ist. Zuletzt wird die ideale Geometrie des Prüfmerkmals in dieses virtuelle Abbild eingepasst und die Abweichung zwischen Nenngeometrie und erfasstem Prüfmerkmal ermittelt. (WECKENMANN 2012, S. 304; DEMANT et al. 2011, S. 95)

Mit Hilfe der Toleranzwerte aus dem Prüfplan wird das Ergebnis der Prüfung ermittelt und zusammen mit den Messdaten in einem *Prüfbericht* hinterlegt.

---

Zur visuellen Unterstützung werden die Messwerte dabei an den zugehörigen Stellen im CAD-Modell angetragen und über eine farbliche Kodierung das Ergebnis der Prüfung dargestellt. Je nach Prozess folgt eine Interpretation der Ergebnisse durch den Anwender, eine automatische Weitergabe an nachfolgende oder übergeordnete Systeme sowie die Archivierung des Prüfberichts. (WECKENMANN 2012, S. 314)

### 2.1.3 Registrierung von Punktwolken

Durch das begrenzte Sichtfeld des 3D-Sensors wird nur ein Ausschnitt des Prüfobjekts aufgenommen. Ist ein Prüfmerkmal in seinen geometrischen Ausdehnungen größer als der Sichtbereich, kann die Messung nicht durch die Aufnahme einer einzelnen Punktwolke durchgeführt werden. Stattdessen muss der Messwert aus den Daten mehrerer Punktwolken fusioniert werden. Dafür ist die Einordnung sämtlicher Aufnahmen in ein gemeinsames, globales Koordinatensystem notwendig. Die Transformation der Punktwolken von einem lokalen in ein globales Koordinatensystem wird *Registrierung* genannt (BENNAMOUN & MAMIC 2002, S. 101). Um Punktwolken innerhalb der Fertigungsmesstechnik zu registrieren, existieren verschiedene Methoden.

Im Falle der Handhabung durch einen Industrieroboter kann die Lage des 3D-Sensors im Raum durch Selbstlokalisierung des Roboters bestimmt werden (Abschnitt 2.2.1). Der Industrieroboter ist in diesem Fall Teil der Messkette, berechnet die Lage des 3D-Sensors und beeinflusst daher die erreichbare Messunsicherheit (WECKENMANN 2012, S. 231). Eine alternative Methode ist das optische Verfolgen (engl. Tracking) des 3D-Sensors im Raum durch ein externes System. Technische Realisierungen dieses Prinzips sind Lasertracker und Multi-Kamera-Systeme (WECKENMANN 2012, S. 156). Während der Messung muss die Sichtlinie zwischen *Tracking-System* und 3D-Sensor stets erhalten bleiben (KEFERSTEIN & MARXER 2015, S. 243). Abbildung 2.3 zeigt schematisch den Messaufbau beider Verfahren.

Die Registrierung in Abbildung 2.3.c durch Photogrammetrie verzichtet im Gegensatz dazu auf externe Messdaten. Zur Vorbereitung werden codierte *Passmarken* auf das Prüfobjekt aufgebracht, die während der Messung von Kameras erfasst und im Bild durch ihre einzigartige Gestalt leicht identifiziert werden

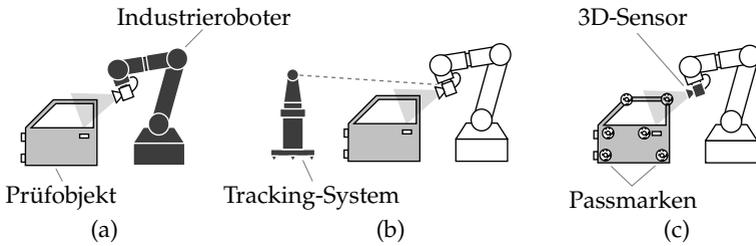


Abbildung 2.3: Registrierung von Punktwolken durch Selbstlokalisierung des Roboters (a), externes Tracking-System (b) und Passmarken, die durch den 3D-Sensor erfasst werden (c).

können. Durch Triangulation der Positionsdaten sämtlicher gefundenen Passmarken wird die Position und Orientierung bestimmt. In jeder möglichen Aufnahmeposition des 3D-Sensors müssen dabei stets mehrere Passmarken sichtbar sein. (KEFERSTEIN & MARXER 2015, S. 239; WECKENMANN 2012, S. 138)

## 2.2 Vertikal-Knickarmroboter

### 2.2.1 Aufbau und Funktionsweise

*Industrieroboter* sind automatisch gesteuerte Manipulatoren mit mindestens drei frei programmierbaren Gelenken. Die mechanischen Gelenke werden als *Achsen* des Roboters bezeichnet und sind geführte, unabhängig angetriebene Glieder der Kinematik. Eine eindeutige Gelenkstellung aller Achsen wird als *Pose* bezeichnet. (DIN EN ISO 8373; VDI 2861)

Die Anpassung an unterschiedliche Anwendungsgebiete ist durch eine mechanische Schnittstelle an der letzten Achse möglich, an der ein für den konkreten Einsatzzweck ausgewählter *Endeffektor* fixiert wird (z. B. Greifer für Handhabungsaufgaben). Als „Tool Center Point“ (TCP) wird ein frei wählbarer Arbeitspunkt bezeichnet, auf den sich die Bewegungsprogrammierung bezieht. Hierfür wird meist ein charakteristischer Punkt des Endeffektors gewählt. Der *Arbeitsraum* eines Roboters entspricht dem Raum, den der TCP durch die Gesamtheit

---

aller Achsbewegungen erreichen kann. (DIN EN ISO 8373; VDI 2861; WEBER 2009, S. 15)

Der kinematische Aufbau eines *Vertikal-Knickarmroboters* besteht überwiegend aus sechs seriell angeordneten, rotatorischen Achsen. Diese werden in Haupt- und Nebenachsen aufgeteilt, wobei die Hauptachsen zur Positionierung des Endeffektors im Raum dienen, während die Nebenachsen hauptsächlich dessen Orientierung festlegen. (WEBER 2009, S. 16)

Der *Freiheitsgrad*  $f$  beschreibt nach VDI 2861 die Anzahl möglicher Bewegungen des Robotersystems in Bezug zu einem festgelegten Koordinatensystem. Vertikal-Knickarmroboter besitzen den Freiheitsgrad  $f = 6$  und ermöglichen demnach eine beliebige Positionierung und Orientierung des Endeffektors im Arbeitsraum.

## Genauigkeit

Die Präzision, mit der ein Roboter den Endeffektor im Raum positionieren kann, wird mit Hilfe von zwei Kennwerten beschrieben. Ausgangsbasis ist dabei eine durch den Anwender vorgegebene Position und Orientierung des Endeffektors, welche mehrfach aus derselben Richtung angefahren wird. Nach DIN EN ISO 9283 und VDI 2861 beschreibt die *Pose-Genauigkeit* dabei die Abweichung des erreichten Mittelwerts von der Sollvorgabe, während die *Wiederholgenauigkeit* die Streuung um diesen Mittelwert angibt (Abbildung 2.4).

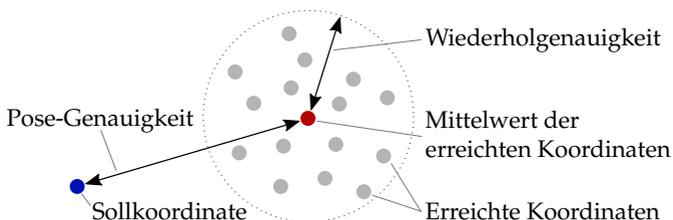


Abbildung 2.4: Pose- und Wiederholgenauigkeit eines Industrieroboters gemäß DIN EN ISO 9283 und VDI 2861

Handelsübliche, konventionelle Industrieroboter erreichen Wiederholgenauigkeiten von etwa  $60\ \mu\text{m}$  und Pose-Genauigkeiten im Bereich von mehreren

Millimetern (GERKE 2015, S. 156; ULRICH et al. 2014). Einflussfaktoren hierfür sind in Effekten wie Getriebeispiel, Elastizitäten, temperaturinduzierten Längenänderungen der Achselemente und Abweichungen bei der numerischen Interpolation innerhalb der Steuerung zu finden (GERKE 2015, S. 157; SHIAKOLAS et al. 2002). Diese Effekte können durch erhöhten Aufwand bei der Modellierung oder erweiterte Systemtechnik zum Teil kompensiert werden (BONGARDT 2004).

### **Vorwärts- und Rückwärtstransformation**

Als *Vorwärtstransformation* wird die eindeutige Abbildung der Pose eines Roboters auf die Position und Orientierung des zugehörigen Endeffektors bezeichnet (SICILIANO & KHATIB 2008, S. 26). Es gilt demnach, aus der Stellung sämtlicher Gelenke sowie den Abmessungen der Einzelelemente des Roboters den kartesischen Bezug des Arbeitspunkts zu einem feststehenden Koordinatensystem zu bestimmen. Dazu können beispielsweise nach dem Verfahren von Denavit und Hartenberg schrittweise Transformationsmatrizen entlang der kinematischen Kette des Roboters multipliziert werden (HUSTY et al. 1997, S. 425).

Die Umkehrung dieser Berechnung ist die *Rückwärtstransformation* (SICILIANO & KHATIB 2008, S. 27), bei der die Gelenkwinkel des Roboters aus der kartesisch angegebenen Position und Orientierung des Endeffektors berechnet werden. Im Gegensatz zur Vorwärtstransformation ist diese mathematisch nicht eindeutig (WEBER 2009, S. 56).

### **Bewegungsarten**

Industrieroboter sind in der Lage, Bewegungen zwischen zwei Posen auf unterschiedliche Arten durchzuführen. Die einfachste Art ist „Point-to-Point“ (PTP), bei der die Bahn des TCP im Raum für den Anwender nicht voraussehbar ist (Abbildung 2.5). Diese Bewegungsart eignet sich vor allem für Transferbewegungen, bei denen die Fahrzeit minimiert werden soll, die exakte Bahn des Roboters jedoch irrelevant ist. (WEBER 2009, S. 71)

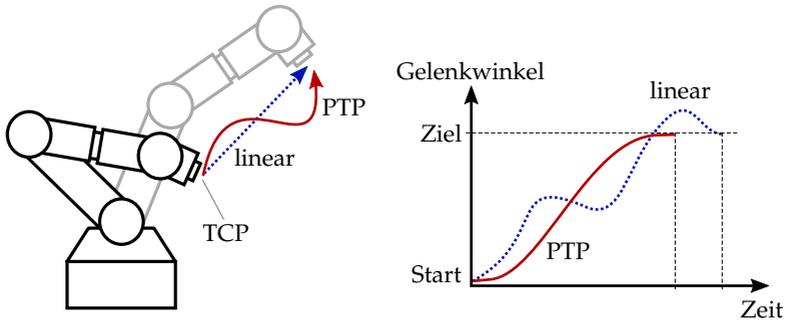


Abbildung 2.5: Der Industrieroboter kann bahngesteuert oder „Point-to-Point“ programmiert werden. Ersteres erzeugt eine definierte Bahn des Endeffektors, letzteres ist potentiell schneller.

Bei *asynchronen* PTP-Bewegungen agieren sämtliche Achsen unabhängig voneinander. Jede Achse fährt mit der programmierten Geschwindigkeit bis der Zielwert erreicht wurde. Bei *synchronen* PTP-Bewegungen hingegen kommen alle Achsen gleichzeitig zum Stillstand. Dazu wird zunächst die *Leitachse* bestimmt, welche für die vorgegebene Bewegung mit der programmierten Geschwindigkeit und Beschleunigung die längste Fahrzeit besitzt. Die verbleibenden Achsen passen ihr Bewegungsprofil daraufhin an, indem die maximale Geschwindigkeit reduziert wird. Die Fahrzeit der PTP-Bewegung wird dadurch nicht verändert, allerdings verringert sich die mechanische Belastung des Roboters durch kürzere Beschleunigungs- und Bremsvorgänge. (POTT et al. 2013, S. 327; GERKE 2015, S. 242; WEBER 2009, S. 79)

Für Prozesse, bei denen eine definierte Bahn des TCP notwendig ist (z. B. beim Lackieren), stehen bahngesteuerte Bewegungsprofile zur Verfügung. Bahnen können dabei linear, kreisförmig oder als frei definierbarer Spline vorgegeben werden (HAUN 2013, S. 246). Das Robotersystem berechnet für die Durchführung in jeder Zwischenstellung die Gelenkwinkel aus den kartesischen Koordinaten durch Rückwärtstransformation (CRAIG 2005, S. 217). Abbildung 2.5 zeigt beispielhaft das Verhalten des Roboters während einer linearen Bewegung des TCP im Vergleich zu einem PTP-Bewegungsprofil.

### 2.2.2 Programmierung

Industrieroboter sind per Definition frei programmierbar, es kann demnach das zur Ausführung einer Aufgabe notwendige Anwenderprogramm ohne physischen Eingriff erstellt werden (DIN EN ISO 8373). Die Anwenderprogramme bestehen dabei aus einer Folge von Bewegungsbefehlen (z. B. PTP-Bewegung zum Zielpunkt), Steuerbefehlen (z. B. Öffnen des Greifers) und Kontrollstrukturen (z. B. definierte Anzahl an Wiederholungen). Für Bewegungsbefehle abstrahiert das Robotersystem die Schnittstelle zum Anwender. Festzulegen sind Start- und Zielpunkt, die Art der Bewegung sowie Geschwindigkeits- und Beschleunigungsparameter. Auf Basis dieser Informationen werden die benötigten Gelenkstellungen kontinuierlich interpoliert. (WEBER 2009, S. 71)

Zur Bewegungsprogrammierung existieren unterschiedliche Methoden, wobei zunächst in *Online-* und *Offline-Verfahren* unterteilt wird. Die Online-Programmierung findet am Einsatzort statt, Posen werden direkt am Roboter vorgegeben und eingespeichert. Diese Verfahren umfassen das Einlernen von Posen durch Anfahren (*Teach-In*), die Programmierung durch Vormachen (*Play-Back*) und im rudimentären Fall die numerische Eingabe der Zielkoordinaten. Vorteilhaft ist dabei die Visualisierung direkt am Prozess inklusive eines sichtbaren Feedbacks der Eingabe. Nachteilig sind die Stillstandszeit des Robotersystems während der Programmierung sowie die mangelnde Skalierbarkeit. Sind komplexe Bewegungsabläufe umzusetzen, ist der Aufwand unverhältnismäßig hoch. (POTT et al. 2013, S. 327; WEBER 2009, S. 106; GERKE 2015, S. 251)

Bei Offline-Verfahren wird das Anwenderprogramm unabhängig vom realen Robotersystem erstellt. Dazu existieren zum einen *textuelle Programmiersprachen*, bei denen das Abstraktionsniveau allerdings sehr hoch ist, da keine visuelle Überprüfung der programmierten Bewegungen erfolgen kann. Zum anderen werden *CAD-basierte Simulationsumgebungen mit grafischer Schnittstelle* eingesetzt, in denen das Robotersystem und dessen direkte Umgebung modelliert werden. Die Erzeugung von Bewegungen erfolgt dann analog zum Teach-In-Verfahren, anstatt des realen Roboters wird jedoch das virtuelle Pendant zu den Zielkoordinaten bewegt. Die entstehende Roboterbahn kann dabei unmittelbar im Hinblick auf Kollisionen geprüft werden. Zuletzt wird das erstellte

---

Programm auf das reale System übertragen. (WEBER 2009, S. 112; BOHLIN 1999, S. 1)

Durch die Erstellung von Anwenderprogrammen unabhängig vom Industrieroboter wird die Stillstandszeit minimiert. Komplexe Aufgaben mit einer großen Anzahl an Bewegungen können zudem algorithmisch beschrieben werden, womit die Vorgabe jeder einzelnen Bewegung entfällt. Für den praktischen Einsatz ist dabei eine exakte und vollständige Modellierung des Roboters und dessen Umgebung notwendig, wodurch ein nicht zu unterschätzender Aufwand entsteht. Ein inhärenter Nachteil der Offline-Methoden ist zudem die Abweichung zwischen realer und virtueller Bewegung durch die begrenzte Pose-Genauigkeit des Industrieroboters. (HAUN 2013, S. 239; PAN et al. 2012)

Um Vorteile von Online- und Offline-Verfahren zu verbinden, steht die hybride Programmierung zur Verfügung. Die Roboterbahn wird hierbei mit Hilfe einer CAD-basierten Simulation grob vorgegeben und während der anschließenden Inbetriebnahme individuell korrigiert (GERKE 2015, S. 254). Im Anwendungsgebiet der Inline-Messtechnik wird auf diese Art der Programmierung zurückgegriffen werden, um den hohen Taktzeitanforderungen gerecht zu werden.

Der hohe Aufwand zur Erstellung von Anwendungsprogrammen ist ein wesentlicher Hinderungsgrund für kleine und mittelständische Unternehmen, Industrieroboter einzusetzen (PAN et al. 2012). Die Vereinfachung der Programmierung ist daher ein wesentliches Ziel, um Roboter in neuen Einsatzgebieten zu etablieren (BUCHMANN et al. 2015). Dazu können beispielsweise „Augmented Reality“-Technologien genutzt werden, durch welche sich die Vorgabe von Roboterbewegungen intuitiver und ergonomischer gestaltet (LAMBRECHT et al. 2013; VOGL 2009).

Ein tiefer greifender Ansatz zur Vereinfachung ist die aufgabenorientierte Programmierung. Hierbei wird die auszuführende Aufgabe vom Anwender abstrakt vorgegeben, während das Robotersystem daraus konkrete Steuer- und Bewegungsbefehle erzeugt. Der Anwender benötigt dabei weniger Systemwissen über den eingesetzten Roboter als bei konventionellen Programmierverfahren. Im besten Fall übernimmt das System zusätzliche Aufgaben wie die Prüfung der Erreichbarkeit und Kollision mit Hindernissen. Bisher vorgestellte Methoden fokussieren stets spezielle Prozesse wie das Remote-Laserstrahl-Schneiden

(HATWIG et al. 2012) oder die Prüfung von Karosserie-Bauteilen mittels Li-niensensoren (TEKOUO MOUTCHHO 2012) und sind damit nicht universell einsetzbar. (HAUN 2013, S. 239; WEBER 2009, S. 114)

### 2.2.3 Industrieller Einsatz

Roboter prägen seit ihrer Einführung um 1960 die Montageautomatisierung (FELDMANN et al. 2013, S. 12). Begründet werden kann dies mit der kinematischen Beweglichkeit, Robustheit, hohen erreichbaren Kräften und Geschwindigkeiten. Eingesetzt werden Industrieroboter dabei im Zusammenschluss mit Endeffektoren, Zuführtechnik und Sensoren zur Steuerung und Kontrolle der Abläufe. Das Gesamtsystem dieser Komponenten wird in einer *Roboterzelle* gekapselt und durch Schutzeinrichtungen vom Menschen getrennt (POTT et al. 2013, S. 316). In jüngerer Zeit werden alternativ Konzepte für die Mensch-Roboter-Kooperation erprobt, um die typischerweise hohen Aufwände für den Betrieb einer Roboterzelle zu vermindern und Vorteile von Mensch und Industrieroboter zu kombinieren (GERKE 2015, S. 121; SPILLNER 2015).

Neben Vertikal-Knickarmrobotern sind in der industriellen Praxis kartesische Roboter (Portal-Roboter), Horizontal-Knickarmroboter (SCARA) sowie parallele Roboter (Delta-Roboter) etabliert. Der kinematische Aufbau unterscheidet sich dabei grundlegend und beeinflusst Eigenschaften wie die erreichbare Genauigkeit, Lastkapazität und Stellfläche (POTT et al. 2013, S. 319). Vertikal-Knickarmroboter sind mit 61 % die weltweit am häufigsten eingesetzte Bauart in der Produktionstechnik (IFR STATISTICAL DEPARTMENT 2016, S. 94). Die Gründe hierfür sind ein vergleichsweise gutes Verhältnis von kleiner Stellfläche zu großem Arbeitsraum sowie die Möglichkeit, Position und Orientierung des Endeffektors frei festlegen zu können (SICILIANO & KHATIB 2008, S. 975).

Industrieroboter werden für zahlreiche Anwendungsgebiete eingesetzt, hauptsächlich in Handhabungs- (44 %), Schweiß- (28 %) und Montage-Prozessen (10 %) (IFR STATISTICAL DEPARTMENT 2016, S. 42). Bei vielen weiteren Applikationen werden zusätzliche Maßnahmen zur Befähigung des Robotersystems für den entsprechenden Prozess getroffen. So ist für den Einsatz beim Umformen, Fräsen und Rührreibschweißen die Betrachtung der Nachgiebigkeit der Kinematik sowie die Kompensation dadurch begründeter Bahnabweichungen

---

unerlässlich (THYSSEN et al. 2016; RÖSCH 2015; VÖLLNER 2010). Durch die kinematische Beweglichkeit sind Industrieroboter auch für die Fertigungsmesstechnik interessant, jedoch ist hierbei die Pose-Genauigkeit am Markt erhältlicher Systeme ohne weitere Modifikationen nicht ausreichend (LETTENBAUER 2002; LUDWIG 2011).



## **3 Zielstellung und Vorgehensweise**

### **3.1 Defizite und Grenzen roboterbasierter Messsysteme**

Roboterbasierte Messtechnik besitzt zahlreiche vorteilhafte Eigenschaften für den Einsatz in der automatischen Qualitätskontrolle (KIMMELMANN 1991). Die eingesetzten Industrieroboter sind kostengünstig, haben eine hohe Beweglichkeit bei gleichzeitig geringer Stellfläche und durch 3D-Sensoren mit rein optischem Messprinzip können Prüfaufgaben schneller als mit konventionellen taktilen Messprinzipien durchgeführt werden. Der betrachtete Einsatzbereich ist die Qualitätssicherung durch Prüfung von Geometrie und Form von Bauteilen in der Automobil- und Luftfahrtindustrie. Die zu prüfenden Toleranzen liegen dabei im Bereich von Zehntelmillimetern, die geometrischen Dimensionen der Prüfmerkmale übersteigen teilweise den Sichtbereich des 3D-Sensors. Es ist daher eine Registrierung der aufgenommenen Punktwolken nötig, wofür in der produktionstechnischen Praxis die in Abschnitt 2.1.3 aufgezeigten Methoden angewandt werden.

Werden die zur Registrierung notwendigen Positionsdaten durch den Industrieroboter berechnet, beeinflusst dieser direkt die Messgenauigkeit des Gesamtsystems. Die Pose-Genauigkeit aktuell am Markt erhältlicher Vertikal-Knickarmroboter liegt im Millimeterbereich und muss daher für den praktischen Einsatz erhöht werden. Ein Lösungsansatz hierfür ist die regelmäßige Kalibrierung des Roboters. Diese Methode setzt an der Ursache der fehlenden Genauigkeit an und modelliert den kinematischen Aufbau, die Getriebedynamik und andere Effekte, welche die Pose-Genauigkeit beeinflussen. Mit Hilfe des entwickelten Modells kann durch Vorwärts-Transformation die Pose des 3D-Sensors errechnet werden. (SCHRÖER 1993; WIEST 2001; LUX & REINHART 2015)

### 3 Zielstellung und Vorgehensweise

---

Für die Parametrierung des Modells sind extern referenzierte Daten nötig, die durch Tracking-Systeme erhoben werden. Diese Abhängigkeit von externen Messgeräten erhöht den Aufwand zur Automatisierung des Prozesses, gleichzeitig steigt die Systemkomplexität für den Anwender. Zu beachten ist weiterhin, dass während der Prüfung keine Informationen zur Verfügung stehen, ob die Kalibrierung weiterhin valide Werte liefert oder sich das Verhalten des Roboters mit der Zeit verändert hat. Ein gut dokumentiertes Beispiel hierfür sind längeninduzierte Änderungen der einzelnen Achselemente durch Erwärmung der Motoren im Betrieb. Es werden daher regelmäßig neue Referenzdaten erhoben, um die Parameter des Modells anzupassen. (REINHART et al. 1998; BONGARDT 2004)

Verfahren zur Kalibrierung streben eine Steigerung der Pose-Genauigkeit an. Die Wiederholgenauigkeit des Roboters bildet dabei eine inhärente Grenze der Methode, die per se nicht unterschritten werden kann. Diese beträgt für am Markt erhältliche, unmodifizierte Vertikal-Knickarmroboter etwa 60  $\mu\text{m}$  (ABB AUTOMATION GMBH 2014b; FANUC DEUTSCHLAND GMBH 2017). Dieser Kennwert sowie der Zustand und die Ausprägung der Hard- und Software eines Industrieroboters sind zusätzlich abhängig von Hersteller, Größe, Alter und bisheriger Verwendung. Für jeden eingesetzten Roboter ist daher mit Hilfe von Referenzmessungen zunächst ein individueller Parametersatz zu identifizieren (SCHRÖER 1993; PFEIFFER & HOLZL 1995).

Alternativ werden für die Registrierung Tracking-Systeme eingesetzt (WECKENMANN 2012, S. 155; SCHMITT et al. 2010). Die Inbetriebnahme ist hierbei komplexer durch die Erweiterung des Systemaufbaus und die stets zu erhaltene Sichtlinie zwischen 3D-Sensor und Tracking-System. Einschränkungen ergeben sich dadurch beim Auswählen der Prüfmerkmale, welche in der praktischen Anwendung aus lediglich einer Perspektive sichtbar sein müssen (KEFERSTEIN & MARXER 2015, S. 242). Zusätzlich sind Systeme mit hinreichender Präzision durch hohe Investitionskosten charakterisiert.

Photogrammetrie als dritter Ansatz zur Registrierung setzt eine aufwändige Arbeitsvorbereitung voraus, bei der zahlreiche codierte Passmarken auf dem Prüfobjekt oder zusätzliche Kulissen angebracht werden (BERTAGNOLLI 2006). Der vorteilhafte Effekt einer potentiell kürzeren Prüfzeit roboterbasierter Mess-

---

systeme wird dabei durch die zusätzliche Zeit zur Vorbereitung der Messung vermindert.

Eine weitere Herausforderung bei der praktischen Anwendung roboterbasierter Messsysteme ist der Aufwand zur Programmierung (REINHART & TEKOUO 2009). Der Prüfplan muss dazu interpretiert und eine kollisionsfreie Roboterbahn abgeleitet werden. Der manuelle Aufwand hierfür ist zeitaufwändig und komplex, sodass der Anwender neben der messtechnischen Expertise zusätzliches Wissen im Bereich der industriellen Robotik benötigt.

### **3.2 Zielstellung und Lösungsansatz**

Aus dem aufgezeigten Stand der roboterbasierten Messtechnik lässt sich der Handlungsbedarf ableiten. Zur Prüfung geometrischer Charakteristika muss die Messgenauigkeit bei der Registrierung erhöht werden. Aktuell verfügbare Lösungen schränken jedoch die ursprünglichen Vorteile der hohen Flexibilität und Geschwindigkeit ein, zusätzlich erhöhen sich Systemkomplexität und Investitionskosten. Die Zielstellung der vorliegenden Arbeit ist daher die Qualifizierung roboterbasierter Messsysteme zur flexiblen, schnellen und kostengünstigen Prüfung geometrischer Prüfmerkmale. Daraus ergibt sich zum einen eine Steigerung der Messgenauigkeit im Vergleich zum unmodifizierten Robotersystem, wobei die genannten Vorteile des ursprünglichen Systems erhalten bleiben sollen. Zum anderen ist der Aufwand zur Programmierung zu verringern, um dem Anwender die Fokussierung auf den Prüfprozess zu ermöglichen.

Grundlegender Lösungsansatz ist die *datengetriebene Registrierung*. Anstatt auf externe Daten angewiesen zu sein, werden die Informationen zur Registrierung allein aus den aufgenommenen Daten des 3D-Sensors berechnet. Dazu wird der Pfad des Roboters durch zusätzliche Posen zur Aufnahme von Punktwolken ergänzt, die sich jeweils paarweise überdecken. Anschließend erfolgt eine Verknüpfung benachbarter Punktwolken, indem im gemeinsamen Bildausschnitt korrespondierende Merkmale gesucht und zueinander ausgerichtet werden (Abbildung 3.1). Dadurch kann die räumliche Beziehung zwischen den Aufnahmen

### 3 Zielstellung und Vorgehensweise

---

errechnet werden, um diese anschließend in ein globales Koordinatensystem zu überführen.

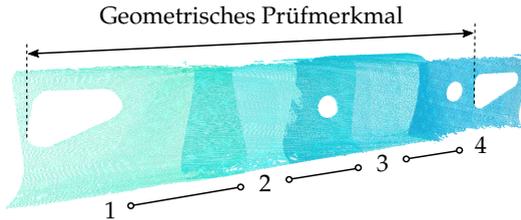


Abbildung 3.1: Für eine datengetriebene Registrierung werden die Punktwolken 1 bis 4 verknüpft, um das geforderte Prüfmerkmal messen zu können.

Die erreichbare Genauigkeit der Registrierung ist dabei vom eingesetzten Vertikal-Knickarmroboter unabhängig und wird stattdessen maßgeblich durch den 3D-Sensor bestimmt. Tracking-Systeme sind ebenso wie physische Referenzmarken nicht vorgesehen, wodurch Flexibilität und Geschwindigkeit des unmodifizierten Systems erhalten bleiben.

Der 3D-Sensor wurde zur Prüfung geometrischer Merkmale im Submillimeterbereich konstruiert. Es wird daher angenommen, dass die Qualität der Punktwolken ebenfalls für eine Verknüpfung im Submillimeterbereich ausreicht. Unabhängig davon verbleibt ein Restfehler bei jeder Verknüpfung. Da die Punktwolken seriell verknüpft werden, pflanzt sich dieser fort und steigt mit der Anzahl verknüpfter Punktwolken. Zu beachten ist dieser Zusammenhang bei der Planung zusätzlicher Aufnahmeposes und der damit einhergehenden Generierung der Roboterbahn. Weiterhin sind geeignete Oberflächen des Prüfbjektivs zu identifizieren und zu einem verknüpfbaren Pfad aus einzelnen Punktwolken zu verbinden. Um diesen Anforderungen gerecht zu werden und die Nutzung des Systems von Expertenwissen weitestgehend zu entkoppeln, wird eine automatisierte Programmerstellung angestrebt. Die Bahn des Roboters soll dafür auf Basis der CAD-Daten des Prüfbjektivs erzeugt werden.

Als „Image Stitching“ wird in der Fotografie die Verknüpfung einzelner Aufnahmen zu einem Gesamtbild bezeichnet (SZELISKI 2011, S. 429). In Analogie zu diesem Begriff wird die vorgestellte Methode *3D-Image-Stitching* genannt. Zur erfolgreichen Umsetzung des Konzepts ergeben sich zwei Aufgabenbereiche:

---

die Registrierung der Punktwolken sowie die zugehörige Programmerstellung für das Messsystem. Daher bezieht sich der Stand der Forschung und Technik in Kapitel 4 zum einen auf die Ausrichtung von Punktwolken auf Basis der Bilddaten und zum anderen auf die Posen- und Bahnplanung von Industrierobotern. Kapitel 5 stellt anschließend die Methode zur Verknüpfung der Punktwolken vor, während Kapitel 6 die Generierung der Roboterbahn adressiert. Das Gesamtsystem wird in Kapitel 7 anhand einer prototypischen Realisierung evaluiert.



## 4 Stand der Forschung und Technik

### 4.1 Datengetriebene Registrierung von Punktwolken

Das Überführen mehrerer Punktwolken in ein gemeinsames Koordinatensystem allein auf Basis der Messdaten wird zur Abgrenzung gegenüber alternativen Techniken als datengetriebene Registrierung bezeichnet. Anwendungsgebiete umfassen unter anderem die Rekonstruktion von Skulpturen (LEVOY et al. 2000), bildgebende Medizintechnik (GRIMSON et al. 1996) und Navigation mobiler Roboter (BIBER & STRASSER 2003).

Mathematisch wird die Relation zwischen zwei Punktwolken im dreidimensionalen Raum  $\mathbb{R}^3$  über *Transformationen* beschrieben. Bleiben die geometrischen Abstände innerhalb der Punktwolken erhalten, handelt es sich um eine *starre* Transformation (DUNN & PARBERRY 2011, S. 158). Diese umfasst ausschließlich Rotationen und Translationen (Verschiebungen). Im betrachteten Anwendungsfall der Fertigungsmesstechnik werden unverzerrte Punktwolken registriert. Zugehörige Transformationen sind daher durchweg als starr zu betrachten, auch wenn dies nicht explizit benannt wird. Die Transformation der Punktwolke  $Y$  nach  $X$  mit einer Rotation  $\mathbf{R}$  und Translation  $\mathbf{t}$  berechnet sich nach Gleichung 4.1.

$$X = \mathbf{R} \cdot Y + \mathbf{t} \quad (4.1)$$

*Homogene Koordinaten* werden verwendet, um Rotation und Translation in einer einzigen Matrix darzustellen. Die  $(3 \times 3)$ -Rotationsmatrix  $\mathbf{R}$  und der  $(3 \times 1)$ -Translationsvektor  $\mathbf{t}$  werden in eine  $(4 \times 4)$ -Transformationsmatrix  $\mathbf{T}$  überführt (Gleichung 4.2). Die Darstellung von Punkten erweitert sich zu  $\mathbf{x} = [x \ y \ z \ 1]^T$ .

Analog zu Gleichung 4.1 ergibt sich die Darstellung einer Transformation nach Gleichung 4.3. (JÄHNE 2005, S. 224)

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$X = T \cdot Y \quad (4.3)$$

### 4.1.1 Registrierung durch bekannte Korrespondenzen

Die Aufgabe der datengetriebenen Registrierung ist die Transformation einer Punktwolke  $Y$  mit  $n$  Punkten, sodass diese sich in optimaler Ausrichtung zur Punktwolke  $X$  befindet. Um die Ausrichtung durchzuführen, muss für jeden beliebigen Punkt  $y_i \in Y$  der korrespondierende Punkt  $x_i \in X$  bekannt sein, der den gleichen realen Bereich darstellt (Gleichung 4.4). Abbildung 4.1.a zeigt beispielhaft derartige zusammengehörige Bereiche, die als *Korrespondenzen* bezeichnet werden. Da Messdaten in der Praxis keine perfekte Darstellung der Realität liefern, verbleibt bei der Ausrichtung wie in Abbildung 4.1.b stets ein unvermeidbarer Restfehler  $\varepsilon$  (Gleichung 4.5). Dessen Minimierung ist das Ziel der aufgezeigten Methoden. (JÄHNE 2005, S. 418; BENNAMOUN & MAMIC 2002, S. 107)

$$x_i = T \cdot y_i \quad \text{mit } i = 1 \dots n \quad (4.4)$$

$$\varepsilon = \frac{1}{n} \sum_{i=1}^n \|x_i - T \cdot y_i\|^2 \quad (4.5)$$

Die ideale Transformation kann analytisch nach der von HORN (1987) vorgestellten Methode mit Hilfe von *Quaternionen* gefunden werden. Als Generalisierung der komplexen Zahlen bilden Quaternionen den mathematischen Schiefkörper  $\mathbb{H} = \{(h_0, h_1, h_2, h_3) \mid h_0, h_1, h_2, h_3 \in \mathbb{R}\}$  und werden für die Darstellung von

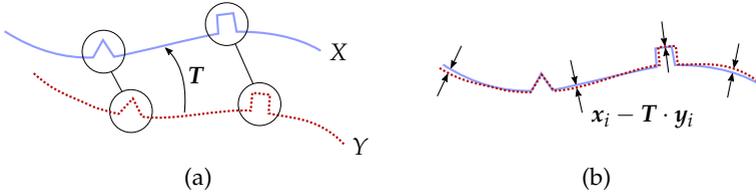


Abbildung 4.1: Korrespondenzen werden bei der datengetriebenen Registrierung zueinander ausgerichtet (a), wobei stets ein Restfehler verbleibt (b).

Rotationen als  $\mathbf{h}_R = [h_0 \ h_1 \ h_2 \ h_3]$  mit  $h_0 \geq 0$  und  $h_0^2 + h_1^2 + h_2^2 + h_3^2 = 1$  genutzt (BEUTELSPACHER 2014, S. 36; BENNAMOUN & MAMIC 2002, S. 108).

Zunächst werden dafür die Schwerpunkte der Punktwolken  $\mathbf{s}_X = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$  und  $\mathbf{s}_Y = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i$  ermittelt. Es folgt die Berechnung der Kovarianzmatrix  $\Sigma_{XY}$  (Gleichung 4.6). Mittels Hilfsmatrix  $\mathbf{A} = \Sigma_{XY} - \Sigma_{XY}^T$  und dem Hilfsvektor  $\Delta = [A_{32} \ A_{31} \ A_{12}]^T$  wird daraus die Matrix  $\mathbf{H}$  nach Gleichung 4.7 gebildet. Dabei entspricht  $\mathbf{I}_{3 \times 3}$  einer Einheitsmatrix mit drei Dimensionen, die Spur einer Matrix ist die Summe ihrer Diagonalelemente (BEUTELSPACHER 2014, S. 252). Anschließend wird  $\mathbf{h}_R$  als Einheits-Eigenvektor zum größten positiven Eigenwert von  $\mathbf{H}$  bestimmt. Gleichung 4.8 und Gleichung 4.9 werden zuletzt zur Berechnung der Rotationsmatrix  $\mathbf{R}$  und des Translationsvektors  $\mathbf{t}$  herangezogen. (HORN 1987; BENNAMOUN & MAMIC 2002, S. 108)

$$\Sigma_{XY} = \sum_{i=1}^n (\mathbf{x}_i - \mathbf{s}_X)(\mathbf{y}_i - \mathbf{s}_Y)^T \quad (4.6)$$

$$\mathbf{H} = \begin{bmatrix} \text{Spur}(\Sigma_{XY}) & \Delta^T \\ \Delta & \Sigma_{XY} + \Sigma_{XY}^T - \text{Spur}(\Sigma_{XY}) \cdot \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (4.7)$$

$$\mathbf{R} = \begin{bmatrix} h_0^2 + h_1^2 - h_2^2 - h_3^2 & 2(h_1 h_2 - h_0 h_3) & 2(h_1 h_3 + h_0 h_2) \\ 2(h_1 h_2 + h_0 h_3) & h_0^2 + h_2^2 - h_1^2 - h_3^2 & 2(h_2 h_3 - h_0 h_1) \\ 2(h_1 h_3 - h_0 h_2) & 2(h_2 h_3 + h_0 h_1) & h_0^2 + h_3^2 - h_1^2 - h_2^2 \end{bmatrix} \quad (4.8)$$

$$\mathbf{t} = \mathbf{s}_X - \mathbf{R} \cdot \mathbf{s}_Y \quad (4.9)$$

Neben der Berechnung durch Quaternionen führen BENNAMOUN & MAMIC (2002, S. 108) die Singulärwertzerlegung (ARUN et al. 1987), die „Dual Quaternion Method“ (WALKER et al. 1991) und die Methode mit orthogonalen Matrizen (HORN et al. 1988) als fundamentale Verfahren zur Registrierung mit gegebenen Korrespondenzen an. Bei einer vergleichenden Analyse der vier Algorithmen durch EGGERT et al. (1997) konnten keine wesentlichen Unterschiede bezüglich Effizienz, Robustheit und Genauigkeit festgestellt werden.

### 4.1.2 Der „Iterative Closest Point“-Algorithmus

Für Anwendungsfälle ohne bekannte Korrespondenzen hat sich der von BESL & MCKAY (1992) vorgestellte „Iterative Closest Point“ (ICP)-Algorithmus zur Registrierung von Punktwolken (aber auch für Kurven und Oberflächen) als De-facto-Standard etabliert (WECKENMANN 2012, S. 232). Ziel ist dabei eine realitätsgetreue Überlagerung der Punktwolken. Dazu werden die Punkte der Punktwolke  $Y = \{y_i\}$  an die Referenz-Punktwolke  $X = \{x_i\}$  durch eine starre Transformation  $T$  schrittweise angenähert. Abbildung 4.2 visualisiert verkürzt den Ablauf einer Ausrichtung mit Hilfe des ICP-Algorithmus.

Im ersten Schritt werden potentielle Korrespondenzen bestimmt, jedem Punkt in  $Y$  wird jeweils ein potentiell zugehöriger Punkt in  $X$  zugewiesen (Abbildung 4.2.a). Für die Auswahl der Punktpaare wird die Abstandsmetrik  $D_{ICP}$  (Gleichung 4.10) genutzt, welche den minimalen Abstand zwischen Punkt  $y$  und der Punktwolke  $X$  definiert. Der zugehörige Punkt  $x_c$  mit  $D_{ICP}(y, X) = D_{ICP}(y, \{x_c\})$  entspricht der Korrespondenz zu  $y$ . Mit dem Operator  $\mathcal{C}$  wird die Suche nach Korrespondenzen mit Hilfe der Metrik  $D_{ICP}$  bezeichnet, womit nach Gleichung 4.11 eine Menge von Korrespondenzen  $X_c$  bezogen auf  $Y$  berechnet werden kann. (BENNAMOUN & MAMIC 2002, S. 113; BESL & MCKAY 1992)

$$D_{ICP}(y, X) = \min_{x \in X} \|x - y\| \quad (4.10)$$

$$X_c = \mathcal{C}(X, Y) \quad (4.11)$$

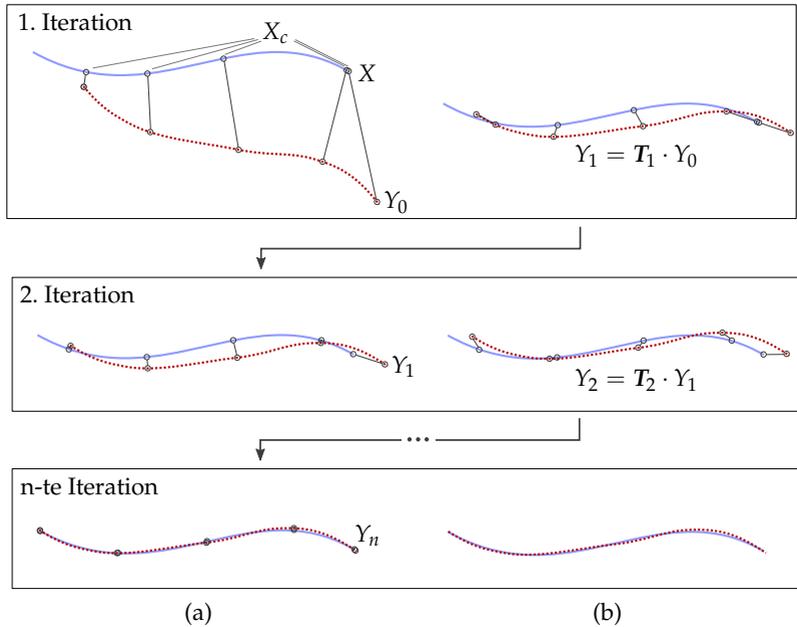


Abbildung 4.2: Beispielhafte Ausrichtung mittels ICP-Algorithmus. Iterativ werden korrespondierende Punkte bestimmt (a), anschließend wird deren Abstand minimiert (b).

Für eine beliebige Transformation  $T$  kann durch das Fehlermaß  $\epsilon$ , basierend auf dem mittleren quadratischen Abstand zwischen korrespondierenden Punkten, die Güte der Überlagerung bestimmt werden (Gleichung 4.5). Das Finden der optimalen Transformation mit minimalem Restfehler wird mit dem Operator  $\mathcal{H}$  bezeichnet (Abbildung 4.2.b). Als geeignete Optimierungsmethode für dreidimensionale Punktwolken wird das Verfahren mit Quaternionen (Abschnitt 4.1.1) empfohlen (BESL & MCKAY 1992).

$$(T, \epsilon) = \mathcal{H}(X, Y) \quad (4.12)$$

Die gefundenen Korrespondenzen sind nicht zwangsweise die tatsächlich zueinander gehörenden Punktpaare. Die Minimierung führt daher nicht zwingend

zu einer optimalen Überlagerung. Erst das wiederholte Durchführen der Korrespondenzsuche  $\mathcal{C}$  und die anschließende Minimierung des Fehlermaßes  $\varepsilon_i$  mit Hilfe einer schrittweise angenäherten Punktwolke  $Y_i$  führen zu einer mutmaßlich optimalen Registrierung. Das iterative Vorgehen ist in Algorithmus 4.1 dargestellt. Der ICP-Algorithmus konvergiert stets gegen ein lokales Minimum, was für Anwendungen mit nicht-konvexen Lösungsräumen beachtet werden muss. Wenn die aktuelle Iteration im Vergleich zur vorherigen keine relevante Verbesserung erreicht, bricht der Algorithmus ab (n-te Iteration in Abbildung 4.2). Dieses Abbruchkriterium wird durch den Grenzwert  $\tau$  quantifiziert. (BENNAMOUN & MAMIC 2002, S. 114; BESL & MCKAY 1992)

---

### Algorithmus 4.1 ICP-Algorithmus nach BESL & MCKAY (1992)

---

```
1: procedure ICP( $Y, X, \tau$ )
2:    $Y_0 \leftarrow Y$ 
3:    $i \leftarrow -1$ 
4:   repeat
5:      $i \leftarrow i + 1$                                 ▷ Laufindex erhöhen
6:      $X_{c,i} \leftarrow \mathcal{C}(X, Y_i)$                     ▷ Korrespondenzen bestimmen
7:      $T_i, \varepsilon_i \leftarrow \mathcal{H}(X_{c,i}, Y_0)$         ▷ Transformation berechnen
8:      $Y_{i+1} \leftarrow T_i \cdot Y_0$                   ▷ Anwenden der Transformation
9:   until  $(\varepsilon_{i-1} - \varepsilon_i) < \tau$              ▷ Verbesserung der Iteration prüfen
10:  return  $T_i$                                        ▷ Ausgabe der optimalen Transformation
11: end procedure
```

---

Für den ICP-Algorithmus existieren zahlreiche Modifikationen und Weiterentwicklungen zur Anpassung auf verschiedenste Anwendungsgebiete. Relevante Beiträge werden im Folgenden vorgestellt.

### Initiale Transformation

Die Robustheit des ICP hängt von der initialen Ausrichtung beider Punktwolken zueinander ab (WECKENMANN 2012, S. 232; ZINSSER et al. 2003). Falls eine geschätzte Transformation  $T_{guess}$  vorliegt, sollte diese daher vor dem Start des Algorithmus appliziert werden (Gleichung 4.13). NEUGEBAUER (1997) berechnen  $T_{guess}$  aus einer interaktiven Eingabe des Anwenders, bei der dieser korrespondierende Bereiche auswählt. Einen automatisierten Ansatz präsentieren FELDMAR & AYACHE (1994), bei dem die Krümmung der Punktwolken zur

---

Ausrichtung herangezogen wird. Für Anwendungen mit Robotern als handhabende Kinematik kann die Transformation zwischen den Punktwolken mit Hilfe der Pose des Roboters bestimmt werden (SURMANN et al. 2003; SHI & XI 2008). In diesem Fall ist die erreichbare Präzision abhängig von der Pose- und Wiederholgenauigkeit des Roboters (Abschnitt 2.2.1).

$$Y_0 = T_{guess} \cdot Y \quad (4.13)$$

### **Auswahl von Punkten**

Während BESL & MCKAY (1992) sämtliche verfügbaren Punkte nutzen, werden die Daten meist auf eine Untermenge der ursprünglichen Punktwolke reduziert, um die Suche nach Korrespondenzen weniger umfangreich zu gestalten (WECKENMANN 2012, S. 232). TURK & LEVOY (1994) nutzen hierfür eine gleichmäßig verteilte Teilmenge der Punktwolke. MASUDA et al. (1996) wählen die Punkte zufällig aus, wobei die Auswahl in jeder Iteration erneut durchlaufen wird. Alternativ kann die stochastische Auswahl einmalig geschehen (PULLI 1999). RUSINKIEWICZ & LEVOY (2001) erhalten eine charakteristische Teilmenge durch die gezielte Suche in Bereichen mit stark variierender Normalenrichtung der Punktwolke. Zusätzlich werden für eine stabilere Bestimmung der Rotation wenige, weit verstreute Punkte in ebenen Bereichen ausgewählt.

Liefert der 3D-Sensor neben dreidimensionalen Ortsinformationen zusätzlich einen Helligkeitswert, kann die Punktauswahl sich am Intensitätsgradienten orientieren (WEIK 1997). Für Punktwolken mit Farbinformation kann der von DRUON et al. (2006) vorgestellte „Color Constrained ICP“ genutzt werden, in dem sich die Auswahl auf im Vorhinein spezifizizierte Farbbereiche beschränkt. Um die Berechnungszeit unabhängig von der konkreten Methode zur Punktauswahl zu verkürzen, wird die Anzahl der Punkte zunächst niedrig gewählt und anschließend mit jeder Iteration sukzessive gesteigert (JOST & HÜGLI 2002; TURK & LEVOY 1994; ZHANG 1994).

### Bestimmung von Korrespondenzen

Zur Bestimmung von Korrespondenzen wird die Abstandsmetrik  $D_{ICP}$  genutzt, um potentiell zugehörige Punkt-Paare zu finden. BESL & MCKAY (1992) nutzen hierfür die euklidische Norm im  $\mathbb{R}^3$  zwischen zwei Punkten (Punkt-zu-Punkt-Methode). Diese Suche nach Punkt-Paaren mit minimalem Abstand kann durch die Nutzung von *k-d-Bäumen* beschleunigt werden (SIMON 1996; ZINSSER et al. 2003). Zur Speicherung von Punkten mit einer beliebigen Anzahl von Dimensionen werden in diesen Datenstrukturen schrittweise gleichmäßige Untermengen gebildet, wodurch effiziente Bereichsanfragen ermöglicht werden (KLEIN 2005, S. 126).

Ein durch CHEN & MEDIONI (1991) eingeführtes Verfahren nutzt hingegen die Normale des betreffenden Punktes, welche aus dem Punkt selbst und dessen Umgebung berechnet wird. Deren Schnittpunkt mit der Referenz-Punkt wolke wird als Korrespondenz (Punkt-zu-Ebene-Methode) gewählt. In Experimenten von RUSINKIEWICZ & LEVOY (2001) & BOUAZIZ et al. (2013) konvergiert der ICP mit diesem Verfahren schneller und durchläuft weniger Iterationen als die Variante auf Basis des minimalen euklidischen Abstands.

Neben diesen hauptsächlich eingesetzten Metriken existiert eine Reihe weiterer Vorschläge zur Bildung von Korrespondenzen. FELDMAR & AYACHE (1994) und SHARP et al. (2002) stellen Methoden vor, bei der in strukturierten Szenen invariante Features einbezogen werden. Für SHARP et al. (2002) hat diese Art der Korrespondenzbildung vor allem dann Vorteile, wenn keine initiale Transformation abgeschätzt werden kann. MINGUEZ et al. (2005) erweitern die ursprüngliche euklidische Metrik um einen rotatorischen Abstand, beschränken sich jedoch auf zweidimensionale Probleme. Eine weitere Möglichkeit besteht in der Projektion des Punktes auf die Referenz-Punkt wolke in Blickrichtung des aufnehmenden Sensors (NEUGEBAUER 1997). Falls wiederum Farbinformationen zur Verfügung stehen, kann eine gewichtete Norm aus Farbunterschied und räumlichem Abstand als Metrik genutzt werden (JOHNSON & BING KANG 1999; JOUNG et al. 2009). Um ungleichverteiltes Rauschen in Messdaten zu beherrschen, nutzen HANSEN et al. (2007) sowie MAIER-HEIN et al. (2012) die Mahalanobis-Distanz (ECKEY et al. 2002, S. 213) anstelle der euklidischen Metrik.

---

## Gewichtung und Zurückweisung

Die gefundenen Korrespondenzen können gewichtet und ungeeignete Punkt-Paare zurückgewiesen werden, um die Robustheit des Algorithmus zu steigern. Beide Schritte sind nicht in der ursprünglichen Variante von BESL & MCKAY (1992) enthalten. GODIN et al. (1994) erweitern das ICP-Verfahren um eine Gewichtung abhängig vom Abstand des betrachteten Punkte-Paars, TURK & LEVOY (1994) & ZHANG (1994) hingegen priorisieren die Korrespondenzen auf Basis lokal differierender Messunsicherheiten innerhalb der Punktwolke.

Um potentiell ungültige Korrespondenzen zu filtern, nutzt der „Trimmed ICP“ für die Registrierung lediglich einen definierten Prozentsatz der gefundenen Korrespondenzen, beginnend mit dem kleinsten Abstand (CHETVERIKOV et al. 2002). Ein weiteres Kriterium für die Güte der Punkt-Paare kann deren Lage sein. TURK & LEVOY (1994) und PULLI (1999) weisen Punkte am Rand der Punktwolke zurück, da diese mit hoher Wahrscheinlichkeit keine gültige Korrespondenz besitzen oder sich bei der Bildung von Punkt-Paaren Mehrdeutigkeiten ergeben (Abbildung 4.3.a). Mehrdeutige Zuweisungen können alternativ durch das alleinige Zulassen des Punkte-Paares mit dem geringsten Abstand unterdrückt werden (ZINSSER et al. 2003). Eine weitere Methode zur Filterung ist das Definieren eines Grenzwerts für die Distanz korrespondierender Punkte, wobei Punkt-Paare mit größerem Abstand wie in Abbildung 4.3.b zurückgewiesen werden (MASUDA et al. 1996). Dieser Grenzwert kann einmalig festgelegt werden (NEUGEBAUER 1997; SEGAL et al. 2009) oder sich pro Iteration dynamisch anpassen (JOHNSON & BING KANG 1999; RIDENE & GOULETTE 2009). Um fehlerhafte Messdaten innerhalb der Punktwolke zu erkennen und aussortieren zu können, verwenden RIDENE & GOULETTE (2009) den „Random Sample Consensus“ (RANSAC)-Algorithmus (SZELISKI 2011, S. 318) zum Filtern der Punktwolke. Als weiteren Lösungsansatz hierfür schlagen ESTÉPAR et al. (2004) die Verwendung von Kovarianzmatrizen vor.

In einer vergleichenden Studie von RUSINKIEWICZ & LEVOY (2001) wird der Einfluss einer Gewichtung generell als gering und stark abhängig vom konkreten Anwendungsfall beschrieben. Die Zurückweisung potentiell ungültiger Punkt-Paare hingegen wirkt sich stabilisierend auf das Konvergenzverhalten des ICP aus, wobei der Berechnungsaufwand als sehr gering eingeschätzt wird.

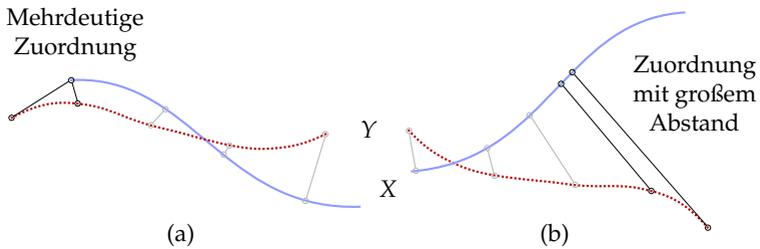


Abbildung 4.3: Durch mehrdeutige Zuordnungen im Randbereich (a) und große Abstände der Punkte-Paare (b) können unplausible Korrespondenzen entstehen.

### Fehlermaß und Minimierung

Mit Hilfe des Fehlermaßes wird die Abweichung von einer perfekten Überlagerung der Punktwolken zueinander quantifiziert. BESL & MCKAY (1992) verwenden hierfür den Mittelwert der quadratischen Abstände sämtlicher Punkte-Paare. Statt des Mittelwerts kann der Median aller quadratischen Abstände genutzt werden, um den Einfluss fehlerhafter Messdaten abzuschwächen (MASUDA & YOKOYA 1995; TRUCCO et al. 1999). Zur Berechnung des Abstands wird eine separate Metrik verwendet. Analog zur Bestimmung von Korrespondenzen mittels  $D_{ICP}$  können hierbei weitere Modifikationen eingefügt werden, um zusätzliche Informationen (z. B. Farbwerte) einzubeziehen.

Zur Minimierung des Fehlermaßes stehen die in Abschnitt 4.1.1 aufgezeigten Methoden zur Verfügung. Der ICP-Algorithmus nach BESL & MCKAY (1992) bricht ab, sobald in zwei aufeinanderfolgenden Iterationen keine relevante Verbesserung des Fehlermaßes erreicht werden kann. Weiterhin ist der direkte Vergleich des Fehlermaßes mit einem Grenzwert möglich, bei dessen Unterschreitung das Verfahren endet (SHI & XI 2008). Falls der Algorithmus nicht konvergiert, wird keines dieser Abbruchkriterien erreicht. Für praktische Anwendungen wird daher zusätzlich eine maximale Anzahl möglicher Iterationen festgelegt (DRUON et al. 2006).

---

### **4.1.3 Anwendung in der Produktionstechnik**

BREITBACH (2015) verwendet die datengetriebene Registrierung für die Modellierung von Roboterzellen. Bestehende Aufbauten werden dafür digitalisiert und mit Modellen der Komponenten verglichen. Durch eine darauf aufbauende Aktualisierung des Zellaufbaus wird die nachfolgende Offline-Programmierung der Industrieroboter ermöglicht.

Im Bereich der Fertigungsmesstechnik werden überlagernde Punktwolken durch die Nutzung externer Positionsdaten grob ausgerichtet und anschließend mit Hilfe des ICP-Algorithmus zueinander registriert (WECKENMANN 2012, S. 232). SHI & XI (2008) und WU et al. (2015) beschreiben Verfahren zur Rekonstruktion von Oberflächen, um deren Qualität zu beurteilen. Das Prüfobjekt wird dabei aus verschiedenen Blickwinkeln aufgenommen und die resultierenden Punktwolken durch die Standard-Variante des ICP-Algorithmus verknüpft. Dabei wird von Prüfobjekten ausgegangen, deren Oberflächen gut unterscheidbare und damit eindeutig identifizierbare Merkmale aufweisen.

## **4.2 Posen- und Bahnplanung von Robotern**

Als Bahnplanung wird die Interpretation einer abstrakt gestellten Aufgabe verstanden, sodass durch das Robotersystem ausführbare Bewegungsanweisungen entstehen, deren Abarbeitung die Ausführung dieser Aufgabe bedeuten (CHOSET et al. 2005, S. 1). Die einzelnen Teilschritte bestehen dabei aus Start- und Endpunkten, welche durch kollisionsfreie Pfade verbunden werden (GUPTA 1998, S. 3).

### **4.2.1 Beschreibung im Konfigurationsraum**

Für die Bearbeitung einer spezifizierten Aufgabe ist meist die Position und Orientierung des Roboter-Arbeitspunkts entscheidend. Um dabei Kollisionsfreiheit sicherzustellen, müssen jedoch sämtliche physischen Elemente des Roboters betrachtet werden. Eine eindeutige Pose eines Industrieroboters lässt sich durch

die Winkelstellungen  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$  der  $n$  voneinander unabhängigen Gelenke beschreiben und nach Gleichung 4.14 in einer Konfiguration  $\mathbf{q} \in \mathbb{R}^n$  zusammenfassen (LATOMBE 1991, S. 7).

$$\mathbf{q} = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_n \end{bmatrix} \quad (4.14)$$

Der Konfigurationsraum  $\mathcal{Q} \subset \mathbb{R}^n$  fasst die Menge aller möglichen Konfigurationen zusammen (CHOSSET et al. 2005, S. 40). Eine eindeutige Pose des Robotersystems im Arbeitsraum  $\mathcal{W} \subset \mathbb{R}^3$  kann somit als Punkt im Konfigurationsraum  $\mathcal{Q}$  abgebildet werden, eine stetige Kurve in  $\mathcal{Q}$  entspricht einer PTP-Bewegung des Roboters (Abschnitt 2.2.1). Ein Pfad zwischen zwei Posen  $\mathbf{q}_{start}$  und  $\mathbf{q}_{end}$  kann demnach als stetige Abbildung  $\psi : [0, 1] \rightarrow \mathcal{Q}$  mit  $\psi(0) = \mathbf{q}_{start}$  und  $\psi(1) = \mathbf{q}_{end}$  beschrieben werden (LATOMBE 1991, S. 10).

Ein Hindernis des Arbeitsraums  $\mathcal{O}_i \subset \mathcal{W}$  kann als Teilmenge  $\mathcal{Q}\mathcal{O}_i \subset \mathcal{Q}$  durch Gleichung 4.15 im Konfigurationsraum modelliert werden (LAVALLE 2006, S. 155).  $R(\mathbf{q})$  entspricht dabei dem Bereich, den der Roboter im Arbeitsraum  $\mathcal{W}$  einnimmt. In den enthaltenen Posen  $\mathbf{q}_{\mathcal{Q}\mathcal{O}_i} \in \mathcal{Q}\mathcal{O}_i$  würde der Roboter mit dem Hindernis  $\mathcal{O}_i$  kollidieren.

$$\mathcal{Q}\mathcal{O}_i = \{\mathbf{q} \in \mathcal{Q} \mid R(\mathbf{q}) \cap \mathcal{O}_i \neq \emptyset\} \quad (4.15)$$

Werden sämtliche Hindernisse modelliert, kann der freie Konfigurationsraum  $\mathcal{Q}_{free}$  nach Gleichung 4.16 berechnet werden (CHOSSET et al. 2005, S. 43). Darin enthaltene Posen kann der Roboter ohne Kollisionen mit sich oder einer Umweltkomponente einnehmen.

$$\mathcal{Q}_{free} = \mathcal{Q} \setminus \bigcup_i \mathcal{Q}\mathcal{O}_i = \left\{ \mathbf{q} \in \mathcal{Q} \mid R(\mathbf{q}) \cap \left( \bigcup_i \mathcal{O}_i \right) = \emptyset \right\} \quad (4.16)$$

---

Für gegebene Start- und Zielpose besteht die Aufgabe der Bahnplanung daher in der Suche nach einer stetigen Kurve im freien Konfigurationsraum  $\mathcal{Q}_{free}$ , die beide Konfigurationen verbindet (BOHLIN 1999, S. 2). Im Arbeitsraum  $\mathcal{W}$  entspricht dies einer kollisionsfreien Trajektorie des Roboters zwischen Start- und Zielpose.

#### 4.2.2 Kollisionsdetektion als Flaschenhals

Kollisionsdetektoren beantworten die Frage ob, wann und wo sich in einer simulierten Szene zwei oder mehr Körper berühren. Die Kollisionsprüfung ist fundamentaler Bestandteil von Physiksimulationen, Computerspielen und Virtual-Reality-Anwendungen, aber auch der Robotik. Meist muss die Antwort des Systems dabei in Echtzeit erfolgen. (ERICSON 2005, S. 1)

Wie in Abbildung 4.4 dargestellt, wird die Kollisionsdetektion zur Steigerung der Effizienz in eine Grob- und Feinphase aufgeteilt (HUBBARD 1993). In der Grobphase wird zunächst die Lage der Körper zueinander abgeschätzt. Dafür erhalten komplexe Körper der Szene primitive Hüllgeometrien, die jeweils paarweise auf eine Überlagerung getestet werden. Nur wenn eine Überschneidung vorliegt, ist eine weitere Prüfung in der Feinphase nötig. Die Effizienzsteigerung der Grobphase ist durch „Sweep and Prune“-Algorithmen möglich (COHEN et al. 1995). Die zu prüfenden Körper werden dabei anhand ihrer räumlichen Lage vorsortiert, um die notwendigen Berechnungsschritte zu reduzieren (BARAFF & WITKIN 1992).

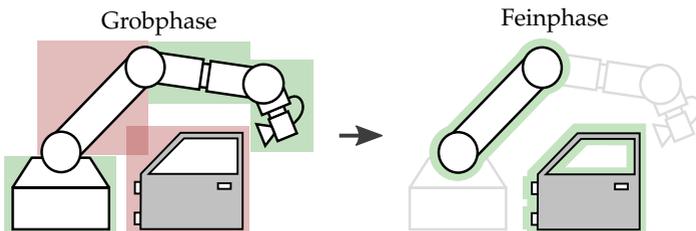


Abbildung 4.4: Durch eine vorgelagerte Grobphase mit Hüllgeometrien verringert sich die Anzahl zu testender Komponenten in der Feinphase.

Wird eine Überschneidung der Hüllgeometrien festgestellt, findet eine Prüfung der entsprechenden Körper anhand ihrer exakten Geometrien in der Feinphase statt. Über Optimierungsverfahren wird hierfür der minimale euklidische Abstand der Geometrien gesucht. Dazu ist das Finden der am nächsten gelegenen Punkte nötig, was im Verfahren nach M. C. LIN & CANNY (1991) für konvexe Polyeder mittels Voronoi-Regionen (ERICSON 2005, S. 69) realisiert wird. Die Herausforderung in der Feinphase besteht in der korrekten Prüfung konkaver Körper, bei denen lokale Minima auftreten können, die vom globalen Minimum abweichen (ERICSON 2005, S. 383).

Kollisionsdetektoren sind fundamentaler Bestandteil von stichprobenbasierten Bahnplanungsverfahren. Der Kollisionsdetektor wird dabei als „Black Box“ verwendet, um das Robotersystem in einer spezifischen Pose zu prüfen (LAVALLE 2006, S. 185). Ein für die industrielle Robotik relevantes Unterscheidungsmerkmal von Kollisionsdetektoren ist dabei die Fähigkeit, Selbst-Kollisionen der Kinematik erkennen zu können (KOCKARA et al. 2007).

Der Kollisionsdetektor testet bei jedem Aufruf eine eindeutige Roboter-Pose. Soll eine Trajektorie der Kinematik geprüft werden, ist diese zunächst in einzelne Posen aufzuteilen, der Diskretisierungsabstand definiert dabei die minimalen Abmessungen der detektierbaren Hindernisse. Je kleiner demnach die modellierten Hindernisse, desto öfter wird der Kollisionsdetektor aufgerufen. Im Gegensatz zu vielen anderen Anwendungen besteht in der Offline-Bahnplanung keine Notwendigkeit für eine Prüfung in Echtzeit. Durch die große Anzahl an Aufrufen ist der Anteil von Kollisionstests an der gesamten Planungszeit jedoch sehr hoch. Die Kollisionsdetektion wird daher als Flaschenhals des Planungsprozesses gesehen (M. LIN & GOTTSCHALK 1998).

### 4.2.3 Modellierung mittels Graphentheorie

Bahnplanungsprobleme lassen sich mit Hilfe von Graphen mathematisch beschreiben. Ein *Graph* ist ein 2-Tupel  $G = (E, K)$  mit  $K \subseteq [E]^2$ . Die Menge  $E$  enthält *Ecken*, die Menge  $K$  *Kanten* des Graphen. Eine Kante  $k \in K$  *verbindet* zwei Ecken  $e_1, e_2 \in E$  miteinander,  $e_1$  und  $e_2$  sind dann *benachbart*. Sind alle Ecken des Graphen untereinander Nachbarn, ist der Graph *vollständig verbunden*. Der *Grad* einer Ecke gibt an, wie viele Kanten von ihm ausgehen. Eine Ecke

---

heißt *isoliert*, wenn ihr Grad 0 beträgt. (DIESTEL 2006, S. 2; VOLKMANN 1996, S. 1)

Ist es relevant, ob die Kante  $k$  von  $e_1$  nach  $e_2$  verläuft oder umgekehrt, heißt der zugehörige Graph *gerichtet*, ansonsten ist der Graph *ungerichtet*. (KRUMKE & NOLTEMEIER 2012, S. 7)

Die *Gewichtung* der Kanten eines Graphen wird durch eine Abbildung der Kantenmenge auf reelle Zahlen  $\theta : K \rightarrow \mathbb{R}$  beschrieben. Die Ausprägung des Gewichts  $\theta(k)$  hängt vom Anwendungsfall ab und kann beispielsweise Wege, Zeiten oder Kosten repräsentieren. (VOLKMANN 1996, S. 22)

Ein *Weg*  $\Psi$  im Graph besteht aus einer Folge von paarweise verschiedenen Ecken und Kanten  $\Psi = (e_0, k_1, e_1, \dots, k_n, e_n)$ , wobei die Kante  $k_i$  dabei jeweils die Ecken  $e_{i-1}$  und  $e_i$  verbindet. Sind die erste und letzte Ecke eines Weges identisch  $e_0 \equiv e_n$ , handelt es sich um einen *Kreis*. Existiert zwischen jedem Ecken-Paar eines Graphen mindestens ein Weg, ist dieser *zusammenhängend*. (KRUMKE & NOLTEMEIER 2012, S. 31; DIESTEL 2006, S. 11)

Findet sich ein Kreis, der jede Kante des Graphen genau einmal durchläuft, spricht man von einem *Eulerschen Kreis* (KRUMKE & NOLTEMEIER 2012, S. 44). Der zugehörige Graph ist *eulersch*. Analog werden Kreise, die jede Ecke einmal berühren, als *Hamilton-Kreise* und die zugehörigen Graphen als *hamiltonisch* bezeichnet (KRUMKE & NOLTEMEIER 2012, S. 50).

Ist ein Graph zusammenhängend, enthält jedoch keine Kreise, wird er als *Baum* bezeichnet. Zwischen zwei beliebigen Ecken eines Baumes existiert genau ein Weg. (BRANDSTÄDT 1994, S. 21)

*Spannende Bäume* (auch *Spannbäume* genannt) sind Teilgraphen, die sämtliche Ecken des Graphen, aber keine Kreise enthalten und zusätzlich zusammenhängend sind (VOLKMANN 1996, S. 41). Wird dabei die Summe der Kantengewichte minimiert, handelt es sich um einen *minimalen Spannbaum* wie in Abbildung 4.5.a dargestellt (DIESTEL 2006, S. 4). Minimale Spannbäume können durch die Algorithmen nach Prim und Kruskal erzeugt werden (HOROWITZ et al. 1998, S. 218). Beide sind mathematisch vollständig, finden demnach garantiert eine Lösung, sofern diese existiert (HUSSMANN 2015, S. 67). Welcher

## 4 Stand der Forschung und Technik

Algorithmus dabei die effizientere Laufzeit aufweist, hängt von der Anzahl an Ecken  $|E|$  und Kanten  $|K|$  im Graph  $G = (E, K)$  ab.

Ein *Matching* ist eine Teilmenge der Ecken und Kanten eines Graphen, sodass je zwei verschiedene Kanten keine gemeinsame Ecke besitzen. Für ein *perfektes Matching* dürfen keine isolierten Ecken verbleiben. (BRANDSTÄDT 1994, S. 149)

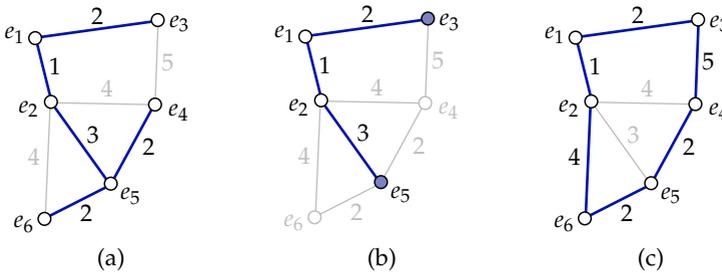


Abbildung 4.5: Graph mit minimalem Spannbaum (a), kürzestem Weg zwischen den Ecken  $\{e_3, e_5\}$  (b) und Hamilton-Kreis mit minimaler Distanz (c).

### Suche nach kürzesten Wegen

Häufig wird der optimale Weg  $\Psi$  zwischen zwei Ecken  $\{e_{start}, e_{end}\}$  eines Graphen gesucht (Abbildung 4.5.b). Dabei wird die *Distanz*  $\Theta(\Psi)$  minimiert, welche nach Gleichung 4.17 der Summe der Kantengewichte des Weges entspricht (KRUMKE & NOLTEMEIER 2012, S. 169).

$$\Theta(\Psi) := \sum_{k \in \Psi} \theta(k) \quad (4.17)$$

Sind Start- und Zielecke identisch  $e_{start} \equiv e_{end}$ , ergibt sich die Distanz  $\Theta(\Psi) = 0$ . Existiert kein Weg zwischen den Ecken  $e_{start}$  und  $e_{end}$ , ist  $\Theta(\Psi) = \infty$  (KRUMKE & NOLTEMEIER 2012, S. 171). Für nicht-triviale Fälle sind hingegen Suchalgorithmen nötig. Bei kleinen Graphen mit wenigen Ecken und Kanten kann hierfür auf die *Enumeration* zurückgegriffen werden. Dabei werden sämtliche Weg-Kombinationen getestet, wodurch die Laufzeit mit der Größe des Graphen

---

exponentiell ansteigt (HUSSMANN 2015, S. 4). Zielführender für komplexe Probleme ist der Einsatz etablierter Methoden wie dem Algorithmus nach Dijkstra oder dessen Weiterentwicklung, dem A\*-Algorithmus (LAVALLE 2006, S. 37).

Voraussetzung zur Durchführung der Wegsuche mittels **Dijkstra-Algorithmus** ist ein gewichteter Graph mit nicht-negativen Kantengewichten. Zusätzlich wird für jede Ecke des Graphen die Distanz zur Startecke sowie der direkte Vorgänger gespeichert. Die Startecke selbst erhält die *permanente* Distanz  $\Theta(e_{start}, e_{start}) = 0$  und gilt zu Beginn als *aktiv*, die verbleibenden Ecken des Graphen werden mit  $\Theta(e) = \infty$  initialisiert. (KRUMKE & NOLTEMEIER 2012, S. 177)

Ausgehend von der aktiven Ecke werden die Distanzen zu allen benachbarten Ecken berechnet, deren Distanz nicht permanent ist. Dazu wird das Gewicht der verbindenden Kante zur Distanz der aktiven Ecke addiert. Die jeweilige Nachbardecke erhält die neue Distanz, falls diese kleiner als die bestehende ist. Zusätzlich wird in diesem Fall die aktive Ecke als Vorgänger gespeichert. Die Nachbardecke mit der kleinsten Distanz wird anschließend als permanent und aktiv markiert. Die beschriebenen Teilschritte werden nun mit dieser neuen aktiven Ecke durchlaufen. Der Algorithmus endet, sobald sämtliche Ecken eine permanente Distanz erhalten haben. Falls nur eine einzelne Zielecke betrachtet wird, kann der Algorithmus bereits abbrechen, sobald diese aktiv gesetzt wird. Der Weg kann aus den gespeicherten Vorgängern, ausgehend von der Zielecke, nachvollzogen werden. Die Wegsuche mittels Dijkstra-Algorithmus findet garantiert die optimale Lösung. (HUSSMANN 2015, S. 32; KRUMKE & NOLTEMEIER 2012, S. 177)

Der **A\*-Algorithmus** verwendet eine Heuristik, um den kürzesten Weg potentiell schneller als der Dijkstra-Algorithmus zu finden. Die Heuristik schätzt dabei die Distanz bis zum Ziel in möglichst kurzer Zeit und wird anschließend eingesetzt, um die Richtung der Suche zu beeinflussen. Der Grundgedanke dabei ist, dass die Richtung mit minimaler abgeschätzter Distanz zum Ziel mit höherer Wahrscheinlichkeit den gesuchten optimalen Weg enthält. Die Vorgehensweise gleicht dem Dijkstra-Algorithmus, ausgenommen die Auswahl der nächsten aktiven Ecke  $e$ . Neben der Distanz  $\Theta(e_{start}, e)$  wird hierbei die geschätzte Entfernung zur Zielecke  $\hat{v}(e, e_{end})$  herangezogen. Als Entscheidungskriterium gilt die Summe  $\Theta(e_{start}, e) + \hat{v}(e, e_{end})$ . Die Heuristik  $\hat{v}(e, e_{end})$  darf

## 4 Stand der Forschung und Technik

---

dabei die exakte Entfernung zum Ziel  $v(e, e_{end})$  nach Gleichung 4.18 niemals überschätzen. (LAVALLE 2006, S. 37)

$$\hat{v}(e, e_{end}) \leq v(e, e_{end}) \quad \forall e \quad (4.18)$$

Bezogen auf praktische Anwendungen ist die direkte Verbindung ohne Beachtung von Hindernissen meist eine adäquate Heuristik. Für den praktischen Einsatz ist jedoch abhängig von Anwendungsfall und Implementierung zu prüfen, ob der A\*-Algorithmus gewinnbringend eingesetzt werden kann, oder die zusätzliche Laufzeit zur Abschätzung der Entfernung den Suchalgorithmus insgesamt verlangsamt (LAVALLE 2006, S. 37).

### Das „Travelling Salesman Problem“

Das „Travelling Salesman Problem“ (TSP) (dt. Problem des Handlungsreisenden) ist ein kombinatorisches Optimierungsproblem. In der ursprünglichen Form besteht die Aufgabe darin, eine möglichst kurze Rundreise durch eine festgelegte Anzahl an Städten zu finden. Die erste bekannte Erwähnung des Problems findet sich in einem Reisehandbuch für Mitteldeutschland von 1832 (APPLEGATE et al. 2011, S. 3) mit dem Titel:

Der Handlungsreisende - Wie er sein soll und was er zu thun hat,  
um Aufträge zu erhalten und eines glücklichen Erfolgs in seinen  
Geschäften gewiß zu sein - Von einem alten Commis-Voyageur.

Ein praktisches Anwendungsbeispiel für die noch immer aktuelle Relevanz dieses Optimierungsproblems ist die Produktion von Leiterplatten, bei der die Reihenfolge von Bohrungen optimiert wird, um die Taktzeit zu minimieren. (HUSSMANN 2015, S. 106; BÜSING 2010, S. 78)

Das TSP kann mit Hilfe der Graphentheorie modelliert werden. Es gilt dann, einen möglichst kurzen Kreis  $\Psi_{HK}$  in einem vollständigen Graphen zu finden, der sämtliche Ecken einmal berührt. Dies entspricht der Suche nach einem Hamilton-Kreis mit minimaler Distanz  $\Theta(\Psi_{HK})$ , welche sich analog zu Gleichung 4.17 berechnet. Ist der Graph dabei gerichtet, spricht man von einem

---

*asymmetrischen* TSP. Die errechnete Lösung gilt dann nur für die angegebene Richtung. Bei *symmetrischen* Problemstellungen ist die Richtung nicht relevant, der Graph wird daher ungerichtet aufgebaut. (HUSSMANN 2015, S. 96)

Die optimale Rundreise wie in Abbildung 4.5.c kann durch Berechnung aller Hamilton-Kreise des Graphen gefunden werden. Für diese Enumeration ergeben sich für ein asymmetrisches TSP  $(n - 1)!$ , für symmetrische  $\frac{1}{2}(n - 1)!$  Möglichkeiten (BÜSING 2010, S. 80). Für praktische Anwendungen mit mehreren hundert Ecken im Graph ist diese Methode daher nicht effizient umsetzbar. Gleiches gilt bisher für sämtliche Ansätze mit garantiert optimaler Lösung (HUSSMANN 2015, S. 114; APPLGATE et al. 2011, S. 525).

Es wird daher auf Heuristiken zurückgegriffen, deren Laufzeit um ein Vielfaches kleiner ist, die jedoch nicht zwangsweise den Hamiltonschen Kreis mit minimalem Kantengewicht finden. Es existiert eine Vielzahl derartiger Approximationsalgorithmen zur Lösung des TSP (REINELT 1994). Für praktische Anwendungen sind vor allem Algorithmen interessant, die eine definierte Güte im Vergleich zur optimalen Lösung garantieren.

Eine dieser Methoden ist die **Doubletree-Heuristik**, bei der zunächst ein minimaler Spannbaum mit Hilfe der Algorithmen nach Prim oder Kruskal gebildet wird. Es folgt die namensgebende Verdopplung der Kanten, um einen eulerschen Graphen zu erhalten. Im anschließenden Schritt wird ein Eulerscher Kreis gebildet und ausgehend von der Startecke durchlaufen. Dabei wird dem Kreis jede Ecke hinzugefügt, die noch nicht berührt wurde. Trifft der Algorithmus auf eine bereits besuchte Ecke, wird diese durch die direkte Verbindung zur nachfolgenden Ecke ersetzt. Das Ergebnis ist ein Hamiltonscher Kreis. (REINELT 1994, S. 90; BÜSING 2010, S. 82)

Die Doubletree-Heuristik hat eine Güte von 1, der zurückgegebene Kreis ist demnach maximal 100% länger als die optimale Lösung. Dies gilt allerdings nur für *metrische* Graphen. Ein Graph ist metrisch, wenn für alle Ecken die Dreiecksungleichung erfüllt wird. Es muss demnach stets Gleichung 4.19 gelten, das Kantengewicht einer direkten Verbindung zwischen zwei Ecken  $\{e_i, e_j\}$  darf niemals größer sein als die Summe der Kantengewichte bei einem Umweg über eine dritte Ecke  $e_m$ . (BÜSING 2010, S. 81)

$$\theta(e_i, e_j) \leq \theta(e_i, e_m) + \theta(e_m, e_j) \quad (4.19)$$

Der **Algorithmus nach Christofides** ist eine häufig verwendete Modifikation der Doubletree-Heuristik mit einer Güte von 0,5. Auch hier wird zunächst ein minimaler Spannbaum gebildet. Anschließend wird ein auf das Kantengewicht bezogen minimales, perfektes Matching aller Ecken mit ungeradem Grad gebildet. Die erzeugten Kanten werden dem ursprünglichen Graphen hinzugefügt, welcher dadurch zwingend eulersch wird. Die nachfolgenden Schritte bestehen analog zur Doubletree-Heuristik aus der Suche nach einem Eulerschen Kreis und der daraus folgenden Bildung des Hamiltonschen Kreises. (REINELT 1994, S. 91; HUSSMANN 2015, S. 122)

### 4.2.4 Stichprobenbasierte Verfahren

Techniken zur Planung von Bahnen lassen sich in kombinatorische und stichprobenbasierte Methoden einteilen. Kombinatorische Verfahren lösen die gestellte Aufgabe mit Hilfe einer vollständigen und expliziten Beschreibung des zugehörigen Konfigurationsraums (CHOSET et al. 2005, S. 110). Diese Techniken liefern für einfache Probleme mit wenigen Freiheitsgraden schnelle, mathematisch optimale Ergebnisse. Der Aufwand zur Modellierung und Berechnung ist allerdings zu hoch für industriell relevante Anwendungsfälle (LAVALLE 2011).

Alternativ stehen stichprobenbasierte Methoden zur Verfügung. Hierbei wird lediglich ein Teil des Konfigurationsraums erfasst, mit dem die Lösung der Aufgabe möglich ist. Werden die Konfigurationen zufällig ausgewählt, handelt es sich um ein randomisiertes Verfahren (CHOSET et al. 2005, S. 200). Stichprobenbasierte Methoden liefern auch bei Problemen höherer Ordnung in kurzer Zeit ein Ergebnis. Dafür wird die Erfassung des Konfigurationsraums nur bis zum Finden der ersten Lösung ausgedehnt. Es kann dabei nicht ausgeschlossen werden, dass der nicht betrachtete Teil des Konfigurationsraums bessere Lösungen ermöglichen würde. Die gefundenen Ergebnisse sind daher nicht zwingend optimal. (LAVALLE 2006, S. 185)

---

Je nach Aufgabenstellung werden verschiedene Verfahren angewandt. Planungsmethoden für mehrmalige Anfragen innerhalb eines Konfigurationsraums werden als *Multi-Query-Planer* bezeichnet, die bedeutendste ist die „Probabilistic Roadmap Method“. Wird hingegen einmalig ein einzelner Pfad gesucht, werden *Single-Query-Planer* wie die Methode der „Rapidly-Exploring Random Trees“ eingesetzt. (CHOSSET et al. 2005, S. 227)

### Die „Probabilistic Roadmap Method“

Die durch KAVRAKI et al. (1996) vorgestellte „Probabilistic Roadmap Method“ (PRM) wurde für Roboter mit mehreren Freiheitsgraden innerhalb einer statischen Arbeitsumgebung entwickelt. Die namensgebende *Roadmap* (dt. Straßenkarte) repräsentiert dabei den Konfigurationsraum in Form eines Graphen. Die Methode unterteilt sich in eine Explorationsphase, in der die Roadmap  $R$  aufgebaut wird, und die anschließende Suche nach Pfaden darin.

Die Explorationsphase wiederum besteht aus zwei Schritten, dem Hinzufügen zufälliger kollisionsfreier Konfigurationen sowie dem anschließenden Verbinden dieser mit ihren nächsten Nachbarn. Innerhalb der Roadmap repräsentieren Ecken Konfigurationen des Robotersystems und Kanten kollisionsfreie Bewegungen zwischen den zugehörigen Ecken. Der prinzipielle Ablauf zum Aufbau der Roadmap  $R$  der ursprünglichen PRM ist in Algorithmus 4.2 dargestellt. Um den Algorithmus anzuwenden, muss zunächst die Anzahl der zu erstellenden Zufallskonfigurationen  $n$  sowie die Art der Generierung (Zeile 6) festgelegt werden. KAVRAKI et al. (1996) erzeugen die Zufallskonfigurationen mit Hilfe einer Gleichverteilung innerhalb der zulässigen Wertebereiche der Robotergelenke.

Zusätzlich muss die Anzahl der zu testenden Nachbarn  $i$  sowie die zugehörige *Abstandsmetrik*  $D_{PRM}(q, q')$  (Zeile 11) definiert werden. Diese legt fest, wie die Entfernung zwischen den Konfigurationen  $q$  und  $q'$  berechnet wird. Sie beeinflusst daher, welche Konfigurationen als nächste Nachbarn gelten und potentiell verbunden werden. KAVRAKI et al. (1996) verwenden hierfür den maximalen euklidischen Abstand des Roboters zwischen den beiden Konfigurationen  $q$  und  $q'$ . Für die Berechnung mittels Gleichung 4.20 werden sämtliche Punkte  $x$  des Roboters in Betracht gezogen.

## 4 Stand der Forschung und Technik

---

### Algorithmus 4.2 PRM: Aufbau der Roadmap (CHOSSET et al. 2005, S. 204)

---

```
1: procedure BUILDROADMAP( $n, i$ )
2:    $E \leftarrow \emptyset$ 
3:    $K \leftarrow \emptyset$ 
4:   while  $|E| < n$  do                                ▷ Hinzufügen zufälliger Konfigurationen
5:     repeat
6:        $q \leftarrow$  Zufallspose in  $\mathcal{Q}$ 
7:     until  $q$  ist kollisionsfrei
8:      $E \leftarrow E \cup \{q\}$ 
9:   end while
10:  for all  $q \in E$  do                                  ▷ Verbinden der nächsten Nachbarn
11:     $N_q \leftarrow i$  nächste Nachbarn in  $E$ 
12:    for all  $q' \in N_q$  do
13:      if  $\{q, q'\} \notin K$  &  $\Lambda(q, q') \neq \text{NIL}$  then
14:         $K \leftarrow K \cup \{q, q'\}$ 
15:      end if
16:    end for
17:  end for
18:  return  $R \leftarrow (E, K)$                           ▷ Ausgabe der Roadmap
19: end procedure
```

---

$$D_{PRM}(q, q') = \max_{x \in \text{Roboter}} \|x(q) - x(q')\| \quad (4.20)$$

Mit Hilfe der Abstandsmetrik werden die nächsten  $k$  Nachbarn  $N_q$  der betrachteten Zufallskonfiguration  $q$  ermittelt. Zum Verbinden wird ein *lokaler Planer*  $\Lambda(q, q')$  genutzt. Dieser erzeugt die Bewegung des Roboters zwischen den nahe beieinander liegenden Konfigurationen  $q$  und  $q'$ . Falls die erzeugte Verbindung kollisionsfrei durchführbar ist, wird die Kante  $\{q, q'\}$  der Roadmap  $R$  hinzugefügt. (CHOSSET et al. 2005, S. 203)

Auf diese Weise entsteht eine Roadmap des freien Konfigurationsraums. In der zweiten Phase werden die Zielkonfigurationen analog zur Explorationsphase der Roadmap hinzugefügt und mit Hilfe des lokalen Planers verbunden. Ist die Roadmap nun wie in Abbildung 4.6.a zusammenhängend, kann mit Hilfe konventioneller Wegsuche-Algorithmen die gesuchte Roboterbahn als Sequenz aus Konfigurationen erzeugt werden (siehe Abschnitt 4.2.3). Sind die Zielkonfigurationen in der Roadmap nicht verbunden, kann kein Weg gefunden

---

werden. Dies bedeutet allerdings nicht zwingend, dass keine Lösung existiert. Das Verfahren ist abhängig von der Quantität und Qualität der erzeugten Zufallskonfigurationen und einem lokalen Planer, der existente Verbindungen zuverlässig erkennt (BARRAQUAND et al. 2000).

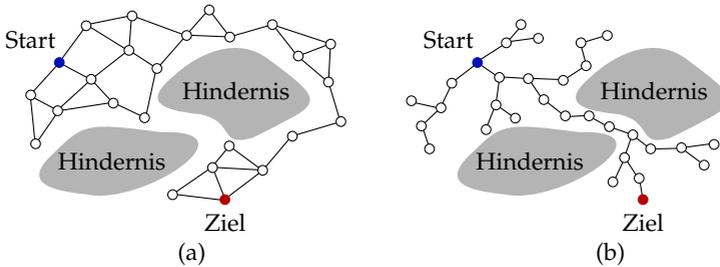


Abbildung 4.6: Die PRM erzeugt und verbindet zufällige Konfigurationen, bis ein Weg zwischen Start und Ziel existiert (a). Das RRT-Verfahren hingegen erkundet den Raum mit Hilfe von Bäumen (b).

Die PRM ist probabilistisch vollständig, die Wahrscheinlichkeit eine vorhandene Lösung nicht zu finden konvergiert mit steigender Berechnungszeit gegen Null. Dafür wird die Roadmap iterativ mit dem vorgestellten Vorgehen verdichtet, bis eine existierende Lösung gefunden werden kann. (CHOSSET et al. 2005, S. 243)

Es existieren zahlreiche Modifikationen und Erweiterungen der PRM für spezifische Problemstellungen. Einer der häufig modifizierten Aspekte ist die Generierung der Konfigurationen. Neben einer zufälligen Erzeugung der ursprünglichen Methode ist ein gitterbasierter Ansatz möglich. Das Gitter wird dabei schrittweise verfeinert, bis die gewünschte Dichte der Roadmap erreicht wurde. BRANICKY et al. (2001) schlagen alternativ ein pseudo-zufälliges Vorgehen basierend auf Hammersley-Halton-Punkten vor. Diese Verteilungsfunktionen weisen einen sehr geringen Unterschied zu wahren Zufallsverteilungen auf, sind aber dennoch deterministisch (MATOUŠEK 2009, S. 38). Durch diesen deterministischen Charakter wird eine zufällige Anhäufung von Konfigurationen in einem lokal begrenzten Bereich vermieden (BRANICKY et al. 2001). (GERAERTS & OVERMARS 2004b)

Ob die zufällige oder deterministische Erzeugung von Konfigurationen erfolgsversprechender ist, hängt vom konkreten Robotersystem, dessen Arbeits-

## 4 Stand der Forschung und Technik

---

umgebung und der gestellten Aufgabe ab. GERAERTS & OVERMARS (2004b) vertreten daher die Meinung, dass keine generelle Aussage über die Güte der Verfahren getroffen werden kann. LAVALLE et al. (2004) hingegen halten die deterministische Auswahl der Konfigurationen für überlegen und untermauern dies anhand experimenteller Studien.

Weitere Methoden konzentrieren sich auf problematische Bereiche der Roadmap wie Engstellen und Hindernisse. Konfigurationen in der Nähe eines Hindernisses sind hilfreich für das Finden von kurzen Wegen um eben jenes Hindernis herum (AMATO et al. 1998b). BOOR et al. (1999) generieren hierfür jeweils zwei Konfigurationen mit einem durch die Normalverteilung definierten euklidischen Abstand. Wenn genau eine der beiden Konfigurationen kollisionsfrei erreichbar ist, wird diese in die Roadmap übernommen.

An kritischen Konfigurationen kann die Roadmap gezielt erweitert werden, um einen zusammenhängenden Graphen zu erhalten. Ein Beurteilungskriterium, ob eine Konfiguration kritisch ist, kann die Anzahl der Verbindungen zu anderen Konfigurationen der Roadmap sein. Im schlechtesten Fall ist die Konfiguration isoliert und somit nicht erreichbar. Kritische Konfigurationen finden sich oft an Engstellen des Arbeitsraums und sind daher in besonderem Maße für die Bahnplanung relevant. (KAVRAKI & LATOMBE 1998)

Um Pfade in engen Passagen des Konfigurationsraumes zu ermöglichen, schlagen HSU et al. (2003) einen Brücken-Test vor. Dafür werden zwei Punkte mit definiertem, von einer Verteilungsfunktion abhängigen Abstand zufällig im Konfigurationsraum platziert. Führen beide Punkte zu Kollisionen, wird der Mittelpunkt betrachtet. Liegt dieser im freien Konfigurationsraum, wird er der Roadmap hinzugefügt. Für die Bahnplanung sind meist zusätzlich Konfigurationen in Bereichen ohne Hindernisse nötig, daher wird das Verfahren in Kombination mit konventionellen Methoden wie dem gitterbasierten Ansatz eingesetzt.

Empirische Analysen zeigen, dass Methoden zur gezielten Erzeugung von Konfigurationen in problematischen Bereichen die Effizienz der PRM wesentlich steigern können (HSU et al. 2006). Durch den erhöhten Aufwand zur Implementierung und die steigende Komplexität sollten diese Modifikationen laut

---

GERAERTS & OVERMARS (2004a) jedoch nur eingesetzt werden, wenn die Anwendung dies erfordert.

Den Großteil der Berechnungszeit der PRM nehmen Kollisionsprüfungen des lokalen Planers in Anspruch. Der Ansatz der „Lazy-PRM“ ist daher die Reduzierung dieser Tests, um in kürzerer Zeit eine Lösung der Bahnplanungsaufgabe zu finden. Dazu werden zufällig hinzugefügte Konfigurationen sowie deren Verbindungen zu den nächsten Nachbarn grundsätzlich als kollisionsfrei angenommen und nur dann geprüft, wenn sie im finalen Pfad vorhanden sind. Wird eine Kollision detektiert, entfernt der Algorithmus die zugehörige Ecke oder Kante und führt die Wegsuche erneut durch. Da die Roadmap eine große Anzahl an Ecken und Kanten enthält, die nicht zur Lösung der Aufgabe notwendig sind, wird die Berechnungszeit für deren Überprüfung eingespart. Die „Lazy-PRM“ ist für den Einsatz als Single-Query-Planer konzipiert. (SÁNCHEZ & LATOMBE 2003; BOHLIN & KAVRAKI 2000)

### **„Rapidly-Exploring Random Trees“**

Das Verfahren „Rapidly-Exploring Random Trees“ (RRT) ist ein Single-Query-Planer und erkundet den freien Konfigurationsraum inkrementell ausgehend vom Startpunkt mit Hilfe von Bäumen. (LAVALLE 1998; SICILIANO & KHATIB 2008, S. 114)

Der Ablauf des RRT-Verfahrens wird in Algorithmus 4.3 dargestellt, ein Beispiel ist in Abbildung 4.6.b visualisiert. Zunächst wird eine zufällige Konfiguration  $q_{rand}$  erzeugt, für welche anschließend der nächste Nachbar  $q_{near}$  im bereits bekannten Graphen bestimmt wird. Die potentiell neue zum Baum hinzuzufügende Konfiguration  $q_{new}$  liegt auf der Verbindungsstrecke von  $q_{near}$  und  $q_{rand}$  mit einem Abstand von  $s$  ausgehend von  $q_{near}$ . Ist die neue Konfiguration  $q_{new}$  kollisionsfrei, wird sie als Ecke dem Graphen hinzugefügt und mit  $q_{near}$  verbunden. Analog zum lokalen Planer der PRM beeinflusst die Schrittweite  $s$  die minimal detektierbare Hindernisgröße. In der ursprünglichen durch LAVALLE (1998) vorgestellten Form endet der Algorithmus nach  $n$  Iterationen. Werden stattdessen bis zum Finden einer Lösung stetig neue Konfigurationen hinzugefügt, ist diese Methode zur Bahnplanung probabilistisch vollständig. (LAVALLE 1998; CHOSSET et al. 2005, S. 233)

## 4 Stand der Forschung und Technik

---

### Algorithmus 4.3 RRT: Aufbau eines Baums (CHOSSET et al. 2005, S. 234)

---

```
1: procedure BUILDRANDOMTREE( $q_{start}, n, s$ )
2:    $E \leftarrow \{q_{start}\}$  ▷ Hinzufügen der Startkonfiguration
3:    $K \leftarrow \emptyset$ 
4:   for all  $n$  do
5:     repeat
6:        $q_{rand} \leftarrow$  Zufallskonfiguration in  $\mathcal{Q}$ 
7:     until  $q_{rand}$  ist kollisionsfrei
8:      $q_{near} \leftarrow$  nächster Nachbar zu  $q_{rand}$  in  $E$ 
9:      $q_{new} \leftarrow$  bewege  $q_{near}$  auf einer Geraden in Richtung von  $q_{rand}$  um  $s$ 
10:    if  $q_{new}$  ist kollisionsfrei then
11:       $E \leftarrow E \cup \{q_{new}\}$ 
12:       $K \leftarrow K \cup \{(q_{near}, q_{new})\}$ 
13:    end if
14:  end for
15:  return  $R \leftarrow (E, K)$  ▷ Ausgabe des Baumes
16: end procedure
```

---

Um die Ausbreitungsrichtung des Graphen zu steuern, kann statt einer zufälligen Konfiguration zu einem definierten Prozentsatz die Zielkonfiguration genutzt werden. Der Prozentsatz darf dabei nicht zu hoch ausfallen, da der Algorithmus sonst Gefahr läuft, in einem lokalen Minimum zu verharren. (CHOSSET et al. 2005, S. 236)

Die Methode „RRT-Connect“ versucht, die gerichtete Ausbreitung weiter zu beschleunigen. Dafür wird die Richtung nicht nach jeder Iteration neu bestimmt. Stattdessen werden fortlaufend neue Konfigurationen mit der Schrittweite  $s$  hinzugefügt, bis die Zielkonfiguration verbunden ist oder eine Kollision auftritt. (KUFFNER & LAVALLE 2000)

Ein weiterer Ansatz zur gerichteten Ausbreitung ist die simultane Planung zweier Bäume, jeweils beginnend an den Endpunkten des gesuchten Pfades. Zunächst wird das konventionelle Vorgehen in Algorithmus 4.3 genutzt, um eine neue Konfiguration  $q_{new,1}$  für den ersten Baum zu erzeugen. Diese dient dem zweiten Baum wiederum als zufällige Konfiguration  $q_{rand,2} = q_{new,1}$ , wodurch sich die Bäume tendenziell zueinander ausbreiten. Die Methode arbeitet alternierend, die Richtung geben die Bäume daher wechselseitig vor. (LAVALLE & KUFFNER 2001)

---

Im Vergleich zur PRM kann bei Single-Query-Planern wie RRT die Berechnungszeit für das Erkunden des freien Konfigurationsraums zu einem großen Teil eingespart werden. Vorteilhaft ist dies vor allem für dynamische Arbeitsumgebungen, in denen eine Roadmap innerhalb kurzer Zeit veraltet und neu berechnet werden muss. In Aufgabenstellungen mit statischer Umgebung, in der eine Vielzahl an Bahnen gefunden werden soll, amortisiert sich hingegen die Berechnungszeit zum Aufbau der Roadmap schnell und Multi-Query-Planer sind vorzuziehen. (LINDEMANN & LAVALLE 2005)

#### **4.2.5 Anwendung in der Fertigungsmesstechnik**

Die automatische Programmerstellung für Prüfsysteme, seien es Koordinatenmessmaschinen oder roboterbasierte Messsysteme, erfordert die Berücksichtigung von zwei Aspekten. Der erste umfasst die Festlegung von Position und Orientierung des Sensors, um die Prüfmerkmale aufnehmen zu können. Diese statischen Posen müssen in einem nachfolgenden Schritt zu einem Pfad im Arbeitsraum verbunden werden. (YAU & MENQ 1991)

Optische Sensoren als Messmittel benötigen spezifische Randbedingungen, um das Prüfobjekt erfolgreich digital abbilden zu können. Nur mit definiertem Abstand zur Oberfläche arbeitet der eingesetzte Sensor fokussiert und kann damit optimale Ergebnisse liefern (SCOTT et al. 2003). Weiterhin muss die Sichtlinie zum beobachteten Objekt frei bleiben, notwendige Beleuchtung darf nicht abgeschattet werden und der sichtbare Bereich muss das zu prüfende Merkmal vollständig enthalten (TARABANIS et al. 1995). Zusätzlich ist die Kollisionsfreiheit gegenüber Prüfobjekt sowie umgebenden Hindernissen in jeder Pose sicherzustellen.

Um Reflexionen zu vermeiden und eine optimale Repräsentation zu erhalten, ist ein charakteristischer Winkel des Sensors zur beobachteten Oberfläche einzuhalten (LEE & PARK 2000). Diesen Neigungswinkel stellen WU et al. (2015) sicher, indem Posen aus der Normalenrichtung der beobachteten Oberfläche abgeleitet werden. Genutzt wird dazu ein CAD-Modell des Prüfobjekts, welches die Oberfläche in Form von Dreiecken beschreibt. Die Normalenrichtung dieser Dreiecke wird außerdem genutzt, um eine Aussage darüber zu treffen, ob sämtliche Oberflächenteile im Sichtbereich des Sensors erfasst werden können. Ist

## 4 Stand der Forschung und Technik

---

die Variation der Normalenrichtung größer als der maximale Neigungswinkel des Sensors, kann keine geschlossene Repräsentation der Oberfläche erzeugt werden.

Weitere Anforderungen an die Posen ergeben sich aus den konkreten Prüfmerkmalen, sodass verwendete Methoden an die jeweilige Anwendung angepasst werden müssen (T. S. NEWMAN & JAIN 1995; TRUCCO et al. 1997). Liegt keine digitale Beschreibung des Prüfobjekts vor, können die zur Rekonstruktion notwendigen Posen dynamisch aus den vorhandenen Daten generiert werden (REED & ALLEN 2000).

Ist eine Menge möglicher Posen gefunden, können diese priorisiert werden. Falls dies durch die Anforderungen des Sensors selbst nicht inhärent gegeben ist, kann die Auswahl anhand der nachfolgenden Bahn erfolgen. MAHMUD et al. (2011) schlagen vor, die Orientierungsänderungen des Sensors zu minimieren, da hierdurch die größte Zunahme des Messfehlers zu erwarten ist.

Im Anschluss an die Planung der Posen erfolgt die Bahnplanung des Prüfsystems. ZHAO et al. (2012) testen hierfür den direkten Weg zwischen je zwei Posen. Ist dieser nicht kollisionsfrei, wird ein Ausweichpfad in konstantem Abstand zum Prüfobjekt generiert. Dazu wird eine Ebene erzeugt, die Start- und Zielpose enthält. Die Schnittkante dieser Ebene mit der Oberfläche des Prüfobjekts wird in senkrechter Richtung verschoben und ergibt den neuen Pfad. Der Ansatz beachtet allein den Sensor, die handhabende Kinematik wird nicht betrachtet.

Für die taktile Messung mit Hilfe von Koordinatenmessmaschinen beschreiben CHO et al. (2005) eine weitere Methode zur Bahngenerierung. Die Reihenfolge der Messposen wird durch das näherungsweise Lösen des TSP optimiert, um die Prüfzeit zu minimieren. Findet der Algorithmus kollisionsbehaftete Bewegungen, werden Ausweichposen generiert. Dazu entfernt sich der Messtaster senkrecht zur Oberfläche ausreichend weit vom Prüfobjekt.

### 4.3 Fazit und Handlungsbedarf

Für die datengetriebene Registrierung ist der ICP-Algorithmus das grundlegende Werkzeug zur Ausrichtung von Punktwolken. Zahlreiche Implementie-

---

rungen zeigen die Funktionalität auf, verschiedenste Varianten bieten Lösungsansätze für konkrete Einsatzszenarien. Eine universell nutzbare Version des Algorithmus existiert indessen nicht. Die einzelnen Teilschritte müssen zur Anwendung im vorliegenden Einsatzfall modifiziert werden. Das Auffinden des globalen Optimums ist mit den bisherigen Ansätzen nicht sichergestellt. Initiale Schätzungen verringern die Wahrscheinlichkeit gegen ein lokales Optimum zu konvergieren, die Detektion des globalen Optimums wird dadurch allerdings nicht garantiert.

Limitiert wird die datengetriebene Ausrichtung durch die Randbedingung, dass stets unterscheidbare Merkmale im Sichtbereich vorhanden sein müssen. Neben geometrischen Eigenschaften der Szene können hierfür Farbe und Helligkeit der Bildpunkte genutzt werden. Diese Ansätze wurden für Einsatzgebiete mit heterogenen Sichtbereichen konzipiert (z. B. Aufnahme von Häuserfassaden) und sind daher nicht direkt auf den vorliegenden Anwendungsfall übertragbar (vergleichsweise homogene Bauteile). Es existieren keine Vorschläge, wie zusätzliche Informationen eingebracht werden können, falls weder geometrische Merkmale noch Farb- oder Helligkeitsinformationen vorhanden sind.

In der Fertigungsmesstechnik werden mit Hilfe des ICP-Algorithmus bereits vorliegende Daten korrigiert. Bisher existiert kein Prüfsystem, bei dem die Registrierung allein auf der Verknüpfung von Punktwolken basiert. Des Weiteren wurden keine Untersuchungen zum Einfluss des Restfehlers bei einer seriellen Verknüpfung veröffentlicht.

Aufbauend auf einem robusten Verfahren zur Verknüpfung der Punktwolken ist es nötig, die zugehörigen Aufnahmen zu planen. Die vorgestellten Bahnplanungsmethoden sind hierfür geeignet, können allerdings aufgrund der speziellen Anforderungen von 3D-Image-Stitching nicht ohne weiteres übernommen werden. Es sind zahlreiche implementierte Varianten verfügbar, Anpassungen und Modifikationen müssen je nach konkreter Aufgabe gewählt werden. Grundsätzlich ist eine Lösung mit mehreren Methoden möglich. Es gilt das Verfahren zu finden, welches in der kürzesten Berechnungszeit das beste Ergebnis in Bezug zum gewählten Optimierungskriterium erzeugt. Die veröffentlichten Ansätze adressieren meist eine Minimierung der Fahrzeit des Roboters. Ob und wie die Bahnplanung für Prüfsysteme zur Verringerung des Messfehlers beitragen kann, wurde bisher nicht untersucht.

## 4 Stand der Forschung und Technik

---

Im Bereich der Fertigungsmesstechnik existieren Ansätze für die Pfadgenerierung von Koordinatenmessgeräten. Durch den kartesischen Aufbau und die systembedingte Vermeidung von Eigenkollisionen sind diese Methoden jedoch nicht unmittelbar auf Vertikal-Knickarmroboter übertragbar.

## 5 Registrierung durch Verknüpfung von Punktwolken

Die Aufgabe eines roboterbasierten Messsystems ist die Prüfung der Maßhaltigkeit geometrischer Merkmale eines Prüfobjekts. Dazu wird ein 3D-Sensor mit Hilfe eines Industrieroboters positioniert, sodass sich relevante Teile der Oberfläche des Prüfobjekts im Sichtfeld des Sensors befinden. Der 3D-Sensor erzeugt anschließend eine dreidimensionale Repräsentation dieses Bereichs in Form einer Punktwolke. Diese Aufnahme wird in einer Auswertesoftware verarbeitet, um die geforderten Messwerte zu erzeugen. Zuletzt werden sämtliche Ergebnisse dokumentiert und in einem Prüfbericht zusammengefasst. Die zu messenden Dimensionen der geometrischen Merkmale übersteigen dabei in der Regel das Sichtfeld des 3D-Sensors und müssen daher in ein globales Koordinatensystem überführt werden.

Um eine ausreichende Genauigkeit dieser Registrierung sicherzustellen, wird die Methode des 3D-Image-Stitching verwendet. Dabei gilt es die Punktwolken so zu verschieben und zu drehen, dass diese sich im Rahmen der Messunsicherheit perfekt überdecken. Die resultierende Transformation beschreibt die Relation der Aufnahmen untereinander. Sind sämtliche Punktwolken auf diese Weise miteinander verbunden, kann für die Messdatenauswertung ein globales Koordinatensystem festgelegt werden.

### 5.1 Anforderungen und Rahmenbedingungen

Das Verfahren des 3D-Image-Stitching nutzt den für die Messung vorgesehenen 3D-Sensor zusätzlich für die Registrierung der Aufnahmen. Ein Vertikal-Knickarmroboter positioniert den 3D-Sensor, weist dabei jedoch Fehler in der Positionierung im Millimeterbereich auf. Diese werden durch die vorgestellte

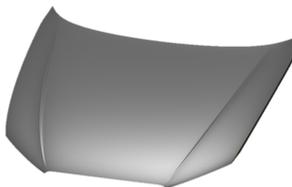
## 5 Registrierung durch Verknüpfung von Punktwolken

---

Methode erkannt und in der Messdatenauswertung kompensiert. Das Vorgehen bringt neue Herausforderungen mit sich, welche für eine erfolgreiche Implementierung beachtet werden müssen.

Der Prozess zur Prüfung geometrischer Merkmale eines Prüfobjekts gibt dabei die Rahmenbedingungen vor. Zunächst zählt hierzu der eingesetzte 3D-Sensor. Dieser erstellt als in sich geschlossene Messeinheit ein Abbild des beobachteten Ausschnitts des Prüfobjekts. Eine perfekte Abbildung des realen Prüfobjekts wird dabei durch systematische und stochastische Abbildungsfehler verhindert. Als nutzbare Daten für die Verknüpfung steht eine Punktwolke zur Verfügung. Neben diesen Daten des 3D-Sensors kann die Selbstlokalisierung des Roboters durch Vorwärtstransformation (Abschnitt 2.2.1) genutzt werden. Diese ist mit einer Messunsicherheit im Millimeterbereich beaufschlagt, bietet jedoch bereits eine Näherungslösung, auf der nachfolgende Algorithmen zur Registrierung aufbauen können.

Die Ausrichtung der Punktwolken kann dabei nur anhand von markanten Details (geometrische Konturen) in den Punktwolken erfolgen. Eingesetzt werden roboterbasierte Messsysteme hauptsächlich in der Automobilindustrie, ferner im Flugzeug- und Turbinenbau (LETTENBAUER 2002; LUDWIG 2011). Die typischerweise untersuchten Prüfobjekte besitzen dabei potentiell großflächige homogene Oberflächen, welche lediglich durch die zu messenden Merkmale unterbrochen werden (Abbildung 5.1). Bei derartigen homogenen Oberflächen existiert keine eindeutige Lösung für die lokale Ausrichtung. Es ist daher nicht sichergestellt, dass das Prüfobjekt grundsätzlich die notwendigen Details für die angestrebte Registrierung aufweist.



*Abbildung 5.1: Frontklappe eines Pkw als beispielhaftes Prüfobjekt. Großflächige homogene Bereiche erschweren eine datengetriebene Registrierung.*

---

Eine weitere Herausforderung ergibt sich aus dem unvermeidbaren Messfehler bei der Aufnahme durch den 3D-Sensor. Die Sensoreinheit liefert eine Punktwolke als dreidimensionale Repräsentation der im Sichtfeld des Sensors befindlichen Oberfläche des Prüfobjekts. Die Tiefeninformation der Bildpunkte wird dabei sensorintern berechnet, der 3D-Sensor wird vom Hersteller vor Auslieferung kalibriert. Durch Verzeichnung bedingte Abbildungsfehler werden daher bestmöglich kompensiert. Es verbleibt allerdings ein unvermeidlicher Restfehler, da jede aufgenommene Punktwolke die Realität nur mit begrenzter Genauigkeit repräsentieren kann. Eine auf diesen Daten basierende Verknüpfung ist demnach ebenfalls per se nicht perfekt. Da weiterhin mehrere Punktwolken hintereinander verknüpft werden, steigt das Fehlerpotential, je mehr Aufnahmen benötigt werden. Während des Prozessablaufs muss sichergestellt werden, dass der dabei entstehende Fehler den für die Prüfmittelfähigkeit erforderlichen Grenzwert nicht überschreitet.

## 5.2 Konzept zur Registrierung durch Verknüpfung

Die Registrierung erfolgt datengetrieben, dabei werden die durch den 3D-Sensor erfassten Messdaten zur Ausrichtung genutzt. In den sich partiell überlagernden Punktwolken werden dafür Korrespondenzen über ein mathematisches Optimierungsverfahren angenähert (JÄHNE 2005, S. 418; BENNAMOUN & MAMIC 2002, S. 107). Als Korrespondenzen können einzelne Datenpunkte, Ausschnitte der Punktwolke oder definierte Bildmerkmale dienen. Zu finden sind diese ausschließlich im gemeinsamen Bereich von zwei Punktwolken, welcher im Folgenden als *Überlagerungsbereich* bezeichnet wird.

Eine etablierte Möglichkeit zur datengetriebenen Registrierung ist die Aufbringung von codierten Passmarken (BERTAGNOLLI 2006). Diese werden bei der Arbeitsvorbereitung am Prüfobjekt oder dessen Spannvorrichtung befestigt und dienen als definierte Referenzpunkte. Innerhalb der zwei zu verknüpfenden Punktwolken sind diese Passmarken eindeutig identifizierbar. Sie bilden daher bekannte Korrespondenzen und können von einem numerischen Algorithmus angenähert werden. Dabei wird der minimale Abstand zwischen den Korrespondenzen angestrebt und die resultierende Transformation als Ergebnis übergeben. Diese Vorgehensweise ist vor allem in Bezug auf den hohen

## 5 Registrierung durch Verknüpfung von Punktwolken

---

Arbeitsaufwand und die Fehleranfälligkeit bei der manuellen Applikation problematisch. Auf physisch aufgebrachte Passmarken wird daher verzichtet, die Ausrichtung stützt sich auf die Konturen des Prüfobjekts. Um dabei auch merkmalarme Oberflächen verknüpfen zu können, ist die Einbringung zusätzlicher Informationen notwendig. Dazu wird die Oberfläche des Prüfobjekts durch Projektion mit einem Muster angereichert (siehe auch ULRICH et al. 2015). In homogenen Bereichen stehen somit in jedem Fall Konturen zur Verfügung, um eine eindeutige Verknüpfung zu ermöglichen.

Die Herausforderung besteht dabei in der Identifikation von geeigneten Korrespondenzen, da diese zunächst unbekannt sind. In der Folge muss auf Möglichkeiten zur Registrierung zurückgegriffen werden, die keine bekannten Korrespondenzen voraussetzen. Zugehörige Bereiche in den zu verknüpfenden Punktwolken müssen vom Algorithmus gefunden und zugeordnet werden. Bei Betrachtung der veröffentlichten Methoden zur Registrierung von Punktwolken ohne bekannte Korrespondenzen findet sich mit dem „Iterative Closest Point“-Algorithmus ein Quasi-Standard (SZELISKI 2011, S. 588). Dieser dient als mathematische Grundlage der Verknüpfung, muss allerdings auf die Gegebenheiten des Anwendungsfalls angepasst werden. Die Vorteile liegen in der bereits durch andere Arbeiten gezeigten Möglichkeit zur effizienten Implementierung, auch im Kontext der Produktionstechnik (BREITBACH 2015), einer garantierten Konvergenz sowie zahlreichen Ansätzen zur Anpassung an unterschiedlichste Rahmenbedingungen (Abschnitt 4.1.2).

Die Ausrichtung durch einen derartigen Algorithmus ist im Ergebnis eine Transformation, welche die räumliche Beziehung der betrachteten Punktwolken beschreibt. Dazu werden diese schrittweise angenähert und über ein Fehlermaß verglichen. Dieser Fehler ist ein Maß für die Übereinstimmung der verschobenen Punktwolken. Bei minimalem Fehler sind die Punktwolken im Rahmen der Messunsicherheit so angeordnet, wie sie in der Realität zueinander liegen. Abweichende Toleranzen des Prüfobjekts (z. B. die lokal verschobene Position einer Bohrung) beeinflussen diese Ausrichtung nicht negativ. Für die Registrierung wird einzig die Eigenschaft der konstanten Form und Ausdehnung zwischen den Aufnahmen genutzt. Dieses Verfahren zur Verknüpfung wird auf sämtliche sich überlagernden Punktwolken angewendet. Sind die zugehörigen Transformationen berechnet, können die Aufnahmen und damit auch die ent-

---

haltenden Prüfmerkmalsinformationen zueinander in Bezug gesetzt und die gesuchten Messwerte errechnet werden.

Bei Verwendung des ICP-Algorithmus kann die Robustheit der Registrierung durch die Vorgabe einer initialen Transformation verbessert werden (WECKENMANN 2012, S. 232; ZINSSER et al. 2003). Eine derartige Näherungslösung verringert die Wahrscheinlichkeit für fehlerhafte Zuordnungen und erhöht gleichzeitig die Geschwindigkeit der Berechnung. Daher wird die durch Vorwärts-Transformation bestimmte Lage des 3D-Sensors herangezogen, um den Bezug der Punktwolken untereinander abzuschätzen (SURMANN et al. 2003; SHI & XI 2008). Der anschließend zu kompensierende Restfehler entspricht damit der Differenz zwischen realer und durch die Robotersteuerung berechneter Pose.

Im vorliegenden Anwendungsfall steht die Prüfung geometrischer Merkmale im Vordergrund, die auf der Oberfläche des Prüfobjekts verteilt sind (Abbildung 5.2.a). Die vollständige Erfassung und Rekonstruktion dieser Oberfläche wie in Abbildung 5.2.c steht hingegen nicht im Fokus. Es existieren vielmehr einzelne geometrische Merkmale, welche zur Sicherstellung der Qualität geprüft werden. Dabei ist davon auszugehen, dass die zugehörigen Punktwolken initial nicht durch partielle Überlagerung verbunden sind. Für die datengetriebene Registrierung ist jedoch eine derartige Verbindung sämtlicher Punktwolken unerlässlich. Es müssen daher voraussichtlich zusätzliche Aufnahmen geplant werden, sodass sich wie in Abbildung 5.2.b eine geschlossene Kette ergibt. Der damit einhergehende Zusatzaufwand und die potentiell steigende Prüfzeit durch einen längeren Fahrweg des Roboters werden in Kapitel 6 diskutiert.

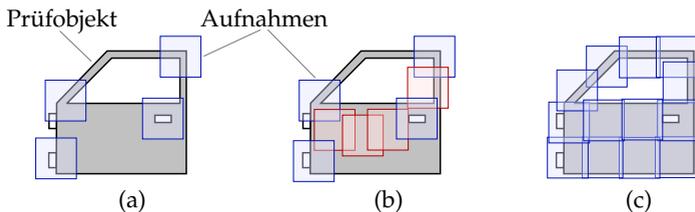


Abbildung 5.2: Das Konzept erfordert neben den Aufnahmen der Prüfmerkmale (a) eine geschlossene Kette von überlagernden Punktwolken (b). Eine vollständige Abdeckung des Prüfobjekts ist dabei nicht nötig (c).

Liegt eine geschlossene Kette aus verbundenen Aufnahmen vor, kann die Registrierung jeder Punktwolke entlang eines zugehörigen *Registrierungspfades* durch paarweise Ausrichtung *inkrementell* erfolgen. Werden mehrere Registrierungspfade pro Punktwolke verwendet, ergeben sich potentiell unterschiedliche Transformationen. Dieses Verhalten ist auf den Restfehler bei der Ausrichtung und die Fehlerfortpflanzung zurückzuführen. Vor allem bei der autonomen Navigation mobiler Roboter ist dieses „Loop Closure“-Problem von Bedeutung (P. NEWMAN & KIN HO 2005). Ansätze für eine *simultane* Registrierung teilen die Abweichung gleichmäßig unter den zu registrierenden Punktwolken auf, sodass sich im Mittel ein Ergebnis mit minimalem Restfehler ergibt (PULLI 1999).

Da eine vollständige Erfassung des Prüfobjekts nicht angestrebt wird, kann zur Verkürzung der Planungs- und Prüfzeit die Anzahl der zusätzlichen Aufnahmen minimal gewählt werden. Für die Methode des 3D-Image-Stitching ergibt sich dadurch ein inkrementelles Vorgehen bei der Registrierung, wenn jeweils ein Registrierungspfad pro Punktwolke ausgewählt wird. Die geschlossene Kette wird somit zu einem *Registrierungsbaum* (in Analogie zur Graphentheorie ein zusammenhängender Graph ohne Kreise, Abschnitt 4.2.3). Das „Loop Closure“-Problem wird somit konzeptionell vermieden. Jede Punktwolke  $X_i$  im Registrierungsbaum wird durch paarweise Ausrichtung mit benachbarten Aufnahmen  $\{X_{i-1}, X_{i+1}\}$  verknüpft. Durch sequentielle Multiplikation der resultierenden Transformationsmatrizen  $T_{i-1 \rightarrow i}$  und  $T_{i \rightarrow i+1}$  werden die Punktwolken zuletzt mit Gleichung 5.1 global registriert.

$$T_{1 \rightarrow n} = \prod_{i=2}^n T_{i-1 \rightarrow i} \quad (5.1)$$

Zusammengefasst wird die Oberfläche des Prüfobjekts mit einer Projektion angereichert, um mit Hilfe eines ICP-basierten Algorithmus Punktwolken paarweise zu verknüpfen. Dazu ist es unabdingbar, neben den zur Messung unmittelbar notwendigen Aufnahmen zusätzliche Punktwolken zu erzeugen, sodass durch partielle Überlagerungen ein Registrierungsbaum entsteht. Die entstehenden Transformationen zwischen benachbarten Aufnahmen werden abschließend sequentiell verrechnet, um damit sämtliche Punktwolken und dar-

---

aus resultierend die enthaltenden Merkmale in ein globales Koordinatensystem zu überführen.

### 5.3 Anreicherung homogener Oberflächen durch Projektion

Aus der Anwendung des angestrebten Prinzips der Registrierung durch Verketten ergibt sich eine Herausforderung für Prüfobjekte mit homogenen Oberflächen. Die zusätzlich eingebrachten Aufnahmen zwischen den Prüfmerkmalen besitzen potentiell keine oder zumindest nur sehr spärliche Oberflächenmerkmale, welche zur Verknüpfung genutzt werden könnten (Abbildung 5.3.a). Eine Einschränkung des Ansatzes auf Oberflächen mit ausreichenden Konturen ist jedoch nicht zielführend, da im betrachteten Anwendungsfall der roboterbasierten Qualitätssicherung im Bereich des Karosseriebaus homogene Oberflächen häufig vorkommen. Zur Befähigung des roboterbasierten Messsystems für den adressierten Anwendungsfall sind demnach zusätzliche Informationen notwendig, auf Basis derer die Transformation berechnet werden kann. Dafür wird ein externes Projektionssystem eingesetzt, das die Oberfläche des Prüfobjekts mit einem optisch aufgetragenen Muster anreichert. Durch diese Projektionen wird die Methode des 3D-Image-Stitching auch an merkmalsarmen Oberflächen anwendbar, wenn der Algorithmus zur Registrierung die zusätzlichen Informationen zur Berechnung der Transformation nutzt (Abbildung 5.3.b).

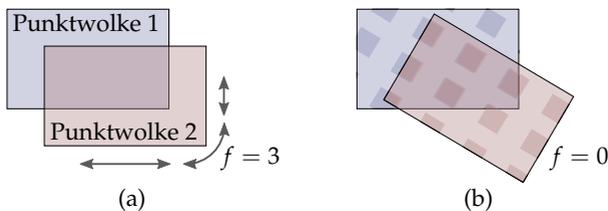


Abbildung 5.3: Bei Aufnahmen von homogenen Oberflächen verbleibt ein Freiheitsgrad  $f > 0$  bei der Ausrichtung (a). Ein projiziertes Muster kann diese Mehrdeutigkeiten unterdrücken (b).

Für die Projektion eines Musters ist zusätzliche Systemtechnik erforderlich. Dabei soll durch die Inbetriebnahme und Nutzung durch den Anwender kein

erheblicher Mehraufwand entstehen. Die zusätzliche Systemtechnik darf dabei das Investitionsvolumen des roboterbasierten Messsystems nur insofern erhöhen, dass der Kostenvorteil gegenüber Tracking-Systemen bestehen bleibt (Abschnitt 3.1). Weiterhin muss für die angereicherten Aufnahmen sichergestellt sein, dass in den betrachteten Bildbereichen das projizierte Muster stets verfügbar und für den 3D-Sensor erfassbar ist. Da die Projektion potentiell die einzige Möglichkeit zur korrekten Registrierung darstellt, muss die Lösung eindeutig bestimmbar sein. Mehrdeutigkeiten sollten demnach bereits konzeptionell unterbunden werden. Es ergeben sich daher folgende Anforderungen an das Projektionssystem:

- Einfache Inbetriebnahme der Systemtechnik durch den Anwender
- Robuste Erfassung der Projektion in sämtlichen Aufnahmen
- Systembedingte Vermeidung mehrdeutiger Lösungen
- Kostenerhöhung durch Systemtechnik vernachlässigbar

### 5.3.1 Systemtechnik für die Muster-Projektion

Die Projektion muss zwischen den zu verknüpfenden Aufnahmen konstant bleiben, um als Bezugsinformation genutzt werden zu können. Die zugehörige Systemtechnik ist folglich von der Roboterkinematik des Messsystems zu entkoppeln. Die weitere Ausgestaltung hängt von der Größe und Art des zu prüfenden Objekts ab. Die Freiheitsgrade hierfür umfassen die Art der Lichtquelle, die Anzahl der Projektionseinheiten und die statische oder dynamische Ausrichtung der Projektion. Für die Lichtquelle bieten sich LED oder Laser-Optiken an. Während LED-basierte Systeme kostengünstiger und unkritisch im Hinblick auf die Arbeitssicherheit sind, bieten auf Lasertechnik basierende Lösungen potentiell schärfere und damit im Bild leichter erkennbare Konturen.

Wie in Abbildung 5.4.a dargestellt, kann die Anreicherung des Prüfobjekts durch statische Projektoren realisiert werden, sodass sich deren Position und damit das Muster nach der Inbetriebnahme nicht ändert. Dabei muss sichergestellt sein, dass sämtliche zur Messung benötigten Bereiche des Prüfobjekts abgedeckt werden. Als Alternative sind dynamische Projektoren denkbar, die von

---

einer separaten Kinematik auf das Prüfobjekt ausgerichtet werden. Dies kann mit Hilfe eines weiteren Industrieroboters geschehen (Abbildung 5.4.b). Ein derartiges System ist flexibler einsetzbar, stellt jedoch höhere Anforderungen an die Bahnplanung, da beide Roboter koordiniert werden müssen. Die Anzahl der notwendigen Projektionseinheiten hängt bei beiden Lösungsvarianten von den zur Messung einzunehmenden Orientierungen des 3D-Sensors ab und sind anwendungsspezifisch festzulegen.

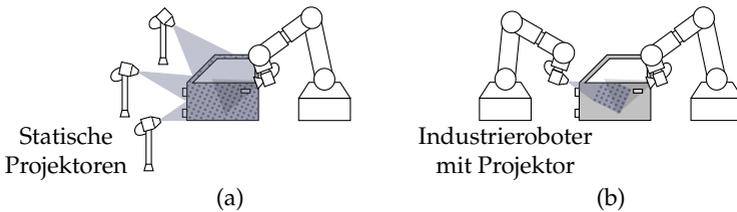


Abbildung 5.4: Die Anreicherung mit einem Muster kann durch statische Projektoren (a) oder eine robotergeführte Projektionseinheit (b) realisiert werden.

Um Komplexität und Kosten gering zu halten, wird der Einsatz statischer Projektoren auf LED-Basis gewählt. Die Ausrichtung durch den Anwender fordert dabei lediglich eine grobe Abdeckung der Oberfläche des Prüfobjekts und kann somit schnell und mit geringem Aufwand durchgeführt werden. Der Algorithmus zur Registrierung sowie die automatische Bahnplanung müssen die gewählte Projektionsmethode entsprechend berücksichtigen.

### 5.3.2 Resultierende Anforderungen

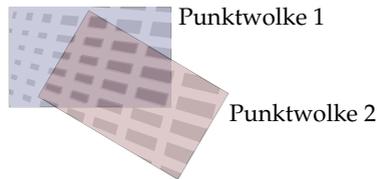
Um die zusätzlich eingebrachten Informationen nutzen zu können, muss der eingesetzte 3D-Sensor die Helligkeit oder Farbe der Oberfläche erfassen und in die resultierende Punktwolke integrieren können. Für den Anwendungsfall der Qualitätssicherung im Automobil- und Flugzeugbau kommen meist 3D-Sensoren nach dem Prinzip der aktiven Triangulation zum Einsatz (Abschnitt 2.1.1). Mit Hilfe der hierfür eingesetzten Kameras kann die Projektion erfasst werden. Zu Anschauungszwecken wird die Anzeige von Farbinformationen in der Praxis bereits realisiert, auch wenn dies zur Erfüllung der Messauf-

## 5 Registrierung durch Verknüpfung von Punktwolken

---

gabe momentan nicht unmittelbar notwendig ist (CARL ZEISS OPTOTECHNIK GMBH 2014; GOM MBH 2010).

Die Projektoren werden vom Anwender grob ausgerichtet. Ein großer Abstrahlwinkel hilft, die Oberfläche vollständig zu bestrahlen. Dadurch ergeben sich lokal differierende Einstrahlwinkel, wodurch das resultierende Muster verzerrt wird. Die Projektion ist daher prinzipiell undefiniert in Form und Gestalt. Die Algorithmen der Bildverarbeitung haben demnach zwar zusätzliche Informationen zur Verfügung, die projizierten Muster besitzen jedoch keine dem Algorithmus im Vorhinein bekannte Form und Ausdehnung. Dies muss bei der Ausgestaltung des Algorithmus zur Registrierung in Abschnitt 5.4 beachtet werden. Bringt dieser Algorithmus die Projektionen zweier Aufnahmen zur Überdeckung, ergibt sich durch den nicht repetitiven Aufbau des Musters eine eindeutige Lösung (Abbildung 5.5). Mehrdeutigkeiten werden somit technisch unterbunden.



*Abbildung 5.5: Die Projektoren erzeugen systembedingt ein lokal verzerrtes Muster. Der nicht gleichförmige Aufbau hilft, eine fehlerhafte Zuordnung einzelner Bildbestandteile zu verhindern.*

Weiterhin ist bei der Auslegung des Algorithmus zur Registrierung zu beachten, dass sich bei unterschiedlichen Posen des 3D-Sensors der Reflexionswinkel und damit die vom Sensor erfasste Helligkeit ändert. Die absoluten Helligkeits- und Farbwerte variieren daher zwischen zwei Aufnahmen. Die Position der Projektion hingegen bleibt konstant und soll daher als alleinige Bezugsinformation zur Registrierung genutzt werden.

Zur Bestimmung der Position sind die Helligkeitswerte der aufgenommenen Bildpunkte ausreichend, der vorgestellte Lösungsansatz beschränkt sich daher auf deren Verarbeitung. Farbinformationen hingegen werden nicht berücksichtigt. Eine analog zu konzipierende technische Lösung auf Basis von Farbwerten

---

ist ebenso möglich. Dabei sind jedoch der Mehraufwand für die Speicherung und Verarbeitung mehrerer Farbkanäle sowie höhere Investitionskosten für 3D-Sensor und Projektor zu beachten, welche Farbinformationen erfassen respektive projizieren können. Vorteile gegenüber einer Berechnung basierend auf Helligkeitswerten sind nicht zu erwarten.

Zuletzt ist bei der Generierung von Roboterbahnen zu beachten, dass die Komponenten des Messsystems die Projektion im Sichtbereich des 3D-Sensors nicht blockieren. Dieses Szenario tritt ein, falls sie während der Aufnahme zwischen Projektor und aufzunehmender Oberfläche positioniert werden. Die automatische Bahnplanung (Kapitel 6) muss dieses Szenario prüfen und gegebenenfalls korrigieren, um eine stets erfassbare Projektion sicherzustellen.

## 5.4 Algorithmus zur paarweisen Verknüpfung

Kern der Methode 3D-Image-Stitching ist ein Algorithmus zur Ausrichtung von zwei sich partiell überlagernden Punktwolken. Das für die globale Registrierung entscheidende Ergebnis ist dabei die Transformation zwischen den Aufnahmen. Aufgrund zunächst unbekannter Korrespondenzen wird hierfür der etablierte ICP-Algorithmus verwendet (WECKENMANN 2012, S. 232). Dieser sucht iterativ nach korrespondierenden Bereichen und minimiert anschließend deren Abstand, bis keine Verbesserung mehr erzielt wird (Abschnitt 4.1.2). Bei homogenen Oberflächen entstehen dabei Mehrdeutigkeiten. Eine eindeutige Lösung wird durch ein projiziertes Muster erzielt, das sich innerhalb der Punktwolke in den Helligkeitswerten der Bildpunkte abbildet. Um diese Informationen in den Algorithmus einzubeziehen, stehen verschiedene Lösungsansätze zur Verfügung.

Für jede neu zu erzeugende Aufnahme ändert die handhabende Kinematik Position und Orientierung des 3D-Sensors. Da sich dadurch der relative Winkel zur beobachteten Oberfläche verändert, variiert der Helligkeitswert korrespondierender Bereiche. Die durch JOHNSON & BING KANG (1999) und JOUNG et al. (2009) vorgeschlagene Metrik in Form einer gewichteten Norm aus Farb- und Helligkeitsunterschied sowie euklidischem Abstand kann daher nicht genutzt werden, da korrespondierende Punkte im Überlagerungsbereich nicht die

## 5 Registrierung durch Verknüpfung von Punktwolken

gleiche absolute Helligkeit aufweisen. Die Autoren weisen außerdem auf die Störanfälligkeit des Helligkeitswerts hin und gewichten diesen daher gering. Zusätzlich wurde der Ansatz für großflächige Areale mit stark texturierter Umgebung entwickelt und ist daher für den hier angestrebten Anwendungsfall der Prüfung gleichförmiger Oberflächen im Submillimeterbereich ungeeignet.

Die Lösungsansätze von WEIK (1997) sowie DRUON et al. (2006) beschreiben eine Einschränkung der betrachteten Punkte auf Basis von Helligkeit oder Farbe. Die örtliche Ausprägung des projizierten Musters sagt jedoch nichts über die Qualität der in diesem Bereich gelegenen geometrischen Informationen aus, daher werden diese Ansätze nicht verfolgt. Für die Ausrichtung werden sämtliche zur Verfügung stehenden Informationen genutzt, um eine möglichst hohe Genauigkeit zu erzielen. Dazu zählen sowohl geometrische Konturen als auch das projizierte Muster.

In beiden Aufnahmen bleibt die Position der Hell-Dunkel-Übergänge des aufgebrauchten Musters konstant (Abbildung 5.6). Allein die Position dieser Kanten soll daher als Zusatzinformation für die Registrierung genutzt werden. Dafür werden die Konturen der projizierten Muster durch eine Kantenerkennung extrahiert und in räumliche Strukturen umgewandelt. Durch diese Umwandlung kann die etablierte, euklidische Metrik genutzt werden. Eine Erweiterung der Abstandsmetrik für zusätzliche Informationen wie Helligkeit oder Farbe sowie deren Gewichtung zueinander entfällt. Zu beachten ist jedoch der Zusatzaufwand in Form einer möglichst robust auszulegenden Kantenerkennung.

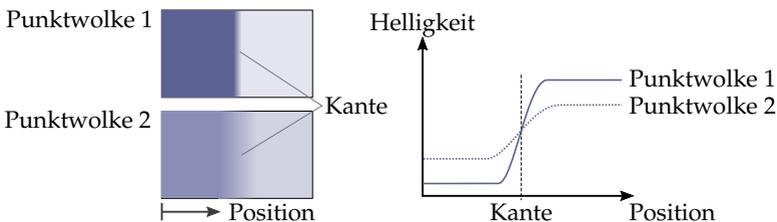


Abbildung 5.6: Unabhängig vom absoluten Helligkeitswert können Kantenpositionen über lokale Änderungen des Helligkeitsverlaufs detektiert werden.

In Abbildung 5.7 ist der Ablauf einer paarweisen Verknüpfung der Punktwolken  $\{X, Y\}$  dargestellt. Im ersten Schritt wird die technische Ausprägung des

---

roboterbasierten Messsystems genutzt und mit Hilfe der Selbstlokalisierung des Roboters eine initiale Ausrichtung  $T_{robot}$  ermittelt. Vor der Anwendung des ICP-Algorithmus erfolgt im zweiten Schritt die Anreicherung der Punktwolken, indem das erfasste Muster in räumliche Strukturelemente umgewandelt wird.

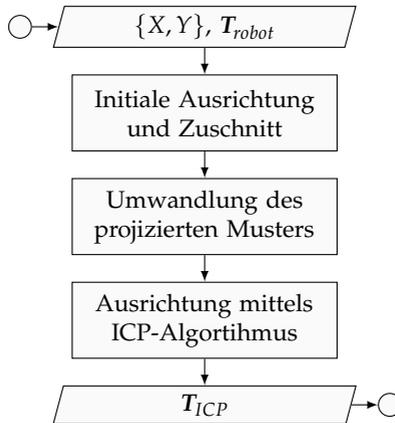


Abbildung 5.7: Ablauf zur paarweisen Verknüpfung von Punktwolken.

#### 5.4.1 Startwert und Zuschnitt

Das handhabende System Industrieroboter kann die Pose des 3D-Sensors durch Vorwärtstransformation im Millimeterbereich bestimmen (Abschnitt 2.2.1). Diese Daten werden mittels Gleichung 5.2 für eine initiale Ausrichtung der Punktwolken genutzt. Ein adäquater Startwert erhöht die Wahrscheinlichkeit, während der Registrierung per ICP-Algorithmus das globale Optimum zu erreichen (WECKENMANN 2012, S. 233).

$$Y_0 = T_{robot} \cdot Y \quad (5.2)$$

Die initiale Ausrichtung wird außerdem genutzt, um die Punktwolken auf den Überlagerungsbereich zu begrenzen, da nur dieser für die Verknüpfung benötigt wird (Abbildung 5.8). Durch die Verringerung der Anzahl betrachteter

## 5 Registrierung durch Verknüpfung von Punktwolken

Bildpunkte verkürzt sich die Berechnungszeit der nachfolgenden Operationen. Die auszurichtende Punktwolke wird weiterhin um den Betrag des maximalen Roboterfehlers  $\epsilon_{robot,max}$  verkleinert, sodass diese unabhängig von der realen Abweichung vollständig in die Zielpunktwolke eingepasst werden kann. Die Korrespondenzsuche des ICP-Algorithmus kann somit stets plausible Korrespondenzen finden, die von TURK & LEVOY (1994) beschriebenen fehlerhaften Mehrfachzuordnungen im Randbereich werden konzeptionell verhindert. In der Folge erhöht sich die Robustheit durch ein stabileres Konvergenzverhalten (RUSINKIEWICZ & LEVOY 2001).

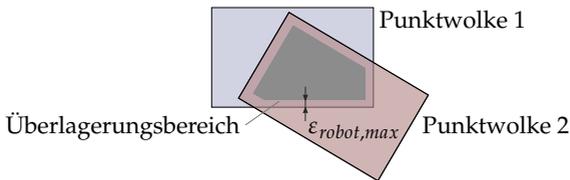


Abbildung 5.8: Der Überlagerungsbereich wird über die initiale Ausrichtung mit  $T_{robot}$  und anschließende Verkleinerung um den Betrag  $\epsilon_{robot,max}$  bestimmt.

### 5.4.2 Umwandlung der Projektion in räumliche Strukturen

Die folgenden Schritte werden sowohl für die Quell- als auch die Zielpunktwolke durchgeführt. Ziel ist die Erzeugung von räumlichen Strukturen auf Basis des projizierten Musters. Dazu werden die Positionen der Kanten detektiert, die durch den Übergang von hellen in dunkle Bildbereiche charakterisiert sind. Anschließend werden am Ort der gefundenen Kanten zusätzliche Punkte senkrecht zur beobachteten Oberfläche generiert. Um die Robustheit der Kantendetektion zu erhöhen, wird in einem vorgelagerten Schritt der Einfluss von Bildrauschen durch Glättung vermindert.

Als Werkzeug der digitalen Bildverarbeitung dienen *lineare Filter*. Für die Anwendung dieser Operatoren wird die durch den 3D-Sensor erzeugte Tiefeninformation während der Berechnung ignoriert und die 3D-Messdaten somit in eine 2D-Bildebene projiziert. Lineare Filter ermitteln den Helligkeitswert eines Bildpunkts  $i^*$  aus der unmittelbaren Nachbarschaft in des Bilds  $I$  mit  $\{m, n\}$  Zeilen und Spalten. Die *Filtermaske*  $F$  definiert, welche Bildpunkte als Nachbarn

---

gelten und wie diese zu gewichten sind. Eine Anwendung linearer Filter entspricht mathematisch einer *diskreten Faltung* nach Gleichung 5.3. (SZELISKI 2011, S. 111; DEMANT et al. 2011, S. 45; BREDIES & LORENZ 2011, S. 65)

$$i_{m,n}^* = (I * F)_{m,n} = \sum_j \sum_k i_{m-j,n-k} \cdot f_{j,k} \quad (5.3)$$

### Unterdrückung von Bildrauschen durch Glättung

Als Bildrauschen werden stochastische Störungen bezeichnet, die zu lokal variierenden Helligkeitswerten führen (SÜSSE & RODNER 2014, S. 190). Nachfolgenden Algorithmen wird durch diesen Effekt die Erkennung spezifischer Strukturen erschwert. Um Bildrauschen zu verringern, wird die Punktwolke daher mit Hilfe eines linearen Filters geglättet (Abbildung 5.9.a im Vergleich zu Abbildung 5.9.b). Glättungsfilter entsprechen einem Tiefpassfilter und basieren auf der Bildung von gewichteten Mittelwerten (DEMANT et al. 2011, S. 48).

Ein optimaler Glättungsfilter nutzt die Form einer Gauß'schen Glockenkurve als Grundlage der Filtermaske (DEMANT et al. 2011, S. 52). Durch Diskretisierung folgt beispielhaft für eine 3x3-Filtermaske  $G_{3 \times 3}$  der in Gleichung 5.4 dargestellte Aufbau. Wird dieser Operator angewandt, beeinflussen benachbarte Punkte umso stärker das Ergebnis, je näher sie sich am Zentrum der Filtermaske befinden. Die Faltung der Punktwolke  $I$  wird durch sequenzielles Anwenden der Filtermaske  $G_{3 \times 3}$  auf jeden Bildpunkt realisiert, die resultierende Punktwolke ist geglättet (JÄHNE 2005, S. 118). Die Größe der Filtermaske beeinflusst die Schärfe des Ausgangsbildes sowie die notwendige Berechnungszeit und ist anwendungsspezifisch festzulegen (DEMANT et al. 2011, S. 53).

$$G_{3 \times 3} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (5.4)$$

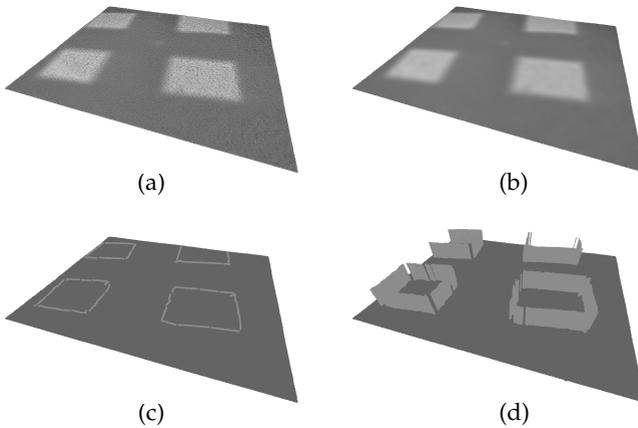


Abbildung 5.9: Die zugeschnittene Punktwolke (a) wird geglättet (b), bevor Kanten detektiert (c) und an diesen Stellen zusätzliche räumliche Strukturen generiert werden (d).

### Kantenerkennung

Kanten entsprechen Diskontinuitäten im Helligkeitsverlauf eines Bilds. Im vorliegenden Fall markieren sie die Ränder des projizierten Musters in der Punktwolke. Zur Detektion dieser Kanten wird das Verfahren von Canny (SZELISKI 2011, S. 239) genutzt (Abbildung 5.9.c). SONKA et al. (1993, S. 88) beschreiben die Methode als optimal, um Kanten zuverlässig zu detektieren. Der Algorithmus zählt zu den isotropen Detektoren, Kanten werden somit unabhängig von ihrer Richtung erkannt (JÄHNE 2005, S. 352).

Durch Faltung der Punktwolke  $I$  mit zwei zueinander orthogonalen Sobel-Filtern (Gleichung 5.5) werden zunächst partielle Ableitungen berechnet, wobei abrupte Hell-Dunkel-Übergänge Extremwerte bewirken. Die Gradientenstärke  $\rho$  (Gleichung 5.6) dient daher als Hinweis, ob ein Bildpunkt zu einer Kante gehört (DEMANT et al. 2011, S. 59). Durch die partiellen Ableitungen kann zusätzlich mit Gleichung 5.7 die Richtung der entsprechenden Kante  $\Theta$  berechnet werden. Anschließend werden gefundene Kanten verdünnt. Dazu wird die Dicke der Kante und anschließend deren Zentrum bestimmt. Kanten werden auf dieses Zentrum und somit auf eine Breite von einem Bildpunkt begrenzt.

Zuletzt werden ungültige Kantenstücke detektiert und unterdrückt. Die hierzu nötigen Schwellwerte werden nach ROSIN & ELLIS (1995) aus dem Median der Helligkeitswerte berechnet. (SONKA et al. 1993, S. 88; BREDIES & LORENZ 2011, S. 75)

$$\mathbf{S}_x = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \mathbf{S}_y = \frac{1}{8} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (5.5)$$

$$\rho_{m,n} = \sqrt{(\mathbf{I} * \mathbf{S}_x)_{m,n}^2 + (\mathbf{I} * \mathbf{S}_y)_{m,n}^2} \quad (5.6)$$

$$\Theta_{m,n} = \arctan \frac{(\mathbf{I} * \mathbf{S}_y)_{m,n}}{(\mathbf{I} * \mathbf{S}_x)_{m,n}} \quad (5.7)$$

### Addition in Richtung der Normalen

Die Punktwolke wird nun mit zusätzlichen Strukturelementen angereichert, indem am Ort der gefundenen Kanten synthetische Bildpunkte erzeugt werden. Da 3D-Image-Stitching vor allem für die Registrierung von homogenen Oberflächen eingesetzt wird, ist von einer geringen räumlichen Streuung orthogonal zur Oberfläche auszugehen. Die zusätzlichen Merkmale werden daher in dieser Richtung konstruiert (Abbildung 5.9.d).

Zunächst wird die Richtung des zugehörigen Bildpunkts senkrecht zur Oberfläche bestimmt. Dazu wird der Normalenvektor aus den räumlichen Informationen innerhalb einer Maske bestimmt. Diese umfasst den in Abbildung 5.10 dargestellten Bereich bestehend aus den Bildpunkten  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_8\}$  mit  $\mathbf{p} = [x \ y \ z]^T$ , in dessen Zentrum sich der betrachtete Kantenpunkt  $\mathbf{p}_0$  befindet. Nach Gleichung 5.8 wird aus jeweils drei benachbarten Punkten ein Normalenvektor berechnet, wobei der zentrale Bildpunkt  $\mathbf{p}_0$  stets beteiligt ist. Anschließend wird durch Vektoraddition  $\mathbf{n}_{add}$  und mittels Gleichung 5.9 der Normaleneinheitsvektor  $\mathbf{n}_{norm}$  gebildet, der als Ergebnis die zur betrachteten Oberfläche orthogonale Richtung repräsentiert (KARPFINGER 2014, S. 562). Je nach Güte der Punktwolken kann zur Berechnung die Maske vergrößert werden, um die Streuung der resultierenden Normalenvektoren zu verringern.

$$\mathbf{n}_{add} = \sum_{i=1}^7 (\mathbf{p}_0 - \mathbf{p}_i) \times (\mathbf{p}_0 - \mathbf{p}_{i+1}) \quad (5.8)$$

$$\mathbf{n}_{norm} = \frac{\mathbf{n}_{add}}{\|\mathbf{n}_{add}\|} \quad (5.9)$$

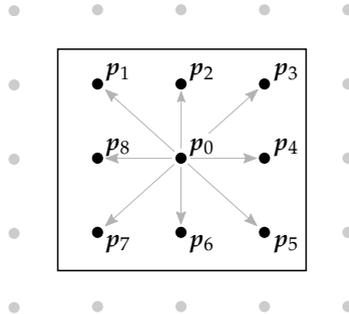


Abbildung 5.10: Der Normalenvektor  $\mathbf{n}_{norm}$  in  $\mathbf{p}_0$  wird innerhalb einer  $(3 \times 3)$ -Maske mit Hilfe der umgebenden Bildpunkte  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_8\}$  berechnet.

Für jeden Kantenpunkt wird nun in Richtung des zugehörigen Normalenvektors eine definierte Anzahl weiterer Bildpunkte hinzugefügt. Der Abstand zwischen den Bildpunkten entspricht dabei dem durchschnittlichen Abstand innerhalb der Punktwolke. Je mehr Punkte konstruiert werden, desto größer wird potentiell der Einfluss des projizierten Musters auf die Registrierung. Das Ergebnis ist eine Punktwolke mit synthetischen Strukturen in Form von Flächen, welche senkrecht auf der Oberfläche des Prüfobjekts stehen. Die optimale Anzahl hinzuzufügender Bildpunkte wird in einer Parameterstudie ermittelt (Abschnitt 7.2).

### 5.4.3 Ausrichtung mittels ICP-Algorithmus

Die angereicherten Punktwolken werden durch einen modifizierten „Iterative Closest Point“-Algorithmus (Abschnitt 4.1.2) zueinander ausgerichtet. Wie in Abbildung 5.11 dargestellt, vereinfachen dabei die im Überlagerungsbereich generierten räumlichen Strukturen die Korrespondenzsuche. Resultat ist die Transformation  $T_{ICP}$  als Beschreibung der räumlichen Verschiebung und

---

Rotation zwischen beiden Punktwolken. Ausgehend von der ursprünglichen Ausprägung des Algorithmus nach BESL & MCKAY (1992) werden notwendige Anpassungen und Modifikationen im Folgenden erläutert.

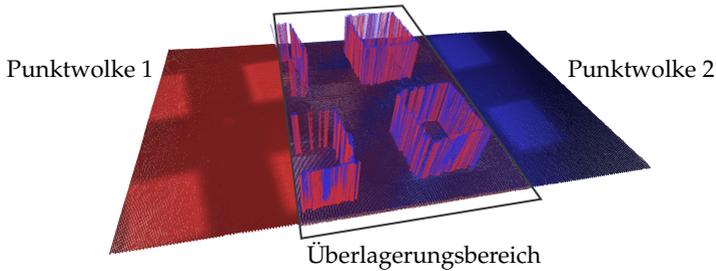


Abbildung 5.11: Die generierten räumlichen Strukturen im Überlagerungsbereich vereinfachen die Korrespondenzsuche des ICP-Algorithmus.

### Vorauswahl der Punkte

Höchste Priorität bei der Registrierung hat die möglichst präzise Ausrichtung der Punktwolken. Wird die Anzahl der Punkte zur Verknüpfung eingeschränkt, kann dies die erreichbare Genauigkeit negativ beeinflussen. Allerdings kann die Beschränkung auf eine Teilmenge von Punkten die Berechnungszeit wesentlich verkürzen. Eine potentielle Vorauswahl ist daher zweckmäßig für den praktischen Einsatz, darf jedoch die Genauigkeit des Algorithmus nicht beeinträchtigen.

Zur Verfügung stehen deterministische und stochastische Auswahlmethoden. Durch die eingesetzte geometrische Codierung der Projektion werden Verfahren auf Basis von Farbe und Intensität (WEIK 1997; DRUON et al. 2006) ausgeschlossen. Zwischen den verbleibenden Methoden konnte in einer ausführlichen Studie kein wesentlicher Unterschied aufzeigt werden, solange ausreichend identifizierbare Merkmale vorhanden sind (RUSINKIEWICZ & LEVOY 2001). Dies ist durch die Vorverarbeitung gegeben. Es wird daher auf das Verfahren einer zufälligen Punktauswahl zurückgegriffen, da dieses Konzept vergleichsweise aufwandsarm implementierbar ist. Um maximale Genauigkeit zu gewährleisten, wird die Anzahl der Punkte pro Iteration des Algorithmus erhöht (ZHANG

1994). Die Berechnungszeit wird somit in den ersten Iterationen durch eine geringe Anzahl an Punkten beschleunigt, gleichzeitig wird zuletzt die vollständige Auflösung der Punktwolke zur Ausrichtung genutzt.

### Bestimmung von Korrespondenzen

Beim Einsatz der durch CHEN & MEDIONI (1991) vorgeschlagenen Punkt-zu-Ebene-Methode konvergiert der ICP-Algorithmus potentiell schneller als bei Nutzung der Punkt-zu-Punkt-Methode nach BESL & MCKAY (1992) (RUSINKIEWICZ & LEVOY 2001). Für den betrachteten Anwendungsfall ist die Punkt-zu-Ebene-Methode jedoch nicht zweckmäßig, da die zur Berechnung benötigte Normalenrichtung der zusätzlich generierten Punkte keine reale Oberfläche repräsentiert. Ferner ist diese Methode laut PULLI (1999) ungeeignet für homogene Oberflächen, da durch Aliasing („Treppeneffekt“) und Rauschen bedingte, geringfügige Änderungen der Normalenrichtung bereits vergleichsweise große Fehler bei der Berechnung verursachen können. Es wird daher auf die nach BESL & MCKAY (1992) eingeführte Punkt-zu-Punkt-Methode zurückgegriffen.

Die Umwandlung der Intensität in räumliche Strukturelemente ermöglicht die Nutzung der euklidischen Norm im  $\mathbb{R}^3$  als Abstandsmetrik  $D_{ICP}$ . Eine Nutzung der Mahalanobis-Distanz würde Vorteile bei stark verrauschten Messdaten bieten (HANSEN et al. 2007). Da die Punktwolken im Rahmen der Vorverarbeitung jedoch geglättet werden, ist kein Vorteil gegenüber der euklidischen Metrik zu erwarten.

Es besteht die Möglichkeit, zusätzlich invariante Features der Punktwolken für die Korrespondenzsuche zu verwenden (FELDMAR & AYACHE 1994). SHARP et al. (2002) empfiehlt dieses Vorgehen für strukturierte Umgebungen ohne geschätzte initiale Transformation. Im vorliegenden Anwendungsfall werden jedoch vorwiegend Prüfobjekte mit sehr geringer Strukturierung betrachtet und es steht eine bereits hinreichend genaue Schätzung der Transformation durch die Selbstlokalisierung des Roboters zur Verfügung. Invariante Features werden daher nicht zur Berechnung verwendet.

---

## Gewichtung und Zurückweisung

Die geometrische Codierung der Intensität wird durch die Generierung von zusätzlichen Punkten realisiert. Je mehr synthetische Punkte pro gefundener Kante eingefügt werden (Abschnitt 5.4.2), desto größer wird deren Einfluss auf die Registrierung. Informationen aus Intensität und Geometrie werden auf diese Weise zueinander gewichtet.

Durch das Zuschneiden der Punktwolken auf Basis der geschätzten Verschiebung in Abschnitt 5.4.1 werden kritische Randpunkte aussortiert. Nach dem Zuschchnitt ist sichergestellt, dass die Quellpunktwolke stets vollständig in der Zielpunktwolke enthalten ist. Dadurch werden Mehrfachzuweisungen im Randbereich verhindert, jeder Punkt der Quellpunktwolke besitzt definitiv einen korrespondierenden Punkt in der Zielpunktwolke. Diese durch TURK & LEVOY (1994) & PULLI (1999) aufgezeigten Fehlerquellen bei der Korrespondenzsuche werden somit ausgeschlossen.

Zuletzt werden potentiell fehlerhaft erfasste Punkte (z. B. durch Reflexionen) identifiziert und zurückgewiesen. Verfahren wie der RANSAC-Algorithmus (RIDENE & GOULETTE 2009) und Kovarianzmatrizen (ESTÉPAR et al. 2004) können durch die Beschaffenheit der synthetisch konstruierten Merkmale nicht angewandt werden, da eben diese zusätzlich generierten Punkte gefiltert würden. Zielführend ist stattdessen ein Grenzwert für den maximalen Abstand korrespondierender Punkte (MASUDA et al. 1996). In Anlehnung an NEUGEBAUER (1997) wird daher für jedes Punkte-Paar die Anforderung nach Gleichung 5.10 geprüft. Falls der Abstand  $D_{ICP}$  größer ist als die erwartete Abweichung des Roboters  $\varepsilon_{robot}$ , wird das betreffende Punkte-Paar  $\{y_i, x_i\}$  in der nachfolgenden Minimierung ignoriert.

$$D_{ICP}(y_i, x_i) > \varepsilon_{robot} \quad (5.10)$$

## Fehlermaß und Minimierung

Als Fehlermaß  $\varepsilon$  wird der Mittelwert der quadratischen Abstände sämtlicher  $n$  Punkte-Paare festgelegt (Gleichung 5.11). Der Abstand berechnet sich da-

bei mit Hilfe der Metrik  $D_{ICP}$ . Zur Minimierung des Fehlers wird die in Abschnitt 4.1.1 vorgestellte Methode mit Quaternionen nach HORN (1987) verwendet. Da die Genauigkeit der Ausrichtung höchste Priorität hat und die Berechnungszeit zweitrangig behandelt wird, gilt das Konvergenzkriterium  $\tau$  nach Gleichung 5.12 als Abbruchkriterium. Der Algorithmus endet daher, sobald die Verkleinerung des Fehlers  $\varepsilon$  in zwei aufeinanderfolgenden Iterationen kleiner als  $\tau$  ist. Die Methode folgt damit der durch BESL & MCKAY (1992) vorgeschlagenen Vorgehensweise. Das Auffinden des globalen Optimums kann dabei nicht garantiert werden.

$$\varepsilon = \frac{1}{n} \sum_{i=1}^n \|x_i - T \cdot y_i\|^2 \quad (5.11)$$

$$\tau > (\varepsilon_{j-1} - \varepsilon_j) \quad (5.12)$$

### 5.5 Einfluss von Abbildungsfehlern

Die Methode 3D-Image-Stitching nutzt das Prinzip der datengetriebenen Registrierung und ist daher abhängig von der Güte der erzeugten Punktwolken (KEFERSTEIN & MARXER 2015, S. 238). Der Einfluss von Abbildungsfehlern auf die inkrementelle Verknüpfung wird im Folgenden analysiert. Ziel ist dabei, den entstehenden *Messfehler* (Differenz des Messwerts zum realen Prüfmerkmal) abzuschätzen sowie Stellgrößen zur Beeinflussung und gegebenenfalls Verringerung aufzuzeigen.

#### 5.5.1 Fehler der paarweisen Verknüpfung

Die paarweise Verknüpfung nutzt die aufgenommenen Punktwolken zur Berechnung der räumlichen Verschiebung und Rotation. Der eingesetzte 3D-Sensor ist bei der Erzeugung dieser Punktwolken in der Genauigkeit begrenzt, da jede optische Aufnahme fehlerbehaftet ist (PUENTE LEÓN & KIENCKE 2012, S. 17). Die Abweichungen weisen dabei systematische oder stochastische Ursachen auf (PUENTE LEÓN & KIENCKE 2012, S. 18). Der 3D-Sensor wurde für Prüfaufgaben ausgelegt, in denen eine Erfassung mit höchstmöglicher Präzision

---

essentiell ist. Systematische Fehler sind daher bereits soweit machbar durch den Hersteller kompensiert, die Größenordnung verbleibender systematischer Fehler ist im Verhältnis zum stochastischen Fehler vernachlässigbar klein. Ein derartig kalibrierter 3D-Sensor zur Erzeugung der Punktwolke mit minimal möglichem systematischen Messfehler wird als Rahmenbedingung vorausgesetzt.

Die stochastischen Fehlerursachen sind vielfältig und finden sich unter anderem in der Übertragung analoger in digitale Bildsignale (Quantisierungsrauschen), in Verzeichnungen durch die eingesetzte Optik und in numerischen Abweichungen bei der softwaretechnischen Verarbeitung (LÖFFLER-MANG 2012, S. 65; TIPLER et al. 2015, S. 1066). Der resultierende zufällige Messfehler wird zunächst durch Mittelung mehrerer Aufnahmen abgeschwächt. Der Fehler wird hierdurch bei  $m$  Aufnahmen um den Faktor  $\sqrt{m^{-1}}$  gedämpft (SÜSSE & RODNER 2014, S. 192). Stochastische Fehler können somit vermindert, jedoch nicht vollständig unterdrückt werden. Zwei beliebige Aufnahmen unterscheiden sich somit im Bereich des resultierenden Restfehlers. Eine Verknüpfung mit dem in Abschnitt 5.4 vorgestellten Algorithmus wird somit von der realen Transformation abweichen.

Zur Verknüpfung selbst stehen nicht sämtliche Daten der Punktwolken zur Verfügung. Der Algorithmus ist vielmehr auf den Bereich begrenzt, in dem sich die Aufnahmen überschneiden. Abhängig von der Transformation  $T_{i \rightarrow j}$  zwischen den Punktwolken  $\{X_i, X_j\}$  definiert sich dieser Überlagerungsbereich  $\kappa_{i \rightarrow j} = \kappa(T_{i \rightarrow j}, X_i, X_j)$ . Die resultierende Streuung wird als *Einzelverknüpfungsfehler*  $\varepsilon_{pair; i \rightarrow j}$  nach Gleichung 5.13 zusammengefasst.

$$\varepsilon_{pair, i \rightarrow j} = \varepsilon_{pair}(\kappa_{i \rightarrow j}) \quad (5.13)$$

## 5.5.2 Fehlerfortpflanzung bei inkrementeller Registrierung

Der Einzelverknüpfungsfehler  $\varepsilon_{pair}$  gilt für je zwei verknüpfte Punktwolken im Registrierungsbaum. Für eine inkrementelle Registrierung aller Punktwolken werden die errechneten Verschiebungen aufeinander aufbauend verrechnet, wodurch sich jeder Fehler auf nachfolgende Verknüpfungen auswirkt. Bedingt

## 5 Registrierung durch Verknüpfung von Punktwolken

---

durch diese Fehlerfortpflanzung ist der Messfehler  $\varepsilon_{path, 1 \rightarrow n}$  für eine Prüfaufgabe abhängig von den Einzelverknüpfungsfehlern entlang des verwendeten Registrierungs-pfads  $\Psi_{1 \rightarrow n}$  (Gleichung 5.14).

$$\varepsilon_{path, 1 \rightarrow n} = \varepsilon_{path}(\varepsilon_{pair, 1 \rightarrow 2}, \varepsilon_{pair, 2 \rightarrow 3}, \dots, \varepsilon_{pair, n-1 \rightarrow n}) \quad (5.14)$$

Um diesen stochastischen Messfehler  $\varepsilon_{path}$  entlang eines Registrierungs-pfades zu berechnen, wird das **Gauß'sche Fehlerfortpflanzungsgesetz** genutzt. Dies gilt für kleine, zufällige und voneinander unabhängige Messfehler und besagt, dass sich die Varianz zusammengesetzter Messwerte  $Var(\varepsilon) = \varepsilon^2$  aus der Summe der gewichteten Einzelvarianzen errechnet (PUENTE LEÓN & KIENCKE 2012, S. 176). Im vorliegenden Anwendungsfall der zusammengesetzten geometrischen Messung ergibt sich dabei der Gewichtungsfaktor 1. Der zufällige Messfehler  $\varepsilon_{path; 1 \rightarrow n}$  steigt demnach mit der Anzahl zu verknüpfender Aufnahmen  $n$  im Registrierungs-pfad und berechnet sich näherungsweise mittels Gleichung 5.15.

$$\varepsilon_{path, 1 \rightarrow n} \approx \sqrt{\varepsilon_{pair, 1 \rightarrow 2}^2 + \varepsilon_{pair, 2 \rightarrow 3}^2 + \dots + \varepsilon_{pair, n-1 \rightarrow n}^2} \quad (5.15)$$

Der zu erwartende Messfehler ist somit von der konkreten Prüfaufgabe abhängig, da hierdurch die Ausprägung der Überlagerungsbereiche  $\kappa$  sowie die Anzahl an Aufnahmen  $n$  festgelegt wird. Es wird daher das Prinzip der prüfaufgabenbezogenen Überwachung angewendet und die Eignung des Messsystems bezogen auf die jeweils aktuelle Aufgabe bewertet (PFEIFER & SCHMITT 2010, S. 441).

### 5.5.3 Verringerung des Messfehlers

Um im Sinne der Prüfmittelüberwachung die notwendige Genauigkeit des Messsystems zu gewährleisten, kann der Messfehler bereits während der Prüfplanung abgeschätzt werden. Da die beeinflussenden Größen Überlagerungsbereich  $\kappa$  und Anzahl der Aufnahmen  $n$  in der Prüfplanung beeinflusst werden können, wird zusätzlich eine Verringerung des Messfehlers ermöglicht.

Es wird angenommen, dass sich mehr Datenpunkte durch einen vergrößerten Überlagerungsbereich  $\kappa$  positiv auf die Genauigkeit der Verknüpfung auswirken. Mit größerem  $\kappa$  sinkt demnach potentiell der Fehler  $\varepsilon_{pair}$ . Für den Einfluss der Fehlerfortpflanzung hingegen gilt, je weniger Aufnahmen  $n$  verknüpft werden, desto geringer wird der resultierende Fehler. Da die Prüfaufgabe  $\mathcal{M}$  durch den Anwender vorgegeben wird, ist auch der durch die Aufnahmen zu überbrückende Abstand gegeben. Das Sichtfeld des 3D-Sensors weist ebenfalls konstante Abmessungen auf, wodurch ein definierter Überlagerungsbereich  $\kappa$  gleichzeitig die Anzahl notwendiger Aufnahmen  $n$  festlegt. Dieser Zusammenhang ist in Abbildung 5.12 dargestellt. Eine Verringerung von  $n$  kann daher nur durch eine Verkleinerung des Überlagerungsbereichs  $\kappa$  erreicht werden. Dies wiederum führt zu einem erhöhten Messfehler der paarweisen Verknüpfung.

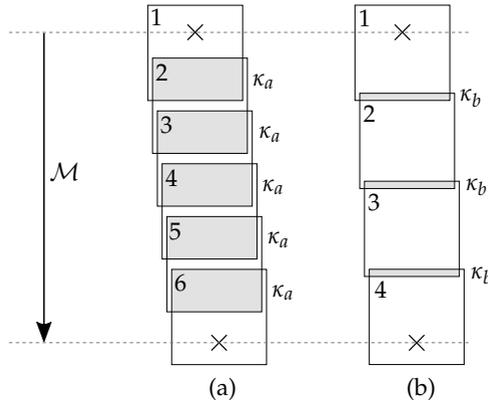


Abbildung 5.12: Die Größe des Überlagerungsbereichs  $\kappa$  beeinflusst die Anzahl der notwendigen Aufnahmen  $n$  bei gegebener Prüfaufgabe  $\mathcal{M}$ .

Zusammenfassend wird demnach zur Minimierung des stochastischen Fehlers bei der Verknüpfung zweier Punktwolken die Vergrößerung des Überlagerungsbereiches wie in Abbildung 5.12.a angestrebt. Gleichzeitig wäre zur Verringerung des Einflusses der Fehlerfortpflanzung eine Verkleinerung desselben wie in Abbildung 5.12.b notwendig. Beide Anforderungen stehen sich diametral gegenüber. Ohne weitere Untersuchung ist unklar, ob Fall (a) oder Fall (b) den kleineren Messfehler  $\varepsilon_{path}$  aufweist. Ein optimaler Wert für den vereinfachten Anwendungsfall eines gleichbleibenden Überlagerungsbereichs

## 5 Registrierung durch Verknüpfung von Punktwolken

---

$\kappa$  über sämtliche Verknüpfungsoperationen minimiert den Messfehler  $\varepsilon_{path}$  und lässt sich nach Gleichung 5.16 bestimmen.

$$\varepsilon_{path} : \kappa_{optimal} \mapsto \min \varepsilon_{path} \Rightarrow \frac{\partial \varepsilon_{path}}{\partial \kappa} \stackrel{!}{=} 0 \quad (5.16)$$

Die dargelegten Zusammenhänge über den Einfluss von stochastischen Messfehlern während der datengetriebenen Registrierung werden in Kapitel 6 genutzt, um durch Beeinflussung des Überlagerungsbereichs  $\kappa$  die Genauigkeit der Messung zu steigern.

### 5.6 Zusammenfassung der Methode zur Registrierung

Die Registrierung durch 3D-Image-Stitching basiert auf dem Prinzip der datengetriebenen und inkrementellen Registrierung. Anhand der Messdaten werden benachbarte Punktwolken zueinander ausgerichtet. Um homogene Oberflächen einbeziehen zu können, wird das Prüfobjekt mit Hilfe eines projizierten Musters angereichert. Dieses Muster wird in räumlichen Strukturen kodiert, um anschließend durch einen modifizierten ICP-Algorithmus verknüpft zu werden. Das Ergebnis ist die Transformation zwischen beiden Aufnahmen.

Durch Aneinanderreihung sich stets überlagernder Punktwolken werden auf diese Weise sämtliche Aufnahmen ausgerichtet und in ein globales Koordinatensystem überführt. Dazu wird ein Registrierungsbaum gebildet, der die notwendigen Punktwolken enthält. Der bei einer Registrierung durch 3D-Image-Stitching entstehende Messfehler ist abhängig von den jeweiligen Überlagerungsbereichen und der Anzahl zu verknüpfender Punktwolken.

## 6 Automatisierte Bahnplanung

Für den praxisgerechten Einsatz roboterbasierter Messsysteme ist neben der Sicherstellung einer ausreichend hohen Messgenauigkeit die Anwendbarkeit zu betrachten. Dazu zählt die Zeit zur Inbetriebnahme sowie die Komplexität für Programmierung und Bedienung. Ausgehend von einem funktionsfähigen Aufbau besteht der Prüfprozess nach PFEIFER & SCHMITT (2010, S. 13) aus folgenden Phasen:

1. Prüfplanung  
Spezifikation der Messaufgabe und Auswahl zu prüfender geometrischer Merkmale am Prüfobjekt.
2. Prüfdatenerfassung  
Umsetzung der Messaufgabe in ein Ablaufprogramm und Aufnahme der Punktwolken durch das roboterbasierte Messsystem.
3. Prüfdatenauswertung  
Auswertung, Weitergabe und Archivierung der Messergebnisse.

Prüfplanung und Prüfdatenauswertung stehen in der vorliegenden Arbeit außerhalb der Betrachtungsgrenzen, da das roboterbasierte Messsystem keinen Einfluss auf diese Aspekte hat. Für die Prüfdatenerfassung hingegen wird die Erzeugung eines Ablaufprogramms durch den Anwender fokussiert, um eine vorgegebene Prüfaufgabe auszuführen. Bezogen auf roboterbasierte Messsysteme sind dafür zwei Teilarbeitsschritte notwendig:

1. Festlegung konkreter Positionen und Orientierungen des Sensors im Raum und damit verbundener Posen des Roboters, an denen die Prüfmerkmale aufgenommen werden.
2. Verbinden dieser Posen zu einer geschlossenen, kollisionsfreien Roboterbahn.

## 6 Automatisierte Bahnplanung

---

Bei dieser Planung sind die Kriterien Messgenauigkeit und Prüfzeit zu optimieren. Beide werden durch die Ausprägung der generierten Posen und die nachfolgende Trajektorie des Roboters beeinflusst. Bei deren Erzeugung müssen außerdem Anforderungen des Sensorsystems (z. B. Sichtfeldgröße), der handhabenden Kinematik (z. B. Achsbeschränkungen) sowie globale Rahmenbedingungen (z. B. Kollisionsfreiheit) eingehalten werden, wodurch die manuelle Planung als sehr komplex und zeitaufwändig eingeschätzt werden kann. Weiterhin ergeben sich durch den vorgestellten Ansatz des 3D-Image-Stitching zusätzliche Anforderungen (z. B. freie Sichtlinie für die Projektion), welche die Planung weiter erschweren. Für eine praxisgerechte Umsetzung der Methode ist daher eine automatisierte Erzeugung des Prüfablaufprogramms zielführend. Ist die Programmerstellung automatisiert, wird die Bedienung des Gesamtsystems vereinfacht und beschleunigt. Im Folgenden wird daher ein Verfahren zur automatisierten Erzeugung des Prüfablaufprogramms vorgestellt.

### 6.1 Anforderungen und Rahmenbedingungen

Bahnplanungsverfahren erzeugen aus einer abstrakt gestellten Aufgabe ein Ablaufprogramm für das ausführende System (CHOSET et al. 2005, S. 1). Essentiell für den Erfolg ist dabei die Erfüllung im Vorhinein gestellter Anforderungen. Als Beispiel kann die Kollisionsfreiheit genannt werden. Nur wenn diese sichergestellt werden kann, ist die Bahnplanung von praktischem Wert.

Die konkrete Durchführung der Messaufgabe bedeutet, dass geometrische Merkmale erfasst und zu einem Referenzmodell in Bezug gesetzt werden. Da das Prüfobjekt den Sichtbereich des 3D-Sensors überschreitet, müssen die Aufnahmen in ein globales Koordinatensystem überführt werden. Dazu wird eine Anzahl an Aufnahmen in der Art erstellt, dass sich die in Abschnitt 5.2 geforderte geschlossene Kette von Punktwolken zwischen den Prüfmerkmalen ergibt. Durch Registrierung sämtlicher Aufnahmen wird die Messung der Prüfmerkmale ermöglicht. Als *Prüfposen* werden dabei Posen des Roboters bezeichnet, in denen Prüfmerkmale erfasst werden und die somit zwingend zur Erfüllung der Prüfaufgabe notwendig sind. Die verbleibenden Posen zur Verknüpfung hingegen sind *Verknüpfungsposen*. Wie in Abschnitt 5.5 dargestellt, wird der zu

---

erwartende Messfehler durch die Anzahl der Aufnahmen und die Überlagerungsbereiche beeinflusst. Die Prüfzeit wiederum ergibt sich aus der Aufnahmezeit des 3D-Sensors und der Fahrzeit des eingesetzten Industrieroboters. Die gesuchte Roboterbahn entspricht einer Aneinanderreihung von Posen, an denen der 3D-Sensor Punktwolken aufnehmen kann. Diese werden anschließend mit Hilfe des Algorithmus zur paarweisen Verknüpfung (Abschnitt 5.4) zueinander ausgerichtet.

Die Methode zur Erstellung des Ablaufprogramms muss in der Lage sein, Position und Orientierung des 3D-Sensors so zu bestimmen, dass eine adäquate Punktwolke generiert werden kann. Um erfolgreich die Oberfläche des Prüfbodens aufzunehmen, ist dazu ein für den 3D-Sensor charakteristischer Neigungswinkel und Abstand zur Oberfläche einzuhalten (LEE & PARK 2000; SCOTT et al. 2003). Verfahrensbedingt muss außerdem die aufgebrachte Projektion (Abschnitt 5.3) erfasst und demnach eine Abschattung durch den handhabenden Roboter oder den 3D-Sensor vermieden werden. Anschließend müssen aus diesen Informationen gültige Posen des Vertikal-Knickarmroboters erzeugt werden. Zuletzt sind diese Posen zu einer geschlossenen Trajektorie zu verbinden. Dabei muss die Kollisionsfreiheit zu jedem Zeitpunkt sichergestellt sein. Zusätzlich sind ein definierter Arbeitsraum sowie die Achswinkelbeschränkungen des verwendeten Roboters einzuhalten (ULRICH et al. 2016).

## **6.2 Konzept zur Programmerstellung**

Die Positionierung des 3D-Sensors und die daraus abgeleitete Pose des Roboters ist von zahlreichen Parametern abhängig. Eine analytische Beschreibung des Konfigurationsraums ist dabei nicht praktikabel (LAVALLE 2011). Für die Programmerstellung kommt daher ein stichprobenbasiertes Verfahren zum Einsatz, in dem der Konfigurationsraum nicht vollständig abgebildet werden muss. Da stets mehrere Prüfmerkmale zu messen sind, wird mit der PRM ein Multi-Query-Planer als Basis verwendet. Vorteile dieses Verfahrens sind die ausführliche Untersuchung in der Literatur sowie eine vielfache Implementierung, in denen die Leistungsfähigkeit der Methode auch im industriellen Umfeld bereits gezeigt wurde (SAHA et al. 2003; GECKS & HENRICH 2009; ULRICH et al. 2016).

Die PRM muss für die praktische Anwendung an die Methode 3D-Image-Stitching angepasst werden. Im Gegensatz zur ursprünglichen PRM können die Prüfposen nicht über einen beliebigen Pfad im Raum verbunden werden. Stattdessen kann die Messung nur durchgeführt werden, wenn eine geschlossene Kette an überlagernden Aufnahmen bereit steht, um die Punktwolken zu verknüpfen. Es ist daher zunächst ein Registrierungsbaum (Abschnitt 5.2) inklusive Verknüpfungsposen zu finden. Hierbei ist zunächst eine isolierte Betrachtung der statischen Posen ausreichend, ohne die Bewegung des Industrieroboters zwischen diesen Posen zu berücksichtigen. Die erreichbare Genauigkeit der Messung hängt von der Generierung und Auswahl geeigneter Verknüpfungsposen ab. Um diese erzeugen zu können, wird das Prüfobjekt zunächst analysiert. Dazu dient das digitale Referenzmodell, gegen welches die Messdaten letztlich geprüft werden. Durch Modellierung des roboterbasierten Messsystems in Verbindung mit dem aktuellen Prüfobjekt wird eine simulative Generierung von Prüf- und Verknüpfungsposen ermöglicht.

Die Bewegung der handhabenden Kinematik wird separat berechnet. Steht ein Registrierungsbaum aus überlagernden Aufnahmen zur Verfügung, sind sämtliche vom Roboter zu erreichenden Posen bekannt. Ziel ist im Anschluss eine Roboterbahn zu finden, welche diese Posen in möglichst kurzer Zeit abfahren kann. Werden einzelne problematische Posen detektiert, kann zusätzlich auf Single-Query-Planer wie RRT zurückgegriffen werden. Diese Art der Planungsalgorithmen kann für die Verbindung zweier Posen effizienter eingesetzt werden als die PRM (CHOSSET et al. 2005, S. 227).

Die Planung enthält damit zwei aufeinander aufbauende Abläufe. Zunächst werden die Verknüpfungsoperationen in einem Registrierungsbaum und danach eine Sequenz sämtlicher Posen in einer Roboterbahn zusammengefasst. Während im Ersteren der Messfehler minimiert wird, ist das Ziel der Letzteren die Minimierung der für den Prüfzyklus notwendigen Zeit. In Verbindung mit dem initialen Aufbau des Konfigurationsraums gliedert sich die automatische Erstellung des Prüfprogramms in die in Abbildung 6.1 dargestellten drei Phasen:

### 1. Generierung der Prüf- und Verknüpfungsposen

Jedes Prüfmerkmal muss vom 3D-Sensor aufgenommen werden, um es mit Hilfe einer nachgelagerten Software auszuwerten. Es ist daher jeweils

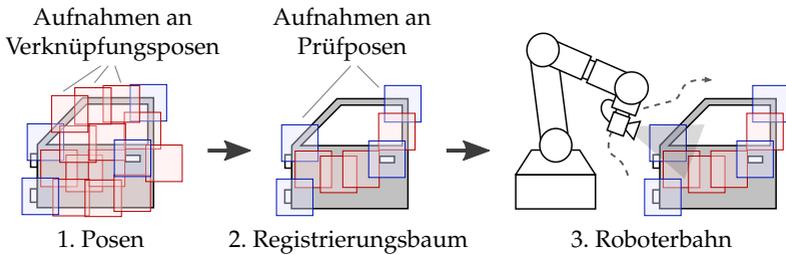


Abbildung 6.1: Für das Prüfprogramm wird ein Set aus Posen erzeugt, auf die notwendigen Aufnahmen reduziert und eine zugehörige Roboterbahn generiert.

eine Prüfpose des Messsystems zu finden, die den Sensor in die Lage versetzt, das Prüfmerkmal korrekt zu erfassen. Zusätzlich erfordert der Lösungsansatz Aufnahmen zur Verknüpfung der Prüfposen, um diese zueinander in Bezug setzen zu können. Es wird daher eine Menge von Verknüpfungsposen gesucht, von denen jeweils eine Überlagerung mit dem Aufnahmebereich benachbarter Posen besteht.

## 2. Aufbau des Registrierungsbaums

Zur Registrierung werden die Prüfposen mit Hilfe der Verknüpfungsposen zueinander in Bezug gesetzt. Posen mit überlapertem Sichtbereich können dazu mit Hilfe des Algorithmus zur paarweisen Verknüpfung registriert werden. Es ist daher eine Menge von Verknüpfungsposen zu finden, welche die Prüfposen adäquat verbindet. Durch eine geeignete Auswahl kann der stochastische Messfehler der inkrementellen Registrierung verringert werden.

## 3. Erzeugung der Roboterbahn

Mit der bekannten Menge von Prüf- und Verknüpfungsposen werden diese im dritten Schritt in eine kollisionsfreie und fahrzeitoptimierte Roboterbahn eingefügt. Der Messfehler kann in diesem Schritt nicht mehr beeinflusst werden, es wird daher die Prüfzeit als Optimierungskriterium genutzt.

### 6.3 Analyse des Prüfobjekts

Um den Anforderungen an die Positionierung des Sensors gerecht zu werden, ist explizites Wissen über das zu prüfende Objekt notwendig. Das CAD-Modell des Prüfobjekts liefert diese Informationen und wird daher im ersten Schritt analysiert. Die aufbereiteten Daten legen die Grundlage zur anschließenden Berechnung der Posen.

Um den Neigungswinkel und Abstand des 3D-Sensors zur Oberfläche des Prüfobjekts zu ermitteln, wird deren Normalenrichtung bestimmt (WU et al. 2015). Hierzu soll das Prüfobjekt im „Standard Tessellation Language“ (STL)-Format vorliegen. Das STL-Format wurde als Austauschformat für Rapid-Prototyping-Anwendungen entwickelt und beschreibt die Oberfläche des Modells in Form von Dreiecken (GRIMM 2004, S. 55). Für jedes dieser Dreiecke berechnet sich der Normalenvektor  $\mathbf{n}$  mit den Ortsvektoren der Eckpunkte  $\mathbf{a}$ ,  $\mathbf{b}$  und  $\mathbf{c}$  wie in Abbildung 6.2 nach Gleichung 6.1 (BÄR 2001, S. 53). Durch die Konvention des Dateiformats STL ist sichergestellt, dass der resultierende Vektor  $\mathbf{n}$  vom Prüfobjekt abgewandt ausgerichtet ist (GRIMM 2004, S. 59).

$$\mathbf{n} = (\mathbf{b} - \mathbf{a}) \times (\mathbf{c} - \mathbf{a}) \quad (6.1)$$

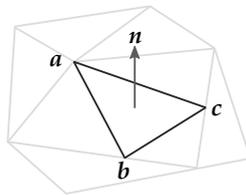


Abbildung 6.2: Der Normalenvektor  $\mathbf{n}$  eines Oberflächen-Dreiecks im Dateiformat STL wird durch die Ortsvektoren der Eckpunkte  $\{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$  bestimmt.

Zur Generierung der Posen und Berechnung des zugehörigen Überlagerungsbereichs  $\kappa$  wird die Oberfläche diskretisiert. Hierzu werden nicht die vorhandenen Dreiecke genutzt, da deren Größe und damit die Genauigkeit der Darstellung variabel ist und sich somit zwischen den Prüfobjekten unterscheiden kann. Um die notwendige Überlagerung sicherzustellen, ist jedoch eine definierte

Dichte an Verknüpfungsposen notwendig, die durch eine parametrisierte Diskretisierung gesteuert wird. Aus der Diskretisierung ergibt sich die Größe der einzelnen Flächenelemente, die zur Berechnung des Überlagerungsbereichs genutzt werden können. Je kleiner diese Elemente, desto genauer kann der Überlagerungsbereich bestimmt werden, jedoch erhöht sich die Berechnungszeit exponentiell. Die Aufteilung erfolgt in Form eines gleichmäßigen Rasters im Abstand  $r$ , jeder Rasterpunkt entspricht demnach etwa einer Fläche von  $r^2$  auf der Oberfläche des Prüfobjekts (Abbildung 6.3). Die Normalenrichtung wird dabei aus dem jeweilig umgebenden Dreieck zugewiesen. Der optimale Wert für die Rastergröße  $r$  wird in einer Parameterstudie ermittelt (Abschnitt 7.2). Zur Erhöhung der Flexibilität werden zusätzliche Rasterpunkte auf den Kanten des Modells erzeugt. Die Normalenrichtung berechnet sich dabei als Mittelwert aus den Normalenvektoren der benachbarten Dreiecke (TEKOUO MOUTCHIO 2012, S. 64).

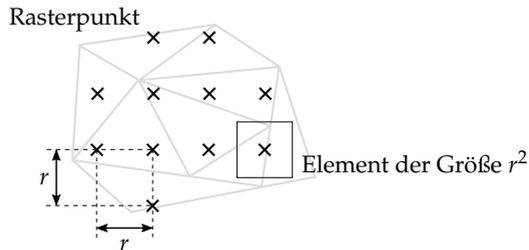


Abbildung 6.3: Diskretisierung der Oberfläche durch ein gleichmäßiges Raster im Abstand  $r$ . Jeder Rasterpunkt repräsentiert ein Element der Größe  $r^2$ .

## 6.4 Generierung von Aufnahmeposen

Im ersten Schritt zur Programmerstellung wird der freie Konfigurationsraum (Abschnitt 4.2.1) partiell abgebildet. Dies entspricht der Generierung einer Menge gültiger Prüf- und Verknüpfungsposen. Diese definieren Position und Orientierung des Messsystems im Raum, sodass durch den 3D-Sensor ein Teil des Prüfobjekts erfasst werden kann. Als *Aufnahmeposen* sei die Gesamtheit aus Prüf- und Verknüpfungsposen bezeichnet. Prüfposen beziehen sich jeweils auf die geometrischen Merkmale, die im Messprozess erfasst werden müssen. Sie

## 6 Automatisierte Bahnplanung

---

müssen daher in jedem Fall bei der Durchführung der Prüfaufgabe vom Roboter mindestens einmal erreicht werden. Ausgangspunkt ist ein Prüfplan, der die zu prüfenden Merkmale charakterisiert. Er enthält sämtliche relevanten Daten, um den aufzunehmenden Bereich zu beschreiben. Aus diesen Daten werden in Verbindung mit den Sensoranforderungen (Abschnitt 6.1) die Prüfposen abgeleitet.

Verknüpfungsposen hingegen erzeugen zusätzliche Aufnahmen für eine geschlossene Kette an überlagernden Punktwolken. Wie viele und welche Posen dazu nötig sind, ist zum Zeitpunkt der Posengenerierung nicht festgelegt. Diese Auswahl wird im nachfolgenden Schritt in Abschnitt 6.5 getroffen. Die Verknüpfungsposen werden durch den Abstand der Rasterpunkte  $r$  aus Abschnitt 6.3 und die Sensoranforderungen aus Abschnitt 6.1 beeinflusst. Sofern Aufnahme-  
posen für einen zusammenhängenden Registrierungsbaum mit ausreichender Überlagerung existieren, werden diese mit ausreichend kleiner Rastergröße  $r$  gefunden. Eine gültige Aufnahmepose erfüllt dabei folgende Anforderungen:

- **Definierter Abstand zur Oberfläche des Prüfobjekts (Abschnitt 6.4.1)**  
Die Oberfläche wird optimal erfasst, wenn der ideale Abstand zum 3D-Sensor eingehalten wird. Sollte dies nicht möglich sein, muss der Abstand innerhalb sensorspezifischer Grenzen liegen, damit gültige Punktwolken erfasst werden können.
- **Neigungswinkel zwischen 3D-Sensor und Oberfläche (Abschnitt 6.4.1)**  
Um Reflexionen zu vermeiden, ist ein sensorspezifischer Neigungswinkel zwischen 3D-Sensor und Oberfläche einzuhalten. Definiert durch den eingesetzten 3D-Sensor existieren für diesen Parameter eine Unter- und Obergrenze, welche zwingend eingehalten werden müssen.
- **Achsbeschränkungen des Roboters (Abschnitt 6.4.2)**  
Der Industrieroboter muss die angestrebte Pose erreichen und darf dabei die Achswinkelbegrenzungen nicht überschreiten.
- **Arbeitsraum der Messzelle (Abschnitt 6.4.2)**  
Das Robotersystem darf die Grenzen des Arbeitsraums beim Einnehmen der Pose nicht überschreiten.

---

- **Kollisionsfreiheit (Abschnitt 6.4.2)**

Das Messsystem muss die ermittelte Pose kollisionsfrei einnehmen können. Es darf in der Pose weder mit dem Prüfobjekt, mit der Arbeitszelle noch mit sich selbst kollidieren.

- **Sichtbereich des Sensors (Abschnitt 6.4.3)**

Der erfasste Bereich des Prüfobjekts muss bekannt sein, um die Überlagerung mit benachbarten Posen berechnen zu können.

- **Sichtbare Projektion (Abschnitt 6.4.3)**

Zur Verknüpfung der Posen wird eine externe Projektion aufgebracht, die weder durch den handhabenden Roboter noch durch den 3D-Sensor abgeschattet werden darf.

### 6.4.1 Anforderungen des Sensors

Die Aufnahmepose wird zunächst mit idealen Bedingungen des 3D-Sensors bezogen auf Abstand und Neigungswinkel berechnet. Abbildung 6.4 zeigt schematisch eine gültige Pose. Der Mittelpunkt  $p_M$  des Sichtbereichs ergibt sich hierfür bei Prüfposen aus dem Mittelpunkt des zu prüfenden Merkmals, bei Verknüpfungsposen entspricht  $p_M$  einem Rasterpunkt, welcher durch die Diskretisierung der Oberfläche entsteht. Die Qualität der Aufnahme ist vom Neigungswinkel  $\alpha$  sowie dem Abstand  $d$  zur Oberfläche des Prüfobjekts abhängig. Als Bezug dient der in Abschnitt 6.3 berechnete Normalenvektor  $n$  am Mittelpunkt des Aufnahmeortes. Der Neigungswinkel  $\alpha$  ergibt sich aus dem Differenzwinkel der Sensor-Sichtachse sowie dem Normalenvektor  $n$ . Der Abstand  $d$  ermittelt sich aus der Strecke zwischen 3D-Sensor und Mittelpunkt  $p_M$ .

Für beide Parameter existieren jeweils minimale, maximale und ideale Werte. Der ideale Wert ist für ein optimales Ergebnis anzustreben. Bezogen auf Abbildung 6.4 ergeben sich allerdings ebenfalls ideale Bedingungen bei Neigung des Sensors um  $-\alpha$ , sowie bei beliebiger Rotation um den Normalenvektor  $n$ . Für einen beliebigen Aufnahmeort existieren daher theoretisch unendlich viele Posen zur optimalen Erfassung. Die verbleibenden Freiheitsgrade der

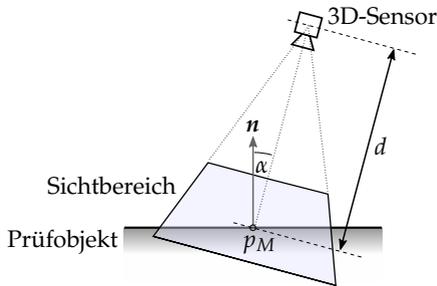


Abbildung 6.4: Gültige Aufnahmen erfordern einen sensorspezifischen Winkel  $\alpha$  und Abstand  $d$  zwischen 3D-Sensor und Prüfobjekt.

Sensorpose haben keinen Einfluss auf die Qualität der späteren Aufnahme und werden zufallsbasiert ausgewählt.

### 6.4.2 Anforderungen des handhabenden Roboters

Nachdem Position und Orientierung des 3D-Sensors festgelegt wurden, muss die physische Erreichbarkeit durch den handhabenden Roboter geprüft werden. Der 3D-Sensor bildet den Endeffektor des Roboters. Dessen Pose  $q$  ist über die Achswinkel der Kinematik eindeutig beschreibbar. Diese werden durch eine Rückwärtstransformation aus der kartesischen Angabe von Position und Orientierung des Endeffektors errechnet (Abschnitt 2.2.1). Das Ergebnis hierbei kann mehrdeutig sein, da mathematisch bis zu acht Lösungen (verschiedene Achswinkel-Konfigurationen) möglich sind (Abbildung 6.5). Eingeschränkt werden diese durch die Grenzen der einzelnen Achswinkel. Die konkreten Werte hierfür ergeben sich durch physikalische Begrenzungen der Drehgelenke oder Anwendereingaben.

Ist eine gültige Achswinkel-Konfiguration gefunden, muss das Gesamtsystem in der entsprechenden Pose auf Kollisionen mit sich selbst oder der Umgebung geprüft werden. Der zugehörige Kollisionsdetektor wird dabei als „Black Box“ betrachtet (Abschnitt 4.2.2). Dieser beantwortet Anfragen mit der Information, ob das Messsystem die vorgegebene Pose kollisionsfrei einnehmen kann. Falls eine Kollision vorliegt, muss eine alternative Pose gefunden und geprüft werden.



*Abbildung 6.5: Für jede Lage des Endeffektors existieren bei Vertikal-Knickarmrobotern acht Lösungen. Rot markierte Gelenkstellungen sind aufgrund mechanischer Begrenzungen nicht erreichbar.*

Zur Prüfung benötigt der Kollisionsdetektor ein Modell der Arbeitsumgebung. Die Modellierung des roboterbasierten Messsystems und der Arbeitsumgebung ist dabei eine vereinfachte Darstellung der Realität und daher stets mit Abweichungen behaftet (BUNGARTZ et al. 2009, S. 3). Zusätzlich weist das Robotersystem Ungenauigkeiten bei der Positionierung auf (ULRICH et al. 2014). Um die Kollisionsdetektion dennoch zuverlässig durchführen zu können, wird ein Toleranzabstand eingeführt. Wird dieser unterschritten, gilt dies innerhalb der Simulation ebenfalls als Kollision, da in der Realität eine Berührung nicht ausgeschlossen werden kann. Der Grenzwert muss dabei auf die Güte des Modells und die Präzision des Roboters angepasst werden. Die Grenzen der Arbeitszelle werden ebenfalls als Körper (virtuelle Wände) modelliert, falls Zellenbegrenzungen (z. B. Zäune) nicht bereits im Modell enthalten sind. Die Prüfung auf Einhaltung von Arbeitsraumgrenzen ist damit der Kollisionsdetektion inhärent.

### **6.4.3 Berechnung des Sichtbereichs und externe Projektion**

Es müssen die Teile der Oberfläche bestimmt werden, die sich zum Zeitpunkt der Aufnahme im Sichtbereich des 3D-Sensors befinden. Dazu wird zunächst mit Hilfe der sensorspezifischen geometrischen Abmessungen des Sichtbereichs

## 6 Automatisierte Bahnplanung

---

die Menge von Rasterpunkten  $P_{sight} = \{p_1, p_2, \dots, p_i\}$  bestimmt, die sich auf der Oberfläche des Prüfobjekts und gleichzeitig im Sichtbereich des 3D-Sensors befinden (Abschnitt 6.3). Bei der Aufnahme sind jedoch nur Rasterpunkte erfassbar, bei denen sich zusätzlich der relative Neigungswinkel und Abstand zum 3D-Sensor innerhalb der charakteristischen Grenzen befinden. Für jeden Punkt in  $P_{sight}$  kann die Erfassbarkeit geprüft werden, indem diese Anforderungen herangezogen werden. Für jeden Aufnahmepunkt  $p_i$  wird daher mittels Gleichung 6.2 geprüft, ob der jeweilige Oberflächenpunkt tatsächlich erfasst wird. Die resultierende Punktmenge  $P_{detect} \subseteq P_{sight}$  beinhaltet durch das Flächenäquivalent  $r^2$  der Rasterpunkte aus Abschnitt 6.3 näherungsweise die Teile der Oberfläche, die während des Prüfvorgangs in der Punktwolke abgebildet werden.

$$p_i := \begin{cases} \text{erfassbar} & , \text{ falls } d_{min} \leq d \leq d_{max} \text{ und } \alpha_{min} \leq \alpha \leq \alpha_{max} \\ \text{nicht erfassbar} & , \text{ sonst} \end{cases} \quad (6.2)$$

Diese Kontrolle ist bei Prüfposen essentiell, um die Erfassung des geometrischen Prüfmerkmals in seiner Gesamtheit sicherzustellen. Um Aufnahmen mit Hilfe des in Abschnitt 5.4 vorgestellten Algorithmus verknüpfen zu können, muss außerdem eine ausreichende Überlagerung vorhanden und eine externe Projektion sichtbar sein. Die für eine Verknüpfung nutzbaren Punkte  $P_{reg}$  ergeben sich daher nach Gleichung 6.3 aus der Schnittmenge sämtlicher durch den 3D-Sensor erfassbaren Punkte  $P_{detect}$  und den Punkten  $P_{proj}$ , an denen das projizierte Muster sichtbar ist. Die Menge  $P_{reg}$  muss eine ausreichende Anzahl an Punkten enthalten, damit die Aufnahme erfolgreich verknüpft werden kann.

$$P_{reg} := P_{detect} \cap P_{proj} \quad \text{mit } P_{detect}, P_{proj} \subseteq P_{sight} \quad (6.3)$$

Die Menge der Punkte mit externer Projektion  $P_{proj}$  wird ermittelt, indem für jeden Aufnahmepunkt im Sichtbereich  $P_{sight}$  der Strahlengang zum Projektionszentrum betrachtet wird. Wird dieser Strahlengang nicht durch das Robotersystem oder andere Objekte unterbrochen, ist die Projektion während

---

der Aufnahme sichtbar und der Rasterpunkt ist Teil von  $P_{proj}$ . Für diesen Nachweis muss ein Modell des vollständigen Messsystems in der aktuellen Pose (Abschnitt 6.4.2) bekannt sein, um den Strahlengang korrekt nachbilden zu können.

#### 6.4.4 Ablauf zur Generierung von Aufnahmeposes

Die Aufnahmeposes werden mit dem in Abbildung 6.6 dargestellten Ablauf erzeugt. Falls dabei nach einer definierten Anzahl von Durchläufen keine gültige Pose gefunden wird, erfolgt der Abbruch des Algorithmus. Der Ablauf initiiert mit optimalen Werten für Abstand  $d$  und Neigungswinkel  $\alpha$ . Ist keine Pose mit optimalen Parameterwerten möglich, werden Ausweichposes gesucht. Deren Parameterwerte liegen zwischen dem jeweiligen Minimum und Maximum von  $d$  respektive  $\alpha$  und werden zufällig ausgewählt. Demnach entstehen stets gültige, bezogen auf die Aufnahmequalität allerdings nicht optimale Aufnahmen.

Das Ziel der Generierung von Aufnahmeposes ist die Erzeugung einer Abtastung  $G_{sample}$  des freien Konfigurationsraums  $Q_{free}$ . Dazu wird der Algorithmus für jedes Prüfmerkmal und jeden Rasterpunkt der Oberfläche durchgeführt. Es entsteht jeweils eine Pose, für welche zusätzlich die Menge nutzbarer Punkte  $P_{reg}$  ermittelt wird. Gefundene Prüfposes bilden die Menge  $Q_P$ , Verknüpfungsposes die Menge  $Q_V$  mit  $Q_P \cup Q_V \subset Q_{free}$ .

#### 6.5 Erstellung des Registrierungsbaums

Mit Bereitstellung von Aufnahmeposes ist die Voraussetzung für die Erstellung des Registrierungsbaums  $G_{tree}$  geschaffen. Das Ziel ist dabei das Verbinden aller Prüfposes, sodass die globale Registrierung ermöglicht wird. Im Gegensatz zu klassischen Bahnplanungsproblemen wird demnach nicht unmittelbar ein räumlicher Weg im freien Konfigurationsraum  $Q_{free}$  gesucht, sondern zunächst ein Pfad aus sich überlagernden Aufnahmen. Das Optimierungskriterium hierbei ist der Messfehler. Dieser wird durch den Überlagerungsbereich  $\kappa_{i \rightarrow j}$  zwischen den benachbarten Aufnahmen  $\{X_i, X_j\}$  bestimmt. Das Ziel der zweiten Phase ist demnach die Auswahl der zur Durchführung der Prüfaufgabe notwendigen

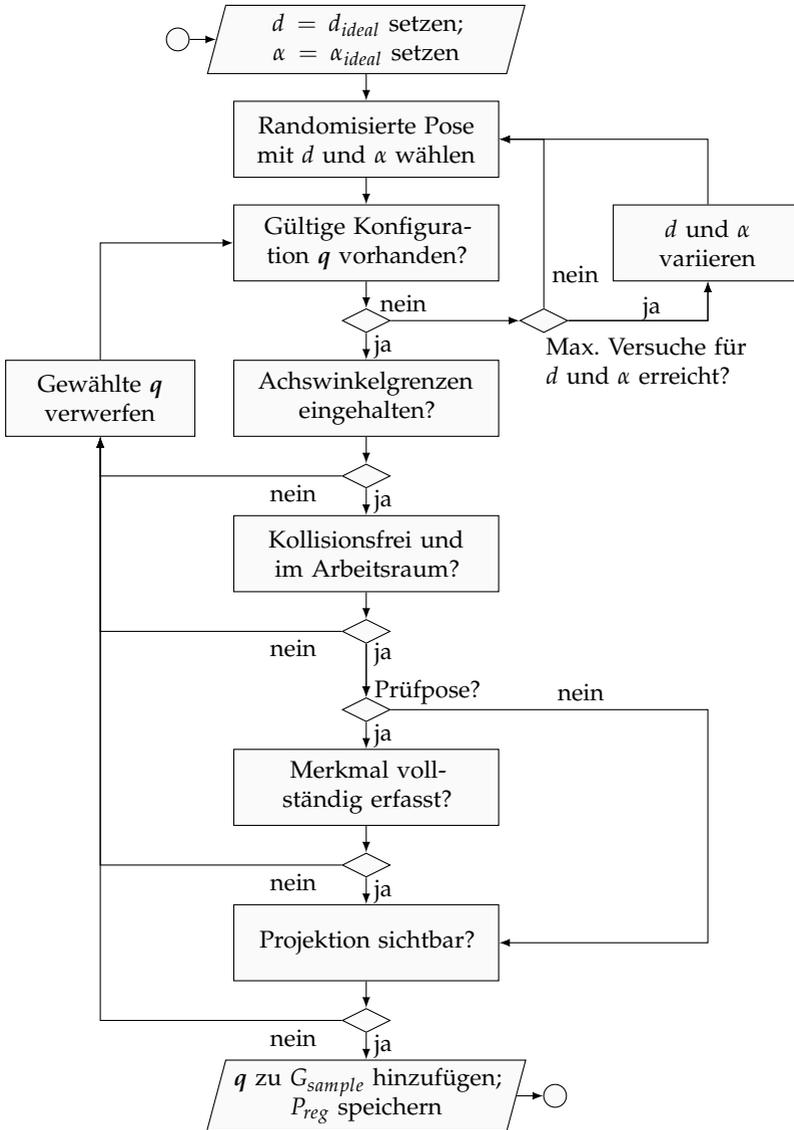


Abbildung 6.6: Ablauf zur Erzeugung einer gültigen Aufnahmepose. Gefundene Posen bilden eine Abtastung  $G_{sample}$  des freien Konfigurationsraums  $\mathcal{Q}_{free}$ .

---

Posen sowie die Zusammenfassung in einem zusammenhängenden Registrierungsbaum  $G_{tree}$ , sodass der Messfehler minimal ausfällt.

### 6.5.1 Modellierung der Bahnplanungsaufgabe

Aufgrund der statischen Ausgangslage (die Posen verändern sich während eines Prüfvorgangs nicht) und des nur partiell bekannten Konfigurationsraums  $\mathcal{Q}_{free}$  wird die „Probabilistic Roadmap Method“ (Abschnitt 4.2.4) zur Lösung der Aufgabe herangezogen. Das Problem wird dazu mit Hilfe der Graphentheorie modelliert. Zunächst wird ein Graph  $G_{reg}$  erzeugt, dem die Prüfposen  $\mathcal{Q}_P$  als Ecken  $\{q_1, q_2, \dots, q_n\}$  hinzugefügt werden. Die Kanten des Graphen repräsentieren in der ursprünglich vorgestellten PRM kollisionsfreie Wege des handhabenden Systems (KAVRAKI et al. 1996). Im vorliegenden Fall können zwei Posen jedoch nur verbunden werden, wenn neben der Kollisionsfreiheit auch eine ausreichende Überlagerung vorliegt. Die PRM wird daher angepasst, Kanten entsprechen einer möglichen Verknüpfung durch den Algorithmus aus Abschnitt 5.4. Die Bewegung des handhabenden Systems wird in diesem Schritt nicht berücksichtigt und in Abschnitt 6.6 behandelt. Da die Prüfposen sich potentiell nicht überlagern, werden in Abschnitt 6.4 Verknüpfungsposen erzeugt. Das Hinzufügen dieser Posen  $\mathcal{Q}_V$  zu  $G_{reg}$  mit anschließender Kantenbildung zu benachbarten Posen verringert die Anzahl nicht verbundener Ecken. Durch die geometrisch gleichförmige Verteilung der Rasterpunkte entspricht dies einem gitterbasierten Ansatz zum Aufbau der Roadmap. Nachbarn werden mit Hilfe der Metrik  $D_{PRM}$  identifiziert, diese entspricht dem euklidischen Abstand der Rasterpunkte. Abhängig vom Anwendungsfall und dem 3D-Sensor muss mit den dargelegten Zusammenhängen aus Abschnitt 5.5 ein minimal notwendiger Wert für den Überlagerungsbereich  $\kappa$  berechnet werden. Die Generierung von Verknüpfungsposen und Erstellung von Kanten wird so lang fortgesetzt, bis sämtliche Prüfposen im Graph  $G_{reg}$  untereinander verbunden sind.

Jeder erstellten Verbindung wird außerdem ein Kantengewicht  $\theta$  zugewiesen. Dafür wird die in Abschnitt 5.5 betrachtete Varianz  $Var(\varepsilon_{pair, i \rightarrow j}) = \varepsilon_{pair, i \rightarrow j}^2$  zwischen den Ecken  $\{q_i, q_j\}$  genutzt (Gleichung 6.4). Der Graph  $G_{reg}$  ist dabei ungerichtet aufgebaut, es gilt  $\varepsilon_{pair, i \rightarrow j}^2 = \varepsilon_{pair, j \rightarrow i}^2$ . Der Messfehler ist abhängig vom Überlagerungsbereich  $\kappa$ , dessen Größe wiederum nach Gleichung 6.5 mit

Hilfe der Rasterpunkte der benachbarten Posen  $\{P_{reg,i}, P_{reg,j}\}$  und der Rastergröße  $r$  abgeschätzt werden kann. Da die in der Realität angefahrne Pose durch die Ungenauigkeit des Industrieroboters geringfügig von der geplanten Pose abweicht, kann der zu erwartende stochastische Messfehler nur näherungsweise bestimmt werden.

$$\theta(q_i, q_j) := \varepsilon_{pair, i \rightarrow j}^2 \quad (6.4)$$

$$\kappa = |P_{reg,i} \cap P_{reg,j}| \cdot r^2 \quad (6.5)$$

### 6.5.2 Spannender Baum mit minimalem Messfehler

Abbildung 6.7.a zeigt beispielhaft den zusammenhängenden Graph  $G_{reg}$  mit den Prüfposen  $\{q_1, q_2, q_3\}$  und den Verknüpfungsposen  $\{q_4, q_5, q_6, q_7\}$ . Für die Generierung des Registrierungsbaums werden zunächst die jeweils kürzesten Wege  $\Psi_{reg, i \rightarrow j}$  zwischen zwei beliebigen Prüfposen  $\{q_i, q_j \mid q_i, q_j \in \mathcal{Q}_P\}$  gesucht. Der eingesetzte Algorithmus von Dijkstra sucht dabei den Pfad aus Posen  $\{q \mid q \in \mathcal{Q}_{free}\}$  mit der geringsten Distanz  $\Theta = \sum \varepsilon_{pair}^2$  von der Start- zur Zielecke und findet stets die optimale Lösung. Der potentiell effizientere A\*-Algorithmus kommt nicht zum Einsatz, weil eine Heuristik zur Abschätzung der Distanz zum Ziel nicht zur Verfügung steht (Abschnitt 4.2.3). Im vorliegenden Anwendungsfall und mit Grundlage des Gauß'schen Fehlerfortpflanzungsgesetzes werden somit die Wege  $\Psi_{reg, i \rightarrow j}$  mit dem geringsten Messfehler  $\min(\varepsilon_{path, i \rightarrow j})$  gefunden. Da die Prüfposen im Graph  $G_{reg}$  verbunden sind, ist das jeweilige Vorhandensein eines gültigen Pfades garantiert.

Neben  $G_{reg}$  wird nun der Graph  $G_{tree}$  erzeugt, der einzig die Prüfposen  $\mathcal{Q}_P$  als Ecken enthält (Abbildung 6.7.b). Jedes Posen-Paar  $\{q_i, q_j \mid q \in \mathcal{Q}_P\}$  wird mit einer Kante verbunden, die dem Weg  $\Psi_{reg, i \rightarrow j}$  entspricht und daher das Gewicht  $\varepsilon_{path, i \rightarrow j}^2$  aufweist. Über  $\Psi_{reg, i \rightarrow j}$  sind die notwendigen Verknüpfungsposen für diesen Weg identifizierbar. Um weiterhin das „Loop Closure“-Problem zu vermeiden (Abschnitt 5.2), müssen vorhandene Kreise in  $G_{tree}$  entfernt werden. Um gleichzeitig den minimalen Messfehler zu erhalten, muss ein spannender Baum mit der kleinstmöglichen Summe aller Kantengewichte  $\sum \varepsilon_{path}^2$  im Graph  $G_{tree}$

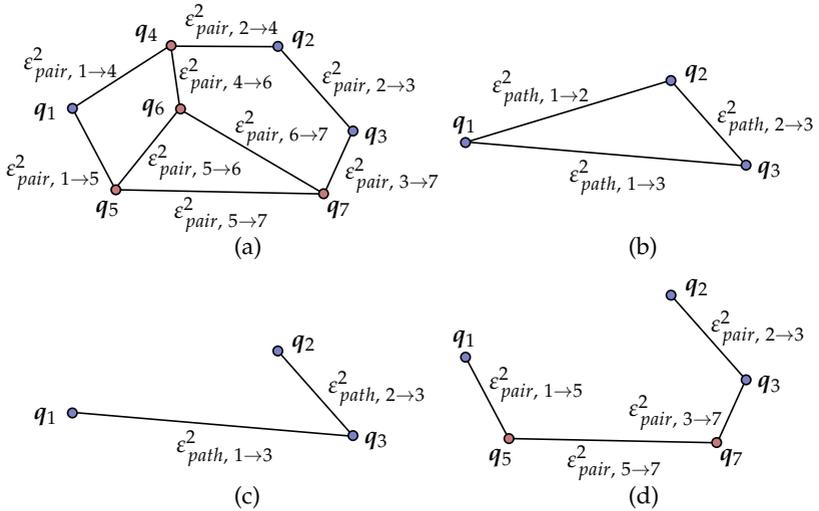


Abbildung 6.7: Die kürzesten Wege zwischen  $\{q_1, q_2, q_3\}$  in  $G_{reg}$  (a) entsprechen den Kantengewichten in  $G_{tree}$  (b). Ein minimaler Spannbaum in  $G_{tree}$  bildet den Registrierungsbaum (c), welcher final um die notwendigen Verknüpfungsposen  $\{q_5, q_7\}$  erweitert wird (d).

gefunden werden. Dies entspricht einem minimalen Spannbaum, dargestellt in Abbildung 6.7.c. (Abschnitt 4.2.3).

Um  $G_{tree}$  entsprechend zu reduzieren, stehen die Algorithmen von Prim und Kruskal zur Verfügung (HOROWITZ et al. 1998, S. 218). Beide garantieren eine optimale Lösung, unterscheiden sich jedoch in der benötigten Berechnungszeit. Durch die paarweise Kantenbildung zwischen je zwei Prüfposen ist der Graph  $G_{tree}$  vollständig verbunden. Für den vorliegenden Fall wird dem Verfahren von Prim der Vorzug gegeben, da dieser bei Graphen mit einer verhältnismäßig großen Anzahl von Kanten potentiell kürzere Berechnungszeiten aufweist (KRUMKE & NOLTEMEIER 2012, S. 128).

Zuletzt wird  $G_{tree}$  um die verwendeten Verknüpfungsposen erweitert, indem jede Kante zwischen zwei Prüfposen  $\{q_i, q_j\}$  durch den entsprechenden kürzesten Weg  $\Psi_{reg, i \rightarrow j}$  aus  $G_{reg}$  mit  $\{q_i, q_j\} := \Psi_{reg, i \rightarrow j}$  substituiert wird (Abbildung 6.7.d). Die Prüf- und Verknüpfungsposen in  $G_{tree}$  bilden zuletzt die

## 6 Automatisierte Bahnplanung

---

Menge relevanter Posen  $Q_{pert} \subset Q_{free}$ , die das Messsystem zur Durchführung der Prüfaufgabe erreichen muss.

### 6.6 Generierung einer kollisionsfreien Roboterbahn

Im letzten Teilschritt werden die zur Durchführung der Messaufgabe notwendigen Posen  $Q_{pert}$  zu einer räumlichen Trajektorie des Roboters verbunden. Der entstehende Pfad darf dabei für das System zu keiner Kollision mit sich, dem Prüfobjekt oder der umgebenden Arbeitszelle führen. Durch den Baum  $G_{tree}$  ist festgelegt, welche Punktwolken während der inkrementellen Registrierung zueinander auszurichten sind. Die zugehörigen Posen müssen jedoch nicht zwangsweise entlang der Kanten dieses Baumes vom Roboter angefahren werden. Die Ausrichtung der Punktwolken kann und wird vielmehr im Anschluss der Aufnahme-prozedur und damit unabhängig von der Bewegung des Roboters durchgeführt. Die Bewegungsbefehle können daher so angeordnet werden, dass sich eine möglichst kurze Fahrzeit ergibt und damit die Belegungszeit der Messanlage minimiert wird. Ziel der dritten Phase ist damit eine kollisionsfreie Roboterbahn  $\Psi_{TSP}$ , die jede relevante Pose mindestens einmal durchläuft. Als Optimierungskriterium dient die Fahrzeit der handhabenden Kinematik.

#### 6.6.1 Auswahl der Bahnplanungsmethode

Für die Planung einer Roboterbahn ist zunächst der Aufbau des Konfigurationsraums nötig, der mit  $G_{sample}$  aus Abschnitt 6.4 bereits teilweise abgebildet ist. Kombinatorische Methoden zur Bahnerzeugung finden hierbei die optimale Lösung, erfordern jedoch einen vollständigen Konfigurationsraum. Dazu muss für jede mögliche Roboterpose die Kollisionsfreiheit geprüft werden. In Vorversuchen konnte die Berechnungszeit für den Aufbau einer Szene zur Kollisionsdetektion abgeschätzt werden, die aus dem Messsystem in einer definierten Pose sowie sämtlichen Hindernissen im Arbeitsraum besteht. Die im industriellen Umfeld eingesetzte Simulationsumgebung „3DCreate 2014“<sup>1</sup>

---

<sup>1</sup><http://www.visualcomponents.com/de/produkte/3dcreate/>

---

benötigte im Schnitt 70 ms pro Szene, während eine eigene Implementierung auf Basis der freien Physik-Engine „Bullet Physics“<sup>2</sup> etwa 25 ms in Anspruch nahm. Beim Aufbau des Konfigurationsraums muss jede eventuell relevante Pose geprüft werden. Ausgehend von der dargestellten Anwendung eines Industrieroboters mit sechs unabhängigen Drehgelenken, einer Einschränkung der Achswinkel auf den Arbeitsbereich und einer adäquaten Diskretisierung der Drehgelenke von 1° ergeben sich 3 Milliarden zu testende Szenen. Für beide Implementierungen ergibt sich damit eine Berechnungszeit von mehreren hundert Tagen.

Da zusätzlich jede Änderung des Prüfobjekts oder anderer Hindernisse im Arbeitsraum eine vollständige Neuberechnung erfordert, ist der Aufbau des vollständigen Konfigurationsraums nicht praktikabel. Es wird daher auf probabilistische Verfahren zur Bahnplanung zurückgegriffen (LAVALLE 2011). Hierbei wird der Konfigurationsraum partiell erweitert, bis eine Lösung gefunden wird. Es ist zu beachten, dass die geplanten Bahnen nicht das optimale Ergebnis darstellen. Sie sind stets ein Kompromiss aus notwendiger Berechnungszeit und erreichbarer Prozesszeit. Da in der praktischen Anwendung potentiell eine große Anzahl an Prüfposen verbunden werden muss, ist der Einsatz eines Single-Query-Planers nicht zielführend (CHOSSET et al. 2005, S. 227). Es wird daher zur Lösung der Bahnplanungsaufgabe mit der „Probabilistic Roadmap Method“ ein Multi-Query-Planer herangezogen.

Die PRM besteht aus zwei Phasen, dem Aufbau der Roadmap mit Hilfe eines lokalen Planers sowie dem Finden der gesuchten Pfade. In der durch KAVRAKI et al. (1996) vorgestellten Form wird dabei jeweils der Weg zwischen zwei Ecken gesucht. Im Gegensatz dazu erfordert der vorliegende Anwendungsfall das Verbinden sämtlicher relevanten Posen zu einem Gesamtpfad. Diese Aufgabe kann abstrahiert als „Travelling Salesman Problem“ aufgefasst werden (Abschnitt 4.2.3). Zur Lösung der Bahnplanungsaufgabe wird daher zunächst eine Roadmap analog zur PRM erzeugt, in der anschließend die optimale Reihenfolge der benötigten Posen durch eine Approximation des TSP gesucht wird.

---

<sup>2</sup><http://bulletphysics.org>

### 6.6.2 Aufbau der Roadmap

Für den Aufbau der Roadmap  $G_{rob}$  werden sämtliche relevanten Posen  $Q_{pert}$  als Ecken in den Graph eingefügt. Die Kanten bilden kollisionsfreie Pfade zwischen diesen Posen und werden durch den lokalen Planer erzeugt.

#### Wahl der Abstandsmetrik

Zum Verbinden der Ecken  $\{q_i, q_j\}$  im Graph  $G_{rob}$  muss zunächst eine Metrik für die Distanz  $D_{PRM}(q_i, q_j)$  definiert werden. Des Weiteren muss die Gewichtung  $\theta(q_i, q_j)$  spezifiziert werden. Nachfolgende Algorithmen zur Pfadsuche minimieren die Distanz bezogen auf diese Gewichtung. Das zu optimierende Kriterium, die Fahrzeit des Roboters  $t_{robot}$ , sollte daher proportional in die Metrik  $D_{PRM}$  und die Gewichtung  $\theta$  eingehen.

KAVRAKI et al. (1996) nutzen als Metrik  $D_{PRM}$  in der ursprünglichen Form der PRM den euklidischen Abstand zwischen den betrachteten Ecken. Um mathematisch eindeutige Posen beschreiben zu können, werden Bewegungen des Roboters jedoch im Konfigurationsraum abgebildet (Abschnitt 4.2.1). Abstände im Konfigurationsraum sind dabei nicht proportional zum Abstand im euklidischen Raum, wodurch diese Metrik für den vorliegenden Fall nicht zielführend ist. AMATO et al. (1998a) vergleichen den Einsatz weiterer Distanzmetriken, bei denen die Drehgelenke unterschiedlich gewichtet werden oder der euklidische Abstand der Massenschwerpunkte einzelner Achselemente verwendet wird. Die Proportionalität zum Optimierungskriterium Fahrzeit ist hierbei nur gegeben, wenn entsprechende Gewichtungsfaktoren für den gewählten Roboter bestimmt werden können.

Im vorliegenden Anwendungsfall soll die Fahrzeit des Roboters minimiert werden. Es liegt daher nahe, die Fahrzeit selbst als Abstandsmetrik zu nutzen. Dabei muss zwischen den in Abschnitt 2.2.1 beschriebenen Bewegungsarten linear und PTP unterschieden werden. Während einer linearen Bewegung werden die Drehgelenke derart gesteuert, dass sich für den Endeffektor eine Gerade im kartesischen Raum ergibt. Bei der Bewegungsart PTP hingegen wird die minimale Drehbewegung der Achsen zur Zielkonfiguration genutzt. Da diese Bewegungsart potentiell schneller ist und der vorliegende Anwendungsfall

keine definierte Bewegung zwischen den Posen voraussetzt, werden ausschließlich PTP-Bewegungen geplant. Ferner wird die mechanische Belastung durch simultane Beschleunigungs- und Bremsvorgänge verringert, wodurch es sich um synchrone PTP-Bewegungen handelt. (POTT et al. 2013, S. 327; WEBER 2009, S. 79)

Zur Berechnung der Fahrzeit werden bei einer PTP-Bewegung die Achswinkel der Start- und Zielkonfiguration betrachtet, ohne Berücksichtigung des Endeffektors. Jede Achse  $i$  des Roboters mit  $m$  Drehgelenken muss dabei eine Winkeldifferenz  $\Delta\varphi_i$  überwinden. Die hierfür notwendige Zeit  $t_i$  hängt von der maximalen Drehgeschwindigkeit  $\dot{\varphi}_{max}$ , Beschleunigung  $\ddot{\varphi}_{acc}$  sowie Bremsleistung  $\ddot{\varphi}_{dec}$  der entsprechenden Achse ab und lässt sich aus Gleichung 6.6 ermitteln. Die achsspezifischen Kennwerte  $\dot{\varphi}_{max}$ ,  $\ddot{\varphi}_{acc}$  und  $\ddot{\varphi}_{dec}$  liefert in der Praxis das jeweilige Datenblatt des Industrieroboters. Die Fahrzeit der Bewegung  $t_{PTP}$  wird anschließend nach Gleichung 6.7 durch die Leitachse bestimmt, da deren Bewegung am längsten dauert und somit die Fahrzeit vorgibt.

$$t_i(\Delta\varphi) = \begin{cases} \frac{\dot{\varphi}_{max}}{2\ddot{\varphi}_{acc}} + \frac{\dot{\varphi}_{max}}{2\ddot{\varphi}_{dec}} + \frac{\Delta\varphi}{\dot{\varphi}_{max}} & , \text{ falls } \Delta\varphi > \frac{\dot{\varphi}_{max}^2}{2\ddot{\varphi}_{acc}} + \frac{\dot{\varphi}_{max}^2}{2\ddot{\varphi}_{dec}} \\ \sqrt{\frac{2\ddot{\varphi}_{dec}\Delta\varphi}{\ddot{\varphi}_{acc}(\ddot{\varphi}_{acc}+\ddot{\varphi}_{dec})}} + \sqrt{\frac{2\ddot{\varphi}_{acc}\Delta\varphi}{\ddot{\varphi}_{dec}(\ddot{\varphi}_{acc}+\ddot{\varphi}_{dec})}} & , \text{ sonst} \end{cases} \quad (6.6)$$

$$t_{PTP} = \max_{i \in \{1, \dots, m\}} (t_i) \quad (6.7)$$

Da eine minimale Prüfzeit angestrebt wird, stellt die Fahrzeit  $t_{PTP}(q_i, q_j)$  des Roboters ein geeignetes Maß sowohl für die Metrik  $D_{PRM}$  als auch zur Gewichtung der Kanten  $\theta$  dar. Es gelten daher die Festlegungen nach Gleichung 6.8 und Gleichung 6.9. Der kürzeste Pfad zwischen zwei Posen im Graph  $G_{rob}$  entspricht dadurch der minimalen Fahrzeit.

$$D_{PRM}(q_i, q_j) := t_{PTP}(q_i, q_j) \quad (6.8)$$

$$\theta(q_i, q_j) := t_{PTP}(q_i, q_j) \quad (6.9)$$

### Der lokale Planer

Der lokale Planer erzeugt Kanten im Graph  $G_{rob}$ . Dazu werden die nächsten Nachbarn einer Ecke bestimmt, wobei die gewählte Abstandsmetrik  $t_{PTP}(q_i, q_j)$  genutzt wird. Zwischen den ausgewählten Ecken und jedem Nachbar wird anschließend die direkte Verbindung  $\{q_i, q_j\}$  auf Kollisionsfreiheit geprüft. Durch Beschränkung auf die nächsten Nachbarn wird die Berechnungszeit zur Erstellung der Roadmap begrenzt. Zusätzlich wird die Wahrscheinlichkeit einer kollisionsfreien Verbindung mit steigendem Abstand geringer (KAVRAKI et al. 1998). Die optimale Anzahl zu betrachtender Nachbarn wird in einer Parameterstudie ermittelt (Abschnitt 7.2).

Generell nimmt der lokale Planer etwa 90% und damit einen Großteil der gesamten Berechnungszeit beim Aufbau der Roadmap in Anspruch (SÁNCHEZ & LATOMBE 2003). Es ist daher auf eine effiziente Ausgestaltung des Algorithmus zu achten. GERAERTS & OVERMARS (2004a) vergleichen hierfür zwei verschiedene Ansätze. Bei der inkrementellen Methode wird der zu testende Pfad in einem festgelegten Abstand von der Start- bis zur Zielpose geprüft, wie in Abbildung 6.8.a schematisch dargestellt. Die binäre Methode in Abbildung 6.8.b hingegen halbiert den Pfad und testet die Pose, welche sich mittig zwischen Start- und Zielpose befindet. Anschließend werden jeweils die Teilstrecken mittig geprüft, bis diese eine Mindestlänge unterschreiten. Die Studie von GERAERTS & OVERMARS (2004a) empfiehlt den Einsatz der binären Methode, da Kollisionen schneller detektiert werden.

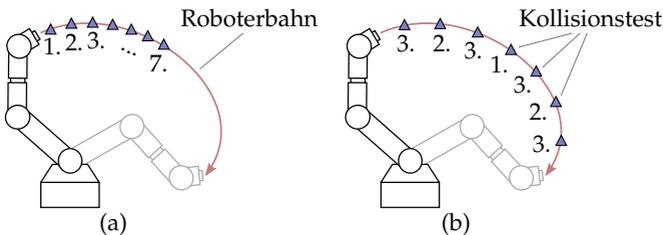


Abbildung 6.8: Bei der Kollisionsprüfung wird eine Roboterbewegung in diskrete Posen aufgeteilt. Dies kann inkrementell (a) oder binär (b) erfolgen.

Auch im vorliegenden Anwendungsfall wird daher die Suche in einem Binärbaum genutzt, indem bezogen auf die Achswinkel des Roboters jeweils die

---

Mitte der zugehörigen Posen betrachtet wird. Bei Auftreten einer Kollision bricht der lokale Planer ab und gibt ein Negativergebnis zurück. Wird der minimale Abstand zwischen den geprüften Posen unterschritten, bricht der Algorithmus hingegen mit positivem Ergebnis ab und die Verbindung gilt im Rahmen des gewählten Minimalabstands als kollisionsfrei. Ist das Ergebnis des lokalen Planers  $\Lambda(q_i, q_j)$  positiv, wird eine Kante  $\{q_i, q_j\}$  mit dem Gewicht  $\theta(q_i, q_j)$  der Roadmap  $G_{rob}$  hinzugefügt.

Für sämtliche Posen in  $G_{rob}$  werden die nächsten Nachbarn bestimmt und zugehörige Kanten durch den lokalen Planer erzeugt. Anschließend wird geprüft, ob die Roadmap zusammenhängend ist. Dazu muss jede Pose von jeder anderen Pose erreicht werden können. Es muss demnach eine Folge von Kanten existieren, die beide Ecken verbinden. Die Existenz dieser Wege ist eine notwendige Bedingung zur Lösung der Bahnplanungsaufgabe. Solang die Roadmap  $G_{rob}$  nicht zusammenhängend ist, muss sie weiter verdichtet werden.

### **Verdichtung der Roadmap**

Die Verdichtung der Roadmap wird durch zusätzliche, kollisionsfreie Posen  $q \in \mathcal{Q}_{free}$  erreicht, die den Bewegungsraum des Roboters erweitern. Ein Ansatz hierzu ist das Hinzufügen zufälliger Konfigurationen im Arbeitsraum, wodurch das Verfahren probabilistisch vollständig wird (ULRICH et al. 2016). Alternativ können komplexere Methoden zur gezielten Erstellung der Posen an voraussichtlich problematischen Bereichen (z. B. Engstellen) implementiert werden (BOOR et al. 1999; HSU et al. 2003). Als dritter Ansatz stehen Single-Query-Planer wie die durch LAVALLE (1998) vorgestellten „Rapidly-Exploring Random Trees“ zur Verfügung, die zum Verbinden von zwei separierten Bäumen im Graph genutzt werden können (LAVALLE & KUFFNER 2001).

Generell ist von einer hohen Verbindungsdichte der Roadmap  $G_{rob}$  auszugehen. Die generierten Posen haben verfahrensbedingt stets Nachbarn, mit denen sie bei Aufnahmen den Sichtbereich teilen. Dadurch ist der euklidische Abstand zwischen diesen Posen in Relation zum Arbeitsraum klein und das Auftreten von Kollisionen damit unwahrscheinlicher (BOOR et al. 1999). Der Empfehlung von GERAERTS & OVERMARS (2004a) folgend wird von einer Implementierung komplexer Algorithmen zur Verdichtung abgesehen, da diese voraussichtlich

nicht notwendig sind. Zur Absicherung wird das iterative Hinzufügen zufälliger Konfigurationen  $q_{avoid} \in \mathcal{Q}_{free}$  nach ULRICH et al. (2016) benutzt, um eine zusammenhängende Roadmap sicherzustellen. Abbildung 6.9.a zeigt die aus dem Beispiel in Abbildung 6.7 hervorgegangene Roadmap  $G_{rob}$ .

### 6.6.3 Bestimmung der optimalen Reihenfolge

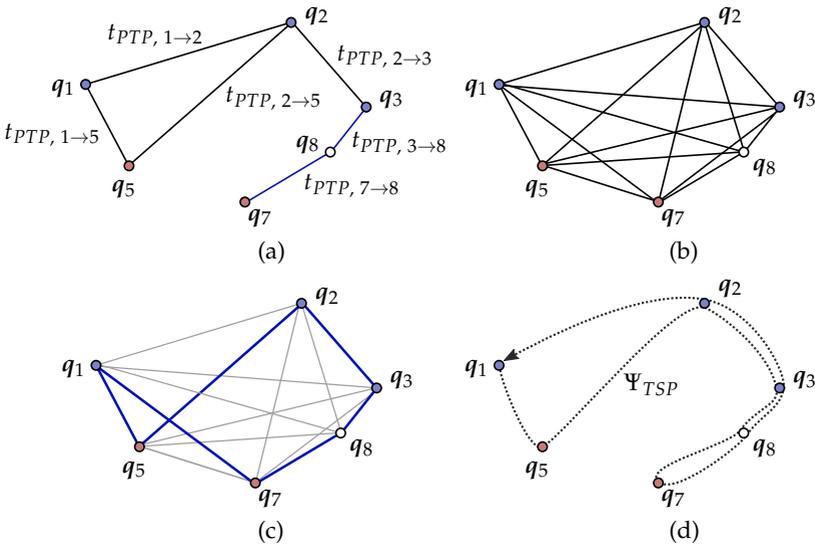


Abbildung 6.9: Mittels  $\mathcal{Q}_{pert}$ , dem lokalen Planer  $\Lambda$  und zufälligen Posen wie  $\{q_8\}$  entsteht  $G_{rob}$  (a). Dieser wird vollständig verbunden (b) und ein Hamilton-Kreis mit minimaler Distanz gesucht (c), woraus durch Einfügen der Zwischenposen der finale Pfad  $\Psi_{TSP}$  gebildet wird (d).

Im letzten Schritt wird die optimale Reihenfolge beim Anfahren der Posen bestimmt, indem ein Algorithmus zur approximativen Lösung des TSP herangezogen wird. Dabei sind eventuell eingefügte Ausweichposen  $\{q_{avoid}\}$  zur Erfüllung der Prüfaufgabe irrelevant und werden daher durch die direkte Kante zwischen den benachbarten Posen ersetzt. Die Gewichtung setzt sich dafür jeweils aus der Summe der beteiligten Kantengewichte zusammen.

---

Die optimale Reihenfolge spiegelt sich in einer minimalen Fahrzeit des Roboters wider, bei der jede relevante Pose angefahren wird. Dies entspricht im Graph einem möglichst kurzen Kreis bezogen auf die Kantengewichtung. Dieser Kreis kann mit Hilfe von Algorithmen zur Lösung des TSP gefunden werden. Mathematisch exakte Verfahren finden hierbei stets die optimale Lösung, die Berechnungszeit steigt jedoch überproportional mit der Anzahl der Ecken im Graph. Exakte Verfahren benötigen selbst bei effizienter Implementierung deutlich mehr Berechnungszeit, als für die vorliegende Anwendung vertretbar wäre (HUSSMANN 2015, S. 110). Es wird daher auf Heuristiken zurückgegriffen, welche zwar nicht zwangsweise die optimale Lösung finden, allerdings wesentlich kürzere Berechnungszeiten aufweisen.

Von den bisher erfolgreich implementierten Heuristiken weist der Algorithmus nach Christofides die beste Güte auf (Abschnitt 4.2.3). Die Methode garantiert einen Kreis als Ergebnis, dessen kumuliertes Kantengewicht maximal 150 % des minimalen Gewichts entspricht. Dabei werden eine metrische Formulierung des Problems und ein vollständiger Graph vorausgesetzt. In einem vollständigen Graphen ist jede Pose mit jeder anderen Pose verbunden. Dies ist im vorliegenden Graphen nicht der Fall, da stets nur die nächsten Nachbarn durch den lokalen Planer  $\Lambda$  verbunden wurden. Durch den vorhergehenden Schritt ist allerdings ein zusammenhängender Graph sichergestellt, es existiert demnach mindestens eine kollisionsfreie Route zwischen jedem Paar von Posen.

Zur Vervollständigung wird daher zwischen jedem bisher nicht verbundenen Posen-Paar eine Kante erzeugt (Abbildung 6.9.b). Die Kanten sind hierbei die kürzesten Wege  $\Psi_{rob, i \rightarrow j}$  im Graph  $G_{rob}$  zwischen den entsprechenden Posen  $\{q_i, q_j\}$ . Der kürzeste Weg wird mit Hilfe des A\*-Algorithmus gefunden und als Kante hinzugefügt. Die Gewichtung ergibt sich aus der Summe der im Weg  $\Psi_{rob, i \rightarrow j}$  enthaltenen Kantengewichte. Der A\*-Algorithmus schätzt dabei die Entfernung zum Ziel wiederum mit  $\hat{v} = t_{PTP}$ .

Der Graph muss außerdem für die Anwendung der Christofides-Heuristik metrisch aufgebaut sein. Metrisch ist das vorliegende TSP, wenn im zugehörigen Graphen  $G_{rob}$  die Dreiecksungleichung gilt. Der Weg zwischen zwei Posen muss demnach stets kürzer sein als der Umweg über eine dritte Pose. Für drei beliebige Posen  $\{q_i, q_j, q_m\}$  in  $G_{rob}$  und der Metrik  $t_{PTP}$  muss daher Gleichung 6.10

## 6 Automatisierte Bahnplanung

---

erfüllt sein. Mit dem Einfügen der garantiert kürzesten Pfade  $\Psi_{rob, i \rightarrow j}$  als Kanten ist dies gegeben. Die direkte Verbindung ist stets der kürzest mögliche, der Umweg über weitere Posen kann niemals kürzer sein. Zuletzt muss das TSP symmetrisch sein. Da die Richtung der Roboterbewegung unerheblich für die Fahrzeit ist und  $t_{PTP}(q_i, q_j) = t_{PTP}(q_j, q_i)$  gilt, wird auch diese Bedingung erfüllt.

$$t_{PTP}(q_i, q_j) \leq t_{PTP}(q_i, q_m) + t_{PTP}(q_m, q_j) \quad (6.10)$$

Die Roadmap  $G_{rob}$  ist damit vollständig und metrisch. Durch Anwendung der Christofides-Heuristik wird ein Hamilton-Kreis gefunden (Abbildung 6.9.c). In dem gefundenen Pfad werden zuletzt Kanten  $\Psi_{rob, i \rightarrow j}$ , die einen Umweg über Zwischenposen repräsentieren, durch die ursprünglichen Kanten und Posen ersetzt. Die entstehende Roboterbahn  $\Psi_{TSP}$  (Abbildung 6.9.d) ist sowohl frei von Kollisionen, als auch fahrzeitoptimiert im Rahmen der angewendeten Heuristik.

### 6.6.4 Ablauf zur Generierung der Roboterbahn

Das in Abbildung 6.10 dargestellte Ablaufdiagramm visualisiert die Teilschritte bei der Suche nach einem Roboterpfad. Je nach mechanischem Aufbau des Robotersystems ist vor den Prüfposen eine systemspezifische Wartezeit zu implementieren, um die Aufnahme nicht durch auftretende Schwingungen des mechanischen Aufbaus zu beeinflussen.

## 6.7 Anpassung auf den konkreten Anwendungsfall

Für jedes Prüfobjekt muss das Verfahren zur Erzeugung der Roboterbahn von Neuem durchlaufen werden. Für den Offline-Anwendungsfall, der durch häufige Wechsel der Prüfobjekte und wenige Prüfzyklen pro Prüfobjekt geprägt ist, bedeutet dies einen großen Einfluss der Planungszeit auf die Prozesszeit. Das Generieren eines Roboterprogramms benötigt im Regelfall wesentlich mehr Zeit als ein einzelner Prüfzyklus. Wird das System hingegen Inline verwendet,

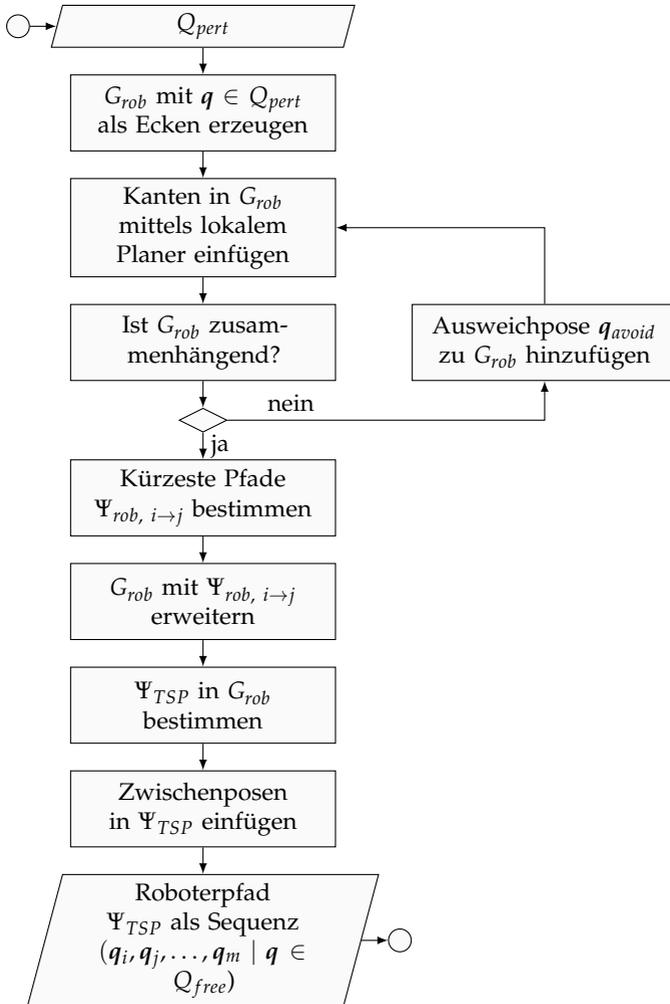


Abbildung 6.10: Ablauf zur Erzeugung eines kollisionsfreien und fahrzeitoptimierten Roboterpfads. Basis ist eine Menge  $Q_{pert}$  mit kollisionsfreien Posen, welche erreicht werden müssen.

## 6 Automatisierte Bahnplanung

---

bedeutet dies eine höhere Priorität der Prozesszeit. Die Prüfmerkmale werden dabei wiederholt gemessen, eine Änderung auf andere Prüfobjekte erfolgt selten. Der Einfluss von Prozesszeiten steigt daher im Verhältnis zur Planungszeit.

Wird die Prozesszeit betrachtet, entfällt ein wesentlicher Teil auf die Aufnahme von Verknüpfungsposen und die Berechnung der inkrementellen Registrierung. Die Registrierung der Prüfposen wird dadurch von der Genauigkeit des Industrieroboters unabhängig. Der Roboter weicht im Bereich der Pose-Genauigkeit, demnach im Millimeterbereich ab. Werden die Prüfposen wiederholt angefahren, streuen die Werte zusätzlich im Bereich der Wiederholgenauigkeit (Abschnitt 2.2.1). Diese liegt bei aktuell im Einsatz befindlichen Industrierobotern im Bereich von etwa  $60\ \mu\text{m}$  (ABB AUTOMATION GMBH 2014b; FANUC DEUTSCHLAND GMBH 2017).

Eine durch die begrenzte Pose-Genauigkeit bedingte Abweichung entspricht dabei einem systematischen Fehler und kann daher durch die Methode des 3D-Image-Stitching kompensiert werden. Dazu wird in einem initialen Prüfzyklus unter Nutzung der inkrementellen Registrierung die systematische Abweichung des Roboters an den Prüfposen bestimmt. Die Registrierung kann in nachfolgenden Prüfzyklen mit Hilfe der Selbstlokalisierung des Roboters durch Vorwärtstransformation und der durch 3D-Image-Stitching bestimmten Kompensation im Bereich der Wiederholgenauigkeit des Roboters erfolgen. Vorteilhaft ist dabei eine potentiell geringere Prozesszeit, da auf die Erzeugung von Aufnahmen an den Verknüpfungsposen und die Ausrichtung der Punktwolken während des Prüfzyklus verzichtet werden kann. Dieses Vorgehen kann für Anwendungsfälle mit gleich bleibenden Prüfobjekten, sich stetig wiederholenden Prüfaufgaben und hohen Anforderungen an die Prozesszeit in Betracht gezogen werden.

Die Verknüpfungsposen selbst bleiben dabei Teil der Roboterbahn, um einen kollisionsfreien Pfad gewährleisten zu können. Zusätzlich ist ein Anfahren der Messposen stets aus der gleichen Richtung zu bevorzugen, da Ungenauigkeiten durch Getriebeispiel auf diese Weise partiell ausgeglichen werden (ULRICH et al. 2014). Negativ zu bewerten ist die potentiell längere Fahrzeit im Vergleich zu Bahnen, die nicht mit den restriktiven Bedingungen des 3D-Image-Stitching geplant wurden. Diese potentiell höhere Prozesszeit ist daher ein Kompromiss, um den erwarteten Messfehler zu minimieren.

---

## 6.8 Zusammenfassung der Bahnplanung

Zur automatisierten Durchführung der Prüfaufgabe wird eine Sequenz von PTP-Anweisungen für das roboterbasierte Messsystem generiert. An den angefahrenen Posen können dabei für die Prüfung notwendige Punktwolken durch den 3D-Sensor erfasst werden. Abbildung 6.11 zeigt die Teilschritte, die hierfür notwendig sind. Zunächst wird eine Abtastung  $G_{sample}$  des freien Konfigurationsraums  $\mathcal{Q}_{free}$  erzeugt, indem geeignete Prüf- und Verknüpfungsposen generiert werden. Anschließend wird ein Registrierungsbaum  $G_{tree}$  aus sich überlagernden Aufnahmen ermittelt. Darin enthaltene Posen werden unter  $\mathcal{Q}_{pert}$  zusammengefasst und sind zur Durchführung der Prüfaufgabe notwendig. Diese werden zuletzt zu einer kollisionsfreien Roboterbahn  $\Psi_{TSP}$  verbunden. Durch Optimierung der Reihenfolge wird dabei die Fahrzeit des handhabenden Systems minimiert.

Nach Abfahren der notwendigen Posen und Aufnahme der Punktwolken werden diese mit dem in Abschnitt 5.4 aufgezeigten Algorithmus zueinander ausgerichtet. Welche Aufnahmen miteinander verknüpft werden, ergibt sich aus dem erzeugten Registrierungsbaum  $G_{tree}$ .

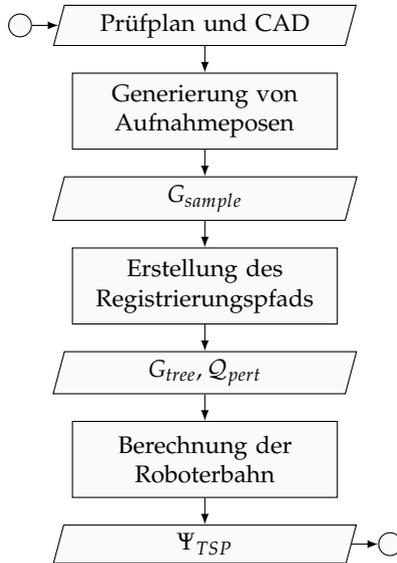


Abbildung 6.11: Ablauf zur Erstellung des Prüfprogramms.

## **7 Validierung und Evaluation**

### **7.1 Prototypische Implementierung**

Um die Machbarkeit des Verfahrens aufzuzeigen, wird 3D-Image-Stitching prototypisch umgesetzt. Anhand dieser Implementierung erfolgt anschließend die Validierung in Form eines experimentellen Anwendungsbeispiels.

#### **7.1.1 Mechanischer und elektrischer Aufbau**

Das roboterbasierte Messsystem bestehend aus Industrieroboter, 3D-Sensor sowie Projektoren ist in einer Messzelle gekapselt (Abbildung 7.1). Unterhalb des Arbeitsbereichs, in der Basis des Aufbaus, sind die elektrische Versorgung, Robotersteuerung und ein PC mit „Intel Xeon E3-1271 v3 Prozessor“ (4 Kerne mit je 3,6 GHz) und 16 GB RAM für Berechnungen untergebracht. Das Bedienfeld in Form eines berührungsempfindlichen Bildschirms befindet sich von außen zugänglich an der Messzelle. Zusätzlich steht ein weiterer Bildschirm zur Anzeige von relevanten Informationen zur Verfügung.

Der Roboter ist auf einem Stahlblock fixiert und über die Grundplatte aus Stahl mit den verbleibenden Komponenten verbunden. Eine Übertragung von niederfrequenten Schwingungen während Beschleunigungs- und Bremsvorgängen des Roboters wird damit auf ein Minimum reduziert. Die Profilstreben und die Verkleidung der Basis bestehen aus Aluminium. Der Arbeitsbereich des Roboters ist mit Scheiben aus Polycarbonat umgeben. Diese lassen sich öffnen, die Bewegung des Roboters wird dann durch gekoppelte Sicherheitsschalter blockiert. Die Scheiben wurden außerdem mit für infrarotes Licht undurchlässigen Folien bezogen. Die Einhausung dient damit sowohl der Arbeitssicherheit als auch der Abschirmung des Messsystems von Emissionsquellen der Umgebung im Infrarotbereich.

## 7 Validierung und Evaluation

---

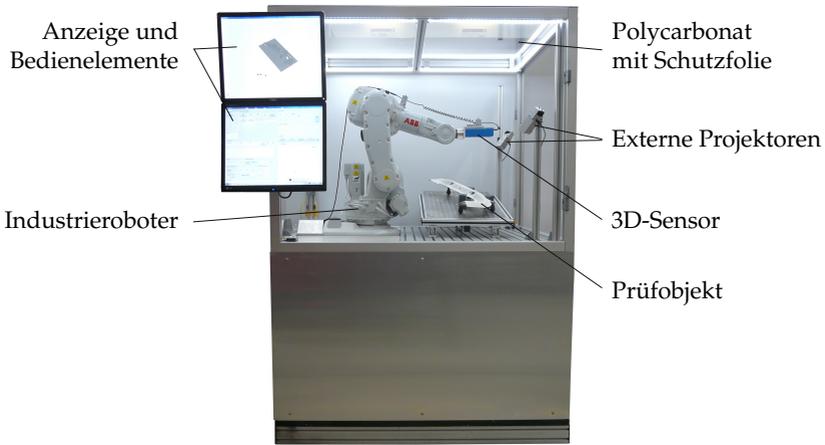


Abbildung 7.1: Prototypische Umsetzung eines roboterbasierten Messsystems auf Basis von 3D-Image-Stitching. Die seitlichen Scheiben wurden zur besseren Darstellung entfernt.

Zur Führung des 3D-Sensors kommt ein „IRB 140 Industrieroboter“ von ABB AUTOMATION GMBH (2014a) zum Einsatz. Der Aufbau entspricht mit sechs seriell angeordneten Drehgelenken dem eines Vertikal-Knickarmroboters (Abschnitt 2.2.1). Im sphärischen Arbeitsbereich besitzt er eine Reichweite von 810 mm und begrenzt damit die maximale geometrische Größe potentieller Prüfobjekte. Die Traglast beträgt 6 kg, die Wiederholgenauigkeit wird mit 30  $\mu\text{m}$  angegeben. Gesteuert wird der Roboter per Ethernet-Schnittstelle, die manuelle Eingabe von Bewegungsdaten ist beim Einsatz als roboterbasiertes Messsystem weder bei der Planung des Prüfablaufs noch bei der Durchführung von Prüfzyklen vorgesehen.

Als 3D-Sensor wird eine „N10-1210-18 Stereo-3D-Kamera“ von IDS IMAGING DEVELOPMENT SYSTEMS GMBH (2017) eingesetzt. Zur Generierung der Punktwolken wird das Prinzip der passiven Triangulation verwendet (Abschnitt 2.1.1). Zwei Kameras betrachten dafür eine Szene aus verschiedenen Winkeln und berechnen über die bekannte geometrische Relation zueinander die Tiefe der einzelnen Bildpunkte. Um homogene Oberflächen vollständig erfassen zu können, wird zusätzlich ein stochastisches Muster durch einen internen Projektor aufgebracht. Dadurch wird die Zuordnung korrespondierender Bereiche in der

---

Szene erleichtert. Der Sichtbereich entspricht einem Pyramidenstumpf mit einer Höhe von 30 mm. Im optimalen Abstand zum Prüfbjekt von 280 mm wird eine Fläche von 85 mm × 61 mm erfasst. Der laterale Abstand der Bildpunkte beträgt dann 0,13 mm mit einer optischen Unschärfe von 0,7 Bildpunkten und einer Auflösung in der Tiefe von 90 µm. Der 3D-Sensor arbeitet im Infrarotbereich von rund 850 nm. Die eingebauten Kameras nehmen daher nur Licht mit dieser Wellenlänge auf und der interne Projektor emittiert in diesem Bereich. Eine derartige Einschränkung erhöht die Robustheit gegenüber Störungen durch Umgebungslicht. Die dennoch in geringem Maße vorhandenen Infrarotanteile des Umgebungslichts werden mit Hilfe von Schutzfolien auf der Einhausung gefiltert, die den gesamten Arbeitsraum abschirmen.

Um die erfassten Punktwolken verknüpfen zu können, wird die Oberfläche des Prüfbjckts durch Projektoren mit zusätzlichen Informationen angereichert. In Abgrenzung zum internen Projektor des 3D-Sensors werden diese im Folgenden als externe Projektoren bezeichnet. Die Konzeption in Abschnitt 5.3 sieht einen statischen Aufbau mit Hilfe von LED-basierter Lichtquelle vor. Es wurden dazu die in Abbildung 7.1 sichtbaren Projektionseinheiten konstruiert. Das Prinzip des Aufbaus entspricht einem Durchlichtprojektor mit Lichtquelle, zwei Kondensorlinsen, einem Objektiv sowie einer partiell lichtdurchlässigen Folie, die das projizierte Muster definiert (NAUMANN et al. 2014, S. 275). Als Lichtquelle dient eine „OSLON Black Series SFH 4715S Infrarot-LED“ von OSRAM OPTO SEMICONDUCTORS GMBH (2016) mit 1030 mW Strahlungsleistung. Die auftretende Wärmeleistung wird über ein Kühlelement aus Aluminium abgeleitet. Die in Relation zum gesamten Messsystem geringen Investitionskosten von etwa 100 € und der wenig komplexe Aufbau ermöglichen die Verwendung von mehreren Projektionseinheiten je nach Größe des Prüfbjckts. Dabei ist durch den Anwender sicherzustellen, dass die zu prüfende Oberfläche vollständig abgedeckt wird. Für das dargestellte Szenario werden zwei baugleiche externe Projektoren verwendet.

Bei simultan aktivierter interner und externer Projektion ergibt sich eine Überlagerung beider Muster in der Aufnahme. Die interne Projektion bewegt sich dabei relativ zum Prüfbjekt zwischen zwei Aufnahmen, die externe Projektion hingegen bleibt statisch. Eine zeitliche Steuerung der Projektoren ist notwendig, um eine Überlagerung der Muster zu vermeiden und die datengetriebene

Registrierung zu ermöglichen. Die Versorgung der externen Projektoren ist daher an die Robotersteuerung angeschlossen und kann über diese aktiviert und deaktiviert werden. Für die interne Projektion ist dies direkt über die Software-Schnittstelle des 3D-Sensors möglich.

### 7.1.2 Softwaretechnische Realisierung

Abbildung 7.2 zeigt die Struktur und Schnittstellen des Messsystems. Der Aufbau folgt dabei dem Modularitätsprinzip (VOGEL et al. 2009, S. 145). Das System ist dazu in einzelne Bausteine mit definierten Verantwortlichkeiten eingeteilt. Diese Module kommunizieren nach außen über festgelegte Schnittstellen, interne Daten sind gekapselt und anderen Modulen nicht zugänglich. Diesem Prinzip folgend wurden die einzelnen Systembausteine objektorientiert integriert (VOGEL et al. 2009, S. 155). Durch die modulare Struktur und spezifizierte Schnittstellen können einzelne Komponenten ausgetauscht werden, ohne dass die verbleibenden Softwarekomponenten und das grundlegende System angepasst werden müssen.

Eine übergeordnete Steuerung übernimmt die zentrale Verwaltung der Systemkomponenten. Für die Umsetzung dieses *Steuermoduls* wurde die Programmiersprache C# gewählt. Die Kommunikation mit den Modulen zur Aufnahme und Registrierung basiert auf dem „Transmission Control Protocol/Internet Protocol“ (TCP/IP). Zur effizienten Implementierung wird die Programmierbibliothek „ZeroMQ“<sup>1</sup> eingebunden. Die grafische Benutzeroberfläche ist Teil des Steuermoduls und bündelt die Anwendereingaben zur Steuerung des Systems sowie die Ausgabe der Prüfergebnisse. Zur Visualisierung dreidimensionaler CAD-Modelle wird außerdem das „Helix Toolkit“<sup>2</sup> genutzt.

Das Steuermodul ruft die einzelnen Module als „Threads“ auf. Zur Verringerung der Prüfzeit und effektiven Auslastung der Ressourcen sämtlicher Prozessoren erfolgt dabei eine parallele Bearbeitung in Form von „Multithreading“. Stehen beispielsweise zwei auszurichtende Punktwolken bereit, wird der Algorithmus zur paarweisen Verknüpfung durchlaufen, selbst wenn zu diesem

---

<sup>1</sup><http://zeromq.org/>

<sup>2</sup><http://www.helix-toolkit.org/>

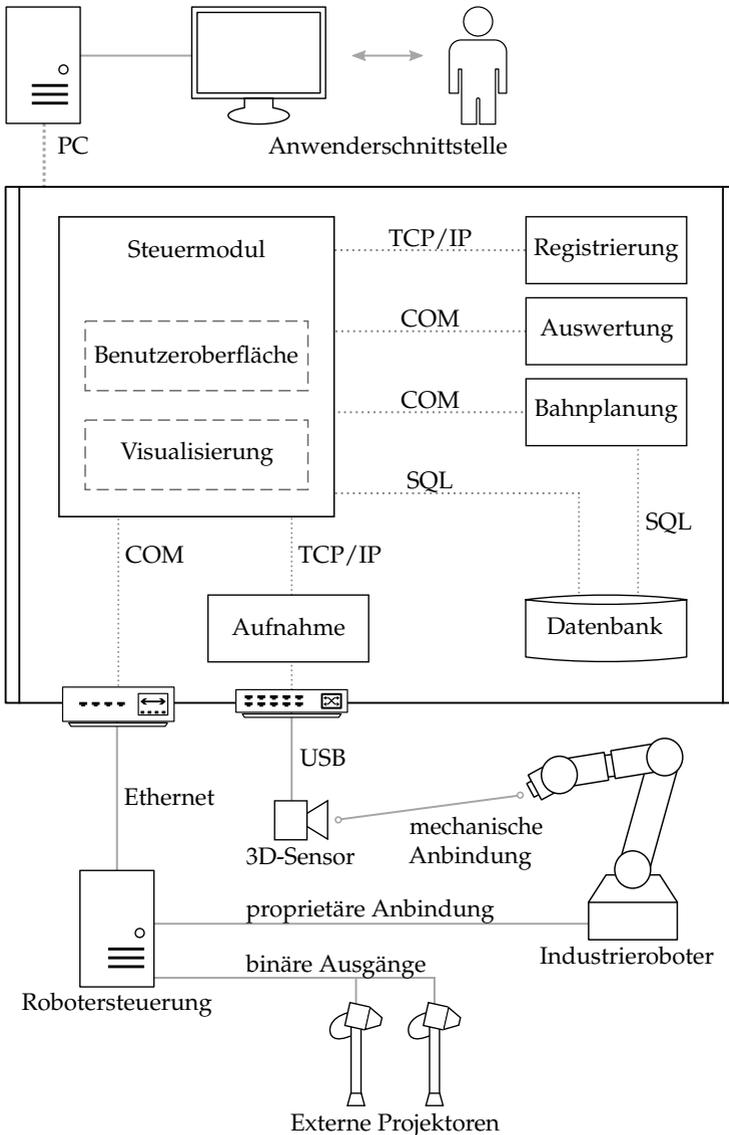


Abbildung 7.2: Systemstruktur des Messsystems: Modularer Aufbau und Kommunikationsschnittstellen zwischen den einzelnen Komponenten.

## 7 Validierung und Evaluation

---

Zeitpunkt die Aufnahme der verbleibenden Punktwolken noch nicht abgeschlossen ist. (ROTTMANN 2003, S. 415)

Mit der Robotersteuerung kommuniziert das Steuermodul über eine „Component Object Model“ (COM)-Schnittstelle. Diese sprachenunabhängige Programmierschnittstelle für „Microsoft Windows“ wird vom Hersteller des Roboters bereit gestellt. Neben der Positionsvorgabe werden auch die Projektoren über diese Schnittstelle aktiviert und deaktiviert, da diese direkt mit der Robotersteuerung verbunden sind.

Die Aufnahme der Punktwolken wird über das *Aufnahmemodul* realisiert. Implementiert ist die Funktionalität in der Programmiersprache C++. Da die interne Projektion des 3D-Sensors für die Berechnung der Tiefe essentiell ist, wird die Punktwolke zunächst mit aktiviertem internen Projektor erzeugt. Für die datengetriebene Registrierung darf jedoch einzig das extern aufgebrachte Muster sichtbar sein. Die Helligkeitswerte der Oberfläche werden daher anschließend aus einer Aufnahme mit deaktivierter interner und aktivierter externer Projektion gebildet. Zum Zwecke der Rauschminderung wird die Punktwolke dabei aus den Messdaten mehrerer Aufnahmen gemittelt (SÜSSE & RODNER 2014, S. 192).

Die datengetriebene Verknüpfung der Punktwolken ist im *Registrierungsmodul* gekapselt. Dessen Routinen bilden das Konzept nach Kapitel 5 ab und sind ebenfalls in C++ verfasst. Für die Implementierung des ICP-Algorithmus wird dabei auf Bestandteile der „Point Cloud Library“<sup>3</sup> zurückgegriffen.

Die in Kapitel 6 konzeptionierte automatische Erzeugung des Prüfprogramms wurde im *Bahnplanungsmodul* mittels „3DCreate 2014“<sup>4</sup> realisiert. Dieses Simulationsprogramm ist für die Offline-Programmierung von Industrierobotern ausgelegt (Abschnitt 2.2.2). Im vorliegenden Fall wird das Programm über die integrierte COM-Schnittstelle vom übergeordneten *Steuermodul* gesteuert und es werden vor allem der integrierte Kollisionsdetektor und die Berechnung der Rückwärtstransformation benutzt. Die Programmierung basiert dabei auf Python. Für die Verarbeitung von Graphen wird außerdem die Bibliothek

---

<sup>3</sup><http://pointclouds.org/>

<sup>4</sup><http://www.visualcomponents.com/de/produkte/3dcreate/>

---

„NetworkX“<sup>5</sup> eingebunden. Zur Kollisionsdetektion wird in „3DCreate 2014“ ein CAD-Modell der Messzelle aufgebaut, das den Hindernisraum repräsentiert (Abbildung 7.3). Durch die Kapselung des mechanischen Aufbaus in einer Messzelle ist dabei eine statische Modellierung der Hindernisse ausreichend, da sich während eines Prüfzyklus der Aufbau und die Anordnung der Komponenten zueinander nicht verändert.

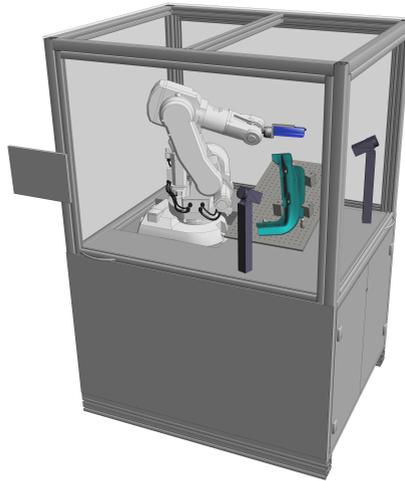


Abbildung 7.3: Für das Bahnplanungsmodul wurde mittels „3DCreate 2014“ ein CAD-Modell des Prototyps mit Prüfobjekt und Hindernisraum erzeugt.

Die Prüfdatenauswertung erfolgt im industriellen Umfeld durch eine Auswertesoftware (z. B. „PolyWorks“<sup>6</sup>). Für die prototypische Umsetzung wird hierfür ein *Auswertemodul* auf Basis von „Matlab 2016a“<sup>7</sup> eingesetzt und über die integrierte COM-Schnittstelle angesprochen.

Während der Prüfdatenerfassung fällt sowohl bei der Erstellung des Prüfprogramms als auch bei dessen Ablauf eine Vielzahl von Daten an, die geordnet und leicht zugänglich abgelegt werden müssen. Dazu wird das relationale

---

<sup>5</sup><https://networkx.github.io/>

<sup>6</sup><http://www.duwe-3d.de/de/Products/polyworks/>

<sup>7</sup><http://de.mathworks.com/products/matlab/>

Open-Source-Datenbanksystem „MySQL“<sup>8</sup> benutzt. Gespeichert werden unter anderem die in Abschnitt 6.3 beschriebenen Rasterpunkte während der Analyse des Prüfobjekts sowie die zugehörige Normalenrichtung und die in Abschnitt 6.6 ermittelte Sequenz der anzufahrenden Posen. Die *Datenbank* dient somit als globaler Datenspeicher, um berechnete Zwischenergebnisse abzulegen. Für Abfragen und somit als Schnittstelle zu anderen Modulen kommt SQL nach ISO/IEC 9075 zum Einsatz.

### 7.1.3 Bedienkonzept und Prüfablauf

Die Benutzerschnittstelle ist in Abbildung 7.4 dargestellt. Neben den zusammengefassten Prüfergebnissen der vergangenen Prüfzyklen werden Meldungen zum aktuellen Status des Messsystems angezeigt. Die Bedienung erfolgt über einen berührungsempfindlichen Bildschirm, daher wurde auf eine zweckmäßige Größe der Bedienelemente geachtet und auf textuelle Eingabe soweit wie möglich verzichtet. Neben dem weitestgehend automatischen Ablauf der Prüfdatenerfassung können die Systemkomponenten mit Hilfe der Bedienoberfläche zu Testzwecken manuell gesteuert werden. Ein zweiter Bildschirm dient der Anzeige des Prüfobjekts in Form des zugehörigen CAD-Modells. Angereichert wird die Darstellung durch eine Visualisierung der Prüfaufgabe sowie aktuell gemessener Werte für die geometrischen Merkmale.

Der in Abbildung 7.5 dargestellte Ablauf zeigt die Teilschritte während der Prüfdatenerfassung. Die manuellen Tätigkeiten beschränken sich dabei auf die Eingabe der Prüfaufgabe, die Positionierung des Prüfobjekts und die Ausrichtung der externen Projektoren sowie die zugehörige Anpassung des Modells der Messzelle. Die Komponenten der realen und virtuellen Messzelle müssen dabei im Rahmen der Möglichkeiten einer manuellen Ausrichtung deckungsgleich positioniert werden. Anschließend werden die Erstellung des zugehörigen Prüfprogramms, die Aufnahme der notwendigen Punktwolken, deren Registrierung und Auswertung sowie die Anzeige der Ergebnisse automatisch durchgeführt.

---

<sup>8</sup><https://www.mysql.de/>

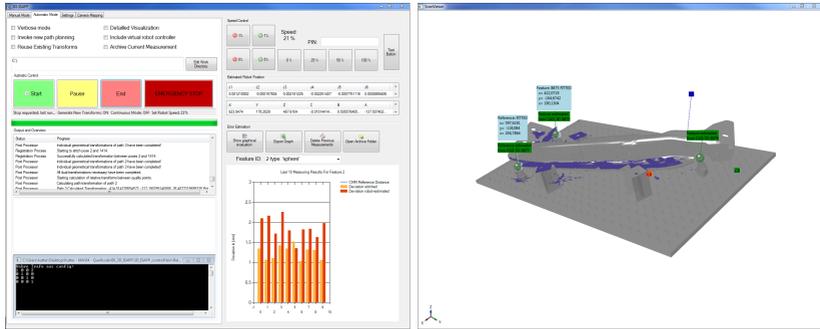


Abbildung 7.4: Über die Benutzerschnittstelle lassen sich sämtliche Arbeitsschritte und Komponenten des Prüfsystems steuern. Die Messergebnisse werden laufend am CAD-Modell des Prüfobjekts visualisiert.

Der Konzeption in Abschnitt 6.7 folgend, existieren zwei Betriebsmodi. Diese unterscheiden sich in der Art der Registrierung bei einer wiederholt durchgeführten, gleichbleibenden Prüfaufgabe. Der *Genauigkeitsmodus* durchläuft den Prüfzyklus in jeder Iteration mittels inkrementeller Registrierung. Im *Zeitmodus* hingegen werden die Aufnahmen lediglich im ersten Prüfzyklus inkrementell registriert. In nachfolgenden Prüfungen dagegen werden die errechneten Ergebnisse des ersten Durchlaufs in Kombination mit der Selbstlokalisierung des Roboters genutzt. Der Zeitmodus besitzt ab dem zweiten Durchlauf eine potentiell geringere Prüfzeit, weil die Anzahl der Aufnahmen verringert wird und die Berechnung der inkrementellen Registrierung entfällt. Der erwartete Messfehler wird dabei neben dem Fehler der inkrementellen Registrierung zusätzlich von der Wiederholgenauigkeit des Roboters negativ beeinflusst. Durch die Fokussierung des Offline-Anwendungsfalls wird für die Validierung der Genauigkeitsmodus genutzt.

## 7.2 Parameterstudie am Prototyp

Die softwareseitige Implementierung der Methode enthält Variablen, deren konkrete Werte mit Hilfe einer Parameterstudie festgelegt wurden. Das zugehörige Vorgehen wird im Folgenden erläutert, dabei werden drei Ziele verfolgt. Erstens wird der Prototyp für das nachfolgende Anwendungsbeispiel parametrierbar, um

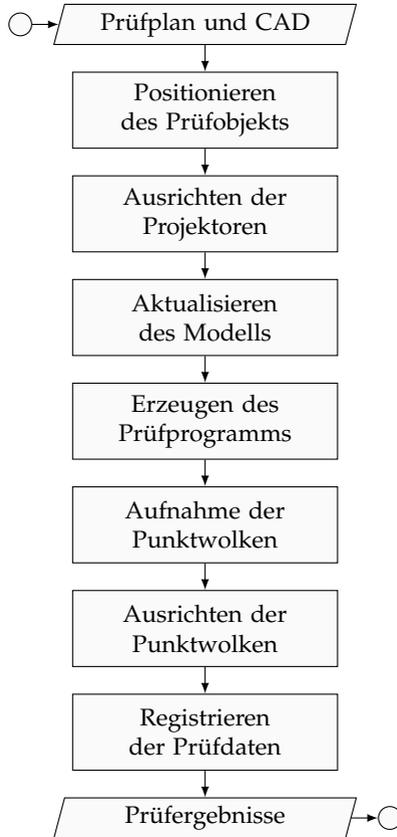


Abbildung 7.5: Prüfablauf am Prototyp.

die Leistungsfähigkeit der Methode bewerten zu können. Zum Zweiten wird der grundsätzliche Einfluss der Variablen auf die Zielgrößen aufgezeigt und drittens kann die Beschreibung der Vorgehensweise genutzt werden, um die Variablen für einen beliebigen Anwendungsfall zweckmäßig zu bestimmen. Die in Tabelle 7.1 aufgezählten Parameter wurden als relevant identifiziert und daher untersucht. Die Zielgrößen nach KLEPPMANN (2013, S. 22) sind eine möglichst geringe Messunsicherheit des Gesamtsystems sowie eine kurze Zeitdauer der Prüfung.

Für sämtliche experimentellen Untersuchungen zur Bestimmung der Parameter

	Parameter	Beschreibung
Ausrichtung	$k_{gauss}$	Größe der Filtermaske bei der Glättung von Punktwolken
	$k_{thresh}$	Faktor für den Schwellwert bei der Kantenerkennung
	$n_{add}$	Anzahl der Pixel, die in Normalenrichtung generiert werden
	$\tau_{ICP}$	Abbruchkriterium: Differenz des Fehlermaßes zwischen zwei Iterationen
Bahnplanung	$\theta = f(\kappa)$	Gewichtung des Registrierungsbaumes abhängig vom Überlagerungsbereich $\kappa$
	$r$	Rastergröße bei der Diskretisierung
	$n_{pose}$	Anzahl Versuche zur Generierung einer gültigen Pose
	$n_{neighbors}$	Anzahl betrachteter Nachbarn beim Aufbau der Roadmap

*Tabelle 7.1: Relevante Parameter der prototypischen Implementierung*

gilt ein Stichprobenumfang  $\geq 50$  und es wird von einer Normalverteilung der Messdaten ausgegangen. Datenpunkte in Diagrammen stellen den Mittelwert der erfassten Daten dar, Fehlerbalken symbolisieren die Standardabweichung. Messunsicherheiten von weniger als 2 % sind nicht visualisiert.

## 7.2.1 Variablen des Algorithmus zur Verknüpfung

Zunächst wurden Parameter des nach Abschnitt 5.4 implementierten Algorithmus zur Verknüpfung analysiert. Grundlage der Versuche bildeten zwei sich um 50 % überlagernde Punktwolken einer ebenen Fläche. Die Transformation des 3D-Sensors  $T_{ref}$  wurde simuliert, indem die zugehörige Punktwolke entsprechend des angesetzten Überlagerungsbereichs beschnitten und synthetisch erweitert wurde. Da einzig der Überlagerungsbereich für die Ausrichtung relevant ist, beeinflussen synthetisch hinzugefügte Bereiche der Punktwolken das Ergebnis der Berechnungen nicht. Durch dieses Vorgehen ist  $T_{ref}$  exakt bekannt. Der Startwert des Algorithmus ist die durch den Industrieroboter geschätzte

## 7 Validierung und Evaluation

---

Anfangstransformation  $T_{robot}$ . Für die Validierung setzt sich diese aus dem Referenzwert  $T_{ref}$  und einem zufälligen Fehler  $\varepsilon_{robot}$  zusammen, welcher in der Anwendung durch den Industrieroboter induziert würde (Gleichung 7.1). Dieser liegt im Bereich von  $\varepsilon_{robot} \leq 1$  mm (ULRICH et al. 2014). Orientierungsfehler wurden in der Parameterstudie nicht betrachtet.

$$T_{robot} = T_{ref} + \begin{bmatrix} 1 & 0 & 0 & \varepsilon_x \\ 0 & 1 & 0 & \varepsilon_y \\ 0 & 0 & 1 & \varepsilon_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{mit } \sqrt{\varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z^2} = \varepsilon_{robot} \quad (7.1)$$

Der Fehler  $\varepsilon_{robot}$  wurde in jeder Instanz variiert. Für jeden Parametersatz wurde die identische Sequenz von fehlerbehafteten Transformationen verwendet, um die Vergleichbarkeit der Ergebnisse zu gewährleisten. Die Zielgröße für die Parametrierung des Algorithmus war zuvorderst die Messunsicherheit  $\varepsilon_{ref}$  nach Gleichung 7.2 bei der Ausrichtung zweier Punktwolken. Die durch 3D-Image-Stitching berechnete Translation  $t_{ICP}$  wurde hierfür mit der tatsächlich vorliegenden Translation  $t_{ref}$  verglichen.

$$\varepsilon_{ref} = \|t_{ref} - t_{ICP}\| \quad (7.2)$$

Im ersten Schritt wurden die Variablen analysiert, welche die Anreicherung der Punktwolken nach Abschnitt 5.4.2 beeinflussen. Die Größe der Filtermaske  $k_{gauss}$  zur Glättung der Eingangsdaten stellt dabei den ersten Parameter dar. Nach ROSIN & ELLIS (1995) werden die Schwellwerte des Kantendetektors aus dem Median der Helligkeitswerte in Kombination mit einem Faktor  $k_{thresh}$  berechnet. Dieser bildet den zweiten zu variierenden Parameter. Zuletzt ist bei der Generierung von räumlichen Strukturen in Normalenrichtung die Höhe dieser Strukturen ausschlaggebend für deren Gewichtung in der späteren Ausrichtung. Die Punkte werden dabei im Raster der Punktwolke erzeugt, der zu variierende Parameter ist daher die Anzahl der zu erzeugenden Punkte  $n_{add}$ . Um Einflüsse der Variablen untereinander einzubeziehen, wurden diese mit Hilfe eines vollfaktoriellen Versuchsplans analysiert (KLEPPMANN 2013, S. 101). Dafür wurden

in Vorversuchen die in Anhang A.1 dargestellten Faktorstufen festgelegt und jede Kombination einem Parametersatz zugewiesen. Abbildung 7.6 zeigt den resultierenden Fehler  $\varepsilon_{ref}$  für sämtliche Parametersätze. Parametersatz 19 mit ( $k_{gauss} = 5 \times 5 \mid k_{thresh} = 0,3 \mid n_{add} = 80$ ) weist dabei die geringste Abweichung  $\varepsilon_{ref} = 8,86 \mu\text{m}$  auf, er wurde daher für nachfolgende Versuche genutzt. Die Standardmessunsicherheit für diesen Parametersatz beträgt  $u_{ref} = 3,91 \mu\text{m}$ .

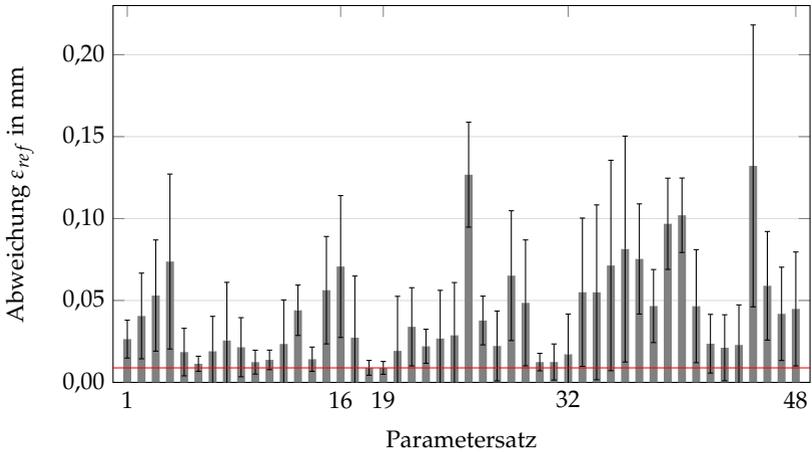


Abbildung 7.6: Verbleibende Abweichung  $\varepsilon_{ref}$  abhängig vom verwendeten Parametersatz. Die Fehlerbalken visualisieren die Standardabweichung. Die Faktorstufen der Parametersätze finden sich in Anhang A.1.

Im zweiten Schritt wurde das Abbruchkriterium des ICP-Algorithmus festgelegt. Das Ziel war hierbei die Minimierung der Laufzeit unter Beibehaltung der potentiell erreichbaren Genauigkeit  $\varepsilon_{ref} = (8,86 \pm 3,91) \mu\text{m}$ . Der Algorithmus soll dafür beendet werden, sobald die Transformation mit maximal möglicher Genauigkeit berechnet wurde. Abbildung 7.7 stellt die Verläufe der Abweichung  $\varepsilon_{ref}$  und des Fehlermaßes  $\varepsilon_{ICP}$  in Bezug zur jeweiligen Iteration des Algorithmus dar. Das Fehlermaß korreliert mit der Abweichung und kann daher zur qualitativen Bewertung der Ausrichtung genutzt werden. Als Abbruchkriterium wurde BESL & MCKAY (1992) folgend das Konvergenzkriterium  $\tau_{ICP}$  festgelegt, welches die Verkleinerung von  $\varepsilon_{ICP}$  in zwei aufeinanderfolgenden Iterationen beschreibt. Abbildung 7.8 zeigt das Abbruchkriterium in Abhängigkeit zur Verkleinerung der tatsächlichen Abweichung  $\tau_{ref}$ . Der Algorithmus

kann beendet werden, sobald die Verbesserung  $\tau_{ref}$  sich im Bereich der Standardmessunsicherheit  $u_{ref}$  für den festgelegten Parametersatz befindet. Als Abbruchkriterium wurde daher  $\tau_{ICP} = 1 \cdot 10^{-5}$  mm gewählt.

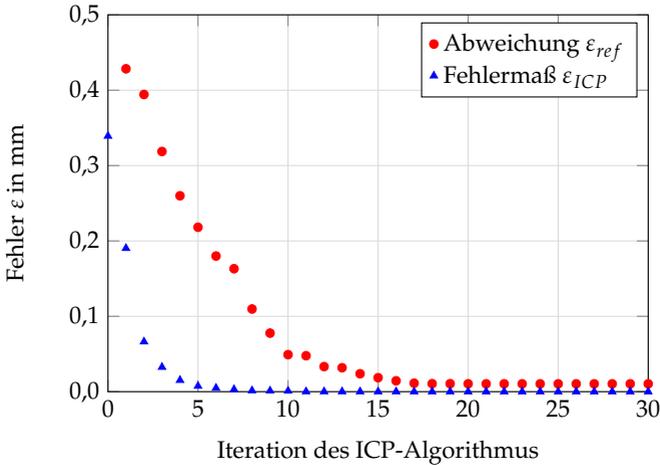


Abbildung 7.7: Das berechnete Fehlermaß  $\epsilon_{ICP}$  folgt der gemessenen Abweichung  $\epsilon_{ref}$  proportional. Es kann daher genutzt werden, um das Ergebnis der Ausrichtung qualitativ zu bewerten.

### 7.2.2 Einfluss des Überlagerungsbereichs

Der Analyse zum Einfluss des Überlagerungsbereichs  $\kappa$  auf die Ausrichtung in Abschnitt 5.5 folgend, wurde dessen Größe in Bezug zur Restabweichung  $\epsilon_{ref}$  untersucht. Die zu verknüpfenden Punktwolken wurden dafür schrittweise zueinander verschoben, sodass sich die Größe von  $\kappa$  jeweils um 10% des Sichtbereichs änderte. Abbildung 7.9 zeigt die resultierende Abweichung  $\epsilon_{ref}$  in Abhängigkeit vom Überlagerungsbereich. Zur besseren Anschaulichkeit wurde  $\kappa$  in Relation zum Sichtbereich des 3D-Sensors  $A_{real}$  gesetzt. Die getroffene Annahme, dass ein größerer Überlagerungsbereich zu einem kleineren Fehler der Ausrichtung führt, wird durch die Messergebnisse bestätigt.

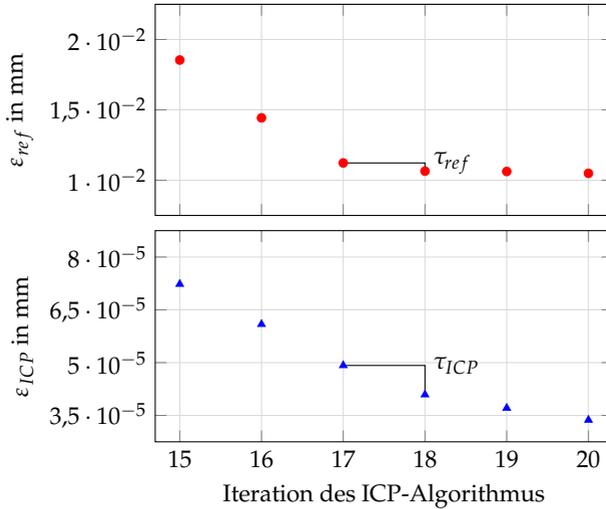


Abbildung 7.8: Zur Festlegung des Abbruchkriteriums  $\tau_{ICP}$  wird dieses in Bezug zur Verbesserung der Abweichung  $\tau_{ref}$  gesetzt. Die erwartete Streuung bei der Ausrichtung dient dabei als Grenzwert.

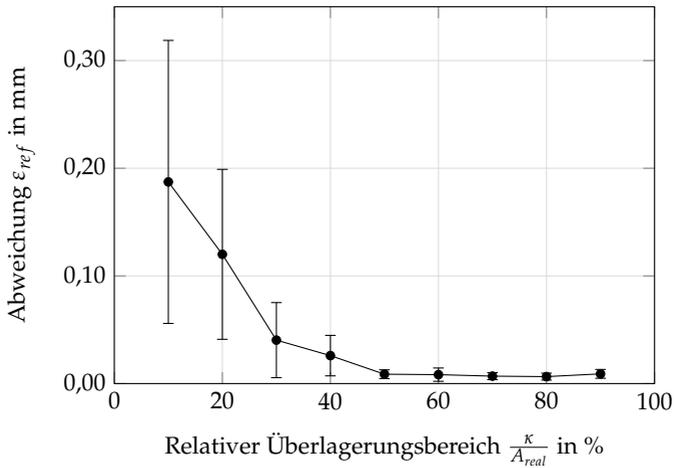


Abbildung 7.9: Abhängigkeit der verbleibenden Abweichung  $\epsilon_{ref}$  von der relativen Größe des Überlagerungsbereichs  $\frac{\kappa}{A_{real}}$ .

Für die Bahnplanung ist der gesamte Verlauf aus Abbildung 7.9 relevant. Die Kanten beim Aufbau des Registrierungsbaums werden mit diesem quantitativen Zusammenhang nach Gleichung 6.4 in Abschnitt 6.5 gewichtet. Die Größe des Überlagerungsbereichs  $\kappa$  wird dabei mit Hilfe der diskretisierten Oberfläche des Prüfobjekts in der Simulation abgeschätzt. Am realisierten Prototyp konnten für  $\kappa < 1000 \text{ mm}^2$  ( $\approx 20\%$  des Sichtbereichs) Ausrichtungen nicht zuverlässig errechnet werden. Derartige Konstellationen werden daher zurückgewiesen und es wird keine Kante im Graph erzeugt (Kantengewicht  $\theta \rightarrow \infty$ ).

### 7.2.3 Variablen der automatischen Bahnplanung

Neben dem Algorithmus zur Verknüpfung weist das nach Kapitel 6 implementierte Planungsverfahren weitere Variablen auf, die für eine erfolgreiche Anwendung der Methode parametrisiert werden müssen. Für die zugehörigen Experimente wird eine ebene Platte als Prüfobjekt verwendet, die relevanten Parameter sind der Rasterabstand  $r$ , die Anzahl der Versuche zur Generierung einer gültigen Aufnahmepose  $n_{pose}$  und die Anzahl der Nachbarn beim Verbinden der Roadmap  $n_{neighbors}$ . Diese sind voneinander unabhängig und werden daher isoliert betrachtet.

Um den während der Messung aufgenommenen Sichtbereich des 3D-Sensors abzuschätzen, wird dieser aus dem diskretisierten CAD-Modell des Prüfobjekts bestimmt (Abschnitt 6.3). Der Rasterabstand  $r$  beeinflusst dabei die Präzision der Berechnung und die Zeitdauer der Diskretisierung  $t_{c,discr}$ . Abbildung 7.10 zeigt die relative Abweichung zwischen geschätztem und realem Sichtbereich  $\frac{\Delta A}{A_{real}}$  abhängig vom Rasterabstand  $r$ . Hierfür wurden jeweils 1000 Aufnahmeposen generiert und der zugehörige Sichtbereich abgeschätzt. Die Laufzeit  $t_{c,discr}$  steigt mit sinkendem Rasterabstand überproportional an. Für den vorliegenden Prototyp wird  $r = 5 \text{ mm}$  als Kompromiss aus Präzision der Flächenberechnung und Laufzeit gewählt. Tendenziell wird der berechnete Sichtbereich unterschätzt und bildet somit eine Sicherheit gegenüber lokalen Störungen bei der realen Aufnahme.

Bei der Generierung von Aufnahmeposen müssen zahlreiche Bedingungen erfüllt werden (Abschnitt 6.4). Dabei kann partiell der Fall eintreten, dass das Auffinden einer gültigen Pose unmöglich ist. Es existiert daher eine Variable

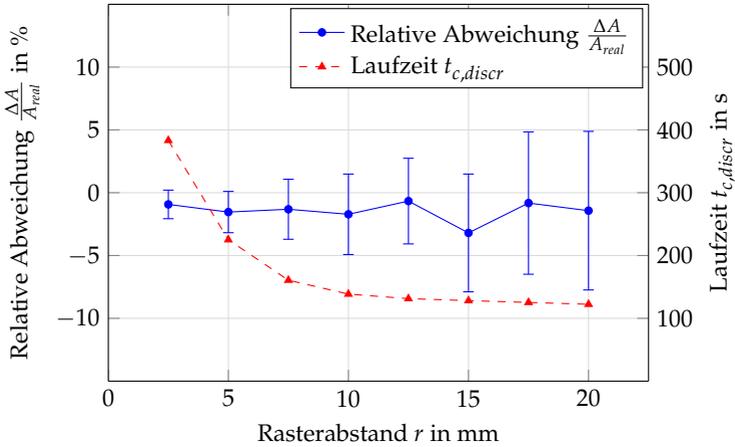


Abbildung 7.10: Abweichung des berechneten Flächeninhalts  $\frac{\Delta A}{A_{real}}$  und zugehörige Laufzeit  $t_{c,discr}$  in Abhängigkeit vom Rasterabstand  $r$ .

zur Begrenzung der maximalen Anzahl an Versuchen zur Generierung einer Aufnahmepose  $n_{pose}$ . Dieser Wert muss so parametrisiert werden, dass eine vorhandene Lösung mit hoher Wahrscheinlichkeit gefunden und gleichzeitig die Zeitdauer zur Berechnung  $t_{c,pose}$  in Bereichen ohne Lösung minimiert wird. Als Quantifizierung dieser Anforderung wurde die Erfolgsquote  $p_{suc}$  gewählt. Sie gibt an, mit welcher Wahrscheinlichkeit an einer beliebigen Stelle im Arbeitsraum eine gültige Aufnahmepose gefunden wird. Abbildung 7.11 stellt für den vorliegenden Prototyp die Abhängigkeit der Erfolgsquote von  $n_{pose}$  dar, wobei sich  $p_{suc}$  bei etwa 61 % sättigt. Die Anzahl der Versuche für eine gültige Pose wird daher auf  $n_{pose} = 60$  festgelegt, um die Laufzeit  $t_{c,pose}$  zu begrenzen und gleichzeitig die maximale Erfolgsquote sicherzustellen.

Für die Erzeugung der Roboterbahn wird ein lokaler Planer  $\Lambda$  genutzt (Abschnitt 6.6.2). Dieser prüft die Verbindung zwischen benachbarten Posen, um einen zusammenhängenden Graphen zu generieren. Je mehr Nachbarn  $n_{neighbors}$  dabei pro Pose betrachtet werden, desto besser wird der freie Bewegungsraum des Roboters erschlossen und es können potentiell kürzere Fahrzeiten  $t_{robot}$  erreicht werden. Gleichzeitig steigt jedoch die Zeitdauer zur Berechnung der Bahn  $t_{c,path}$ , da für jede Verbindung im Graph die zugehörige

## 7 Validierung und Evaluation

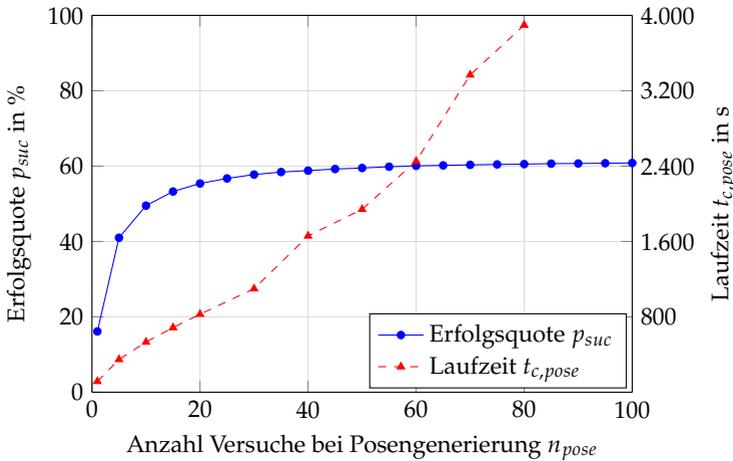


Abbildung 7.11: Erfolgsquote für die Suche nach einer gültigen Pose  $p_{suc}$  und zugehörige Laufzeit  $t_{c,pose}$  in Abhängigkeit von der maximalen Anzahl an Versuchen  $n_{pose}$ .

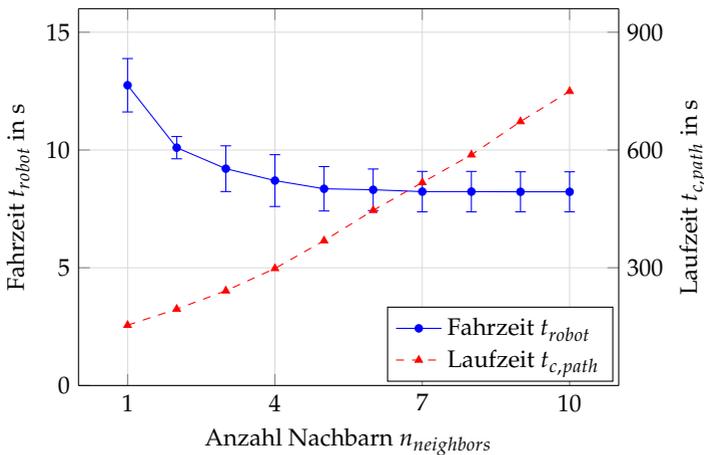


Abbildung 7.12: Fahrzeit des Roboters  $t_{robot}$  in Abhängigkeit von der Anzahl an Nachbarn  $n_{neighbors}$  und in Bezug zur Laufzeit der Berechnung  $t_{c,path}$ .

---

PTP-Bewegung auf Kollisionen geprüft werden muss. Abbildung 7.12 stellt den quantitativen Zusammenhang von Fahrzeit, Laufzeit und Anzahl der Nachbarn dar. Um die Zeitdauer der Berechnung zu begrenzen, wird  $n_{neighbors} = 5$  gewählt, da für größere Werte die Fahrzeit  $t_{robot}$  keine relevante Verbesserung erfährt.

### 7.3 Experiment am Anwendungsbeispiel

Aufbauend auf der prototypischen Umsetzung (Abschnitt 7.1) und deren Parametrierung (Abschnitt 7.2) wurde die vorgestellte Methode anhand eines Anwendungsbeispiels untersucht.

#### 7.3.1 Versuchsaufbau und Durchführung

Als Prüfobjekt diente ein industriell hergestelltes Bauteil der Automobilindustrie mit zusätzlich angebrachten Prüfkugeln (Abbildung 7.13). Die Kugeln besitzen einen Radius von 10 mm, sind temperaturstabil und im Vergleich zu anderen Geometrieelementen sehr robust erkennbar. Ein Einfluss der Auswertesoftware für die Detektion der Kugelgeometrien auf die Messunsicherheit des Gesamtsystems ist für dieses Anwendungsbeispiel vernachlässigbar gering.

Zur Bewertung der Prüfmittelfähigkeit des roboterbasierten Messsystems wird die Kenngröße Kugelabstandsabweichung nach VDI 2634 genutzt. Die Prüfaufgabe besteht daher aus der Messung des geometrischen Abstands der Prüfkugeln. Zur Berechnung der Mittelpunkte wird die Methode der kleinsten Fehlerquadrate genutzt, um eine Kugelgeometrie mit vorgegebenem Radius in die entsprechende Punktwolke einzupassen und anschließend deren Mittelpunkt zu bestimmen. Die in Abbildung 7.13 aufgezeigten Kugelabstände  $\{\mathcal{M}_1, \mathcal{M}_2\}$  werden mit Hilfe dieser Mittelpunkte berechnet und mit Referenzwerten verglichen. Das Bauteil selbst dient in diesem Szenario dem Algorithmus als Träger der zur Verknüpfung notwendigen Aufnahmen.

Die Referenzmessung wurde an einem Koordinatenmessgerät durchgeführt. Die Messunsicherheit betrug dabei  $u_{CMM} = 3,5 \mu\text{m}$ , die Prüfzeit inklusive Arbeitsvorbereitung 1,5 h. Als zweites Vergleichssystem dient der unmodifizierte

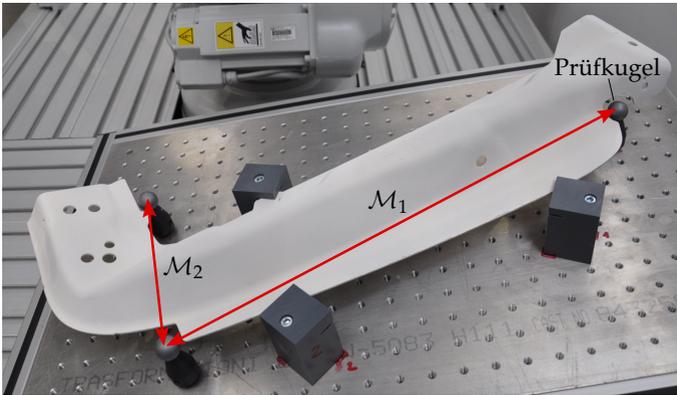


Abbildung 7.13: Prüfobjekt aus Bauteil und Prüfkugeln. Die Prüfaufgabe besteht aus der Bestimmung der geometrischen Abstände  $\{M_1, M_2\}$ .

Industrieroboter, wobei die Registrierung der Aufnahmen direkt aus den Steuerungsdaten abgeleitet wurde. Das Experiment umfasste 30 Prüfzyklen, wobei in jedem Versuch ein vollständiger Prüfzyklus aus Erstellung des Prüfprogramms, Aufnahme der Daten sowie Berechnung und Speicherung der Messergebnisse durchlaufen wurde. Als Bewertungskriterien wurden die Messunsicherheit der Längenmessung im Vergleich zum unmodifizierten Roboter und zur Referenzmessung am Koordinatenmessgerät sowie die zugehörige Prüfzeit gewählt.

### 7.3.2 Auswertung der Messdaten

Die manuellen Arbeitsschritte (Aufstellung der Projektoren, Anpassung des Modells) benötigten etwa 15 min. Die im Anschluss durchgeführte automatische Bahnplanung wurde in jedem Durchgang erfolgreich abgeschlossen und erzeugte ein valides Prüfprogramm. Abbildung 7.14 zeigt beispielhaft einen gefundenen Registrierungspfad. Die Laufzeiten der Programmerstellung sind in Tabelle 7.2 zusammengefasst. Die Planung der Aufnahmeasen und das Prüfen konkreter Roboterbewegungen nahmen erwartungsgemäß den größten Teil der Laufzeit in Anspruch. Der Grund hierfür ist die zeitaufwändige und gleichzeitig sehr oft genutzte Kollisionsprüfung (Abschnitt 4.2.2).

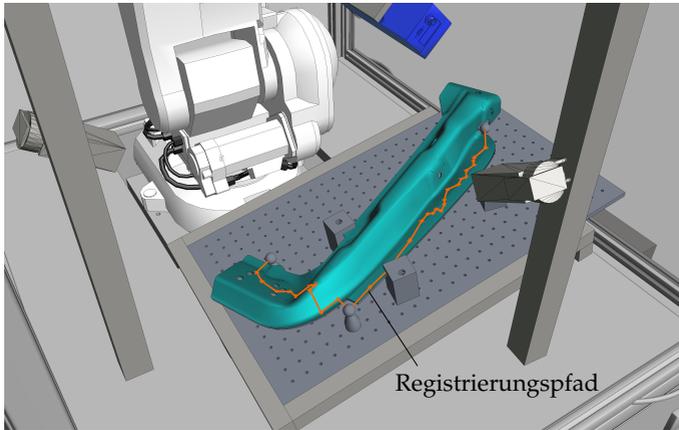


Abbildung 7.14: Dargestellt ist beispielhaft ein berechneter Registrierungspfad. In jedem Versuchszyklus wurde ein valides Prüfprogramm gefunden.

Mit Blick auf die Durchführung der vollständigen Prüfaufgabe ergaben sich die Prozesszeiten in Tabelle 7.3. 3D-Image-Stitching benötigt grundsätzlich eine längere Zeitdauer für die Durchführung einer Messaufgabe als ein unmodifiziertes Robotersystem. Die Gründe hierfür sind im Wesentlichen die Aufnahme von dem Konzept inhärenten Zwischenaufnahmen und die zugehörigen Berechnungen zur Verknüpfung. Die parallele Berechnung der inkrementellen Registrierung bereits während der Aufnahme der Punktwolken verringert dabei die Zeitdifferenz für den vorliegenden Prototyp. Tabelle 7.4 zeigt die

	Laufzeit in s	Anteil in %
Vollständiger Planungszyklus	1161,68 ± 47,90	100,0
Diskretisierung Prüfobjekt	38,40 ± 0,48	3,3
Generierung Aufnahmeposen	616,56 ± 19,82	51,1
Erstellung Registrierungsbaum	160,49 ± 8,92	13,8
Berechnung Roboterbahn	346,23 ± 41,15	29,8

Tabelle 7.2: Berechnungszeit zur automatischen Erstellung des Prüfprogramms. Der Planungsablauf gliedert sich in die vier dargestellten Teilschritte.

## 7 Validierung und Evaluation

Berechnungszeit der Verknüpfung von zwei Punktwolken, den größten Teil verursachte dabei der ICP-Algorithmus. Dieser benötigte  $(20 \pm 6)$  Iterationen, bis das Abbruchkriterium  $\tau_{ICP}$  erreicht wurde.

	Messsystem mit 3D-Image-Stitching	Unmodifiziertes Robotersystem
Prüfzeit in s	$258,20 \pm 15,36$	$37,19 \pm 0,53$
Bewegung des Roboters in s	$30,43 \pm 0,79$	$28,54 \pm 0,52$
Aufnahme der Daten in s	$54,91 \pm 0,07$	$8,65 \pm 0,01$
Registrierung in s	$250,47 \pm 14,58$	0
Messung der Merkmale in s	$0,45 \pm 0,02$	$0,45 \pm 0,02$

Table 7.3: Prüfzeit für die vorgegebene Aufgabe. Die Registrierung startet parallel zur Aufnahme der Punktwolken, die Prüfzeit ergibt sich daher nicht aus der Summe der Teilschritte.

	Laufzeit in s	Anteil in %
Verknüpfung zweier Punktwolken	$13,88 \pm 3,35$	100,0
Einlesen Punktwolken	$1,84 \pm 0,06$	13,2
Zuschnitt	$0,43 \pm 0,02$	3,1
Anreicherung	$0,59 \pm 0,03$	4,2
Ausrichtung mittels ICP	$11,02 \pm 3,34$	79,4

Table 7.4: Berechnungszeit zur Ausrichtung von zwei Punktwolken. Der Ablauf gliedert sich in die vier dargestellten Teilschritte.

Die Ergebnisse der Messungen sind in Tabelle 7.5 dargestellt. Neben den absoluten Abständen der Kugeln  $\{\mathcal{M}_1, \mathcal{M}_2\}$  ist jeweils die Abweichung zur Referenzmessung  $\{\mathcal{M}_{1,ref}, \mathcal{M}_{2,ref}\}$  mit dem Koordinatenmessgerät aufgezeigt. Der absolute und der relative Messfehler  $\varepsilon_{abs}$  respektive  $\varepsilon_{rel}$  berechnen sich dabei nach Gleichung 7.3 und Gleichung 7.4. Bei Nutzung von 3D-Image-Stitching zeigt sich für das Anwendungsbeispiel eine signifikante Steigerung der Genauigkeit gegenüber dem unmodifizierten Robotersystem.

$$\varepsilon_{abs} = \left| \mathcal{M} - \mathcal{M}_{ref} \right| \quad (7.3)$$

$$\varepsilon_{rel} = \left| \frac{\mathcal{M} - \mathcal{M}_{ref}}{\mathcal{M}_{ref}} \right| \quad (7.4)$$

	Messsystem mit 3D-Image-Stitching	Unmodifiziertes Robotersystem
Prüfmerkmal $\mathcal{M}_1$ in mm	475,35 ± 0,03	476,37 ± 0,05
Absoluter Fehler $\varepsilon_{1,abs}$ in mm	0,08 ± 0,03	0,94 ± 0,05
Relativer Fehler $\varepsilon_{1,rel}$ in %	0,017 ± 0,007	0,197 ± 0,010
Prüfmerkmal $\mathcal{M}_2$ in mm	181,98 ± 0,02	182,59 ± 0,03
Absoluter Fehler $\varepsilon_{2,abs}$ in mm	0,04 ± 0,02	0,65 ± 0,03
Relativer Fehler $\varepsilon_{2,rel}$ in %	0,020 ± 0,013	0,356 ± 0,016

*Tabelle 7.5: Messergebnisse für die Prüfmerkmale  $\mathcal{M}_1$  und  $\mathcal{M}_2$  und deren Abweichungen zur Referenzmessung mit dem Koordinatenmessgerät. Dargestellt sind jeweils der Mittelwert und die Standardabweichung sämtlicher Durchläufe.*

### 7.3.3 Bewertung nach technischen Aspekten

Durch den Industrieroboter induzierte Abweichungen werden mit Hilfe von 3D-Image-Stitching im Anwendungsbeispiel robust ausgeglichen. Im Vergleich zur Registrierung mittels unmodifiziertem Robotersystem kann am Prototyp eine Steigerung der Genauigkeit attestiert werden. Die vorliegende Implementierung setzt dabei eine vergleichsweise kleine Kinematik mit relativ hoher Pose-Genauigkeit ein (Tabelle 7.5). Für größere Ausführungen mit entsprechend mehr Bewegungsfreiheit und damit höherer Flexibilität der Anlage in Bezug auf die Bauteilgröße ist von höheren Abweichungen auszugehen. Die Genauigkeit eines Koordinatenmessgeräts konnte indessen nicht erreicht werden.

Bei Betrachtung der Prüfzeit wird die Annahme bestätigt, dass 3D-Image-Stitching im Vergleich zu einer alleinigen Registrierung durch den Industrieroboter eine längere Prozesszeit aufweist. Die Laufzeit des ICP-Algorithmus

## 7 Validierung und Evaluation

---

erwies sich dabei als maßgeblicher Faktor und eine Verkürzung sollte bei einer Weiterentwicklung des Prototyps fokussiert werden.

Im Rahmen der Arbeitsvorbereitung muss das Modell angepasst werden. Die Komponenten der realen und virtuellen Messzelle sollten dabei soweit möglich deckungsgleich positioniert werden, was dem Anwender nur bis zu einem gewissen Grad möglich ist. Da die konstruierten externen Projektoren einen breiten Abstrahlwinkel besitzen und Aufnahmen relativ zum Prüfobjekt registriert werden, ist eine Abweichung im Millimeterbereich jedoch unkritisch und der Prozess damit als robust bei der Inbetriebnahme einzuschätzen. Zur visuellen Orientierung wäre dennoch eine für den Anwender sichtbare Abgrenzung des Sichtbereichs hilfreich, um die korrekte Ausrichtung zu vereinfachen. Eine sensorische Detektion von Position und Orientierung des Projektors würde als Weiterentwicklung eine automatisierte Aktualisierung des Modells ermöglichen. In einem weiteren Schritt können eine softwaregestützte Bewertung und Korrektur der Ausrichtung erfolgen.

Für die Registrierung mussten keine Passmarken auf dem Bauteil befestigt werden, was gegenüber photogrammetrischen Ansätzen als Vorteil zu werten ist. Das Konzept nutzt weiterhin externe Projektoren, die im Vergleich zu einem externen Tracking-System einfach in der Handhabung und kostengünstig sind. Wie bei einem Lasertracker muss jedoch die Sichtlinie zwischen Projektor und Prüfobjekt erhalten bleiben, wodurch die Flexibilität einschränkt wird. Weiterhin steigern diese zusätzlichen Komponenten den Aufwand zur Inbetriebnahme und sind damit im Betrieb komplexer als ein Messsystem mit kalibriertem Roboter. 3D-Image-Stitching zeichnet sich gegenüber einer Kalibrierung durch die Unabhängigkeit vom eingesetzten Roboter und damit von Hersteller, Alterung des mechanischen Aufbaus und Temperatureinflüssen aus. Eine spezifische Modellbildung abhängig vom eingesetzten Robotertyp entfällt.

Gegenüber dem aktuellen Stand in der Industrie ist die weitgehende Automatisierung der Prüfdatenerfassung von der Erstellung einer Roboterbahn bis zur Registrierung der Aufnahmen als vorteilhaft zu bewerten. Die Prozesszeit wird wesentlich verkürzt und der bedienende Prüftechniker entlastet. Im Falle von 3D-Image-Stitching gilt dies besonders, da die Planung des Prüfablaufs eine komplexe Ausrichtung des 3D-Sensors erfordert, sodass eine geeignete Sequenz

---

an sich überlagernden Punktwolken entsteht. Für den Anwender ist die manuelle Generierung der optimalen Bahn durch die zahlreichen Anforderungen und Randbedingungen (Abschnitt 6.1) kaum möglich.

In der Fertigungsmesstechnik steht damit neben dem in Relation hochpräzisen aber langsamen Koordinatenmessgerät und dem sehr schnellen aber unpräzisen Roboter mit dem durch 3D-Image-Stitching verbesserten Robotersystem eine alternative Technologie zur Verfügung. Jede dieser Technologien bietet dabei spezifische Vor- und Nachteile, sodass die Wahl des Messsystems von der Anforderungslage des konkreten Einsatzgebietes abhängen sollte.

#### **7.4 Fallstudie zur Bewertung der Wirtschaftlichkeit**

Die Zielstellung von 3D-Image-Stitching ist die Befähigung roboterbasierter Messtechnik für die Offline-Prüfung, womit diese eine Alternative zu konventionellen taktilen Koordinatenmessgeräten wird. Ein Vergleich beider Messsysteme mit Blick auf deren Wirtschaftlichkeit ist daher zweckmäßig, um die vorgestellte Methode zu bewerten. Unterschiede ergeben sich bei den Investitionskosten, der Vorbereitungs- und Messzeit sowie dem Aufwand zur Programmierung des Prüfablaufs. Die technische Rahmenbedingung einer ausreichenden Messgenauigkeit beider Systeme wird als gegeben angenommen.

Als Vergleichskriterium werden die Kosten für einen Prüfauftrag herangezogen. Diese sind abhängig von der Zeit zur Durchführung der Prüfaufgabe und den Kosten für die Belegung der Messmaschine. Dafür wird zunächst der Maschinenstundensatz nach HORSCH (2015, S. 145) ermittelt, zusätzlich werden die Personalkosten berücksichtigt. Der Umfang einer Prüfaufgabe und damit die notwendige Zeit zur Durchführung spiegelt sich wiederum in der Anzahl zu prüfender Merkmale wieder. Ausgehend von durchschnittlich gleicher Planungs- und Messzeit pro Prüfmerkmal kann somit von der Anzahl notwendiger Aufnahmen auf die Belegungszeit geschlossen werden. Wird stets eine Auslastung von 100 % angenommen, wirkt sich eine höhere Messgeschwindigkeit positiv auf die Kosten aus.

Das beispielhafte Szenario besteht aus einem Prüfobjekt mit einer definierten Anzahl geometrischer Merkmale. Die Fixkosten der für die Prüfung zur Ver-

## 7 Validierung und Evaluation

fügung stehenden Messmaschinen werden in Tabelle 7.6 charakterisiert und orientieren sich an marktüblichen Werten. Es ergibt sich durch die geringeren Investitionskosten ein wesentlich günstigerer Maschinenstundensatz für das roboterbasierte Messsystem mit 3D-Image-Stitching im Vergleich zum Koordinatenmessgerät.

	Messsystem mit 3D-Image-Stitching	Koordinaten- messgerät
Anschaffungskosten	126.000 €	396.000 €
Roboter	50.000 €	
3D-Sensor	20.000 €	
Projektoren	5.000 €	
Koordinatenmessgerät		300.000 €
Peripherie	15.000 €	15.000 €
Auswertesoftware	15.000 €	15.000 €
Planungskosten (20 %)	21.000 €	66.000 €
Nutzungsdauer	6 Jahre	6 Jahre
Zinssatz	5 %	5 %
Abschreibung pro Jahr	21.000 €	66.000 €
Kalkulatorische Zinsen pro Jahr	3.150 €	9.900 €
Energiekosten pro Jahr	3.000 €	3.000 €
Instandhaltungskosten pro Jahr	1.500 €	1.500 €
Raumkosten pro Jahr	1.500 €	1.500 €
Stundensatz Maschine $K_{M,rate}$	14,96 €	40,63 €
Stundensatz Personal $K_{P,rate}$	90,00 €	90,00 €

Tabelle 7.6: Stundensatz der Messsysteme und des bedienenden Prüftechnikers

In simulierten Versuchen und durch Befragung von erfahrenen Prüftechnikern wurden die in Tabelle 7.7 dargestellten Zeitaufwände ermittelt. In die Zeit für Aufbau und Einrichtung  $t_{pre}$  fallen neben grundlegenden Arbeitsschritten, wie der Positionierung des Prüfbobjekts im Arbeitsraum der Maschine, systemspezifische Tätigkeiten an. In der Summe ergibt sich jedoch ein ähnlicher Arbeitsaufwand für den Anwender. Unabhängig vom Umfang der Prüfaufgabe ist bei Nutzung der vorgestellten Methode eine initiale Analyse des Prüfbobjekts

mit der Dauer  $t_{p,0}$  notwendig. Pro Prüfmerkmal ist hingegen durch die automatisierte Planung ein Zeitvorteil gegenüber der manuellen Programmierung des Koordinatenmessgeräts zu erwarten. Für die manuelle Programmierzeit pro Merkmal  $t_{p,man}$  fallen zusätzlich Personalkosten an, da die Arbeitsschritte durch einen Prüftechniker durchgeführt werden. Im Gegensatz dazu ist die Berechnungszeit pro Merkmal  $t_{p,auto}$  unabhängig vom Eingreifen des Anwenders.

	Messsystem mit 3D-Image-Stitching	Koordinaten- messgerät
Aufbau und Einrichtung $t_{pre}$	60 min	60 min
Aufspannen Prüfobjekt	20 min	20 min
Kalibrierung		25 min
Ausrichten Projektoren	15 min	
Anpassung Modell	25 min	15 min
Initiale CAD-Analyse $t_{p,0}$	60 min	0 min
Programmierzeit pro M. $t_{p,man}$	0 min	5 min
Berechnungszeit pro M. $t_{p,auto}$	15 s	0 s
Messzeit pro Merkmal $t_m$	15 s	30 s

Tabelle 7.7: Zeitaufwände pro Prüfauftrag

Aus den Stundensätzen der Messmaschine  $K_{M,rate}$  und des bedienenden Prüftechnikers  $K_{P,rate}$ , den Zeitaufwänden in Tabelle 7.7 sowie der Anzahl zu messender Prüfmerkmale  $n$  lassen sich die Kosten für die Belegung der Maschine  $K_M$  nach Gleichung 7.5 respektive die Personalkosten  $K_P$  nach Gleichung 7.6 berechnen. Die Summe hieraus ergibt die Gesamtkosten für den Prüfauftrag  $K_{task}$  (Gleichung 7.7). Abbildung 7.15 zeigt die Entwicklung von  $K_{task}$  in Abhängigkeit vom Umfang des Prüfauftrags.

$$K_M = (t_{pre} + t_{p,0} + n \cdot t_{p,man} + n \cdot t_{p,auto} + n \cdot t_m) \cdot K_{M,rate} \quad (7.5)$$

$$K_P = (t_{pre} + n \cdot t_{p,man}) \cdot K_{P,rate} \quad (7.6)$$

$$K_{task} = K_M + K_P \quad (7.7)$$

## 7 Validierung und Evaluation

---

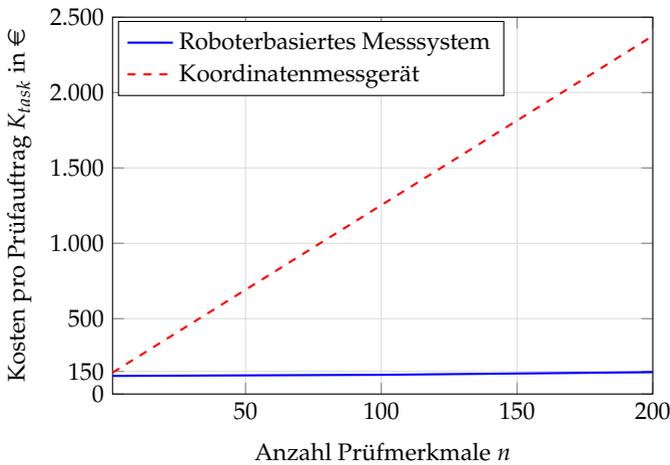


Abbildung 7.15: Kosten eines Prüfauftrags abhängig von dessen Umfang und dem verwendeten Messsystem in der dargelegten Fallstudie.

Diese Fallstudie belegt, dass roboterbasierte Messsysteme auf Basis von 3D-Image-Stitching eine wirtschaftliche Alternative zu konventionellen Koordinatenmessgeräten darstellen. Das monetäre Einsparpotential steigt dabei mit dem Umfang der Prüfaufgabe, wobei die Automatisierung der Programmierfähigkeit ausschlaggebend ist. Die in Abbildung 7.15 dargestellten Kosten steigen für beide Systeme mit der Anzahl zu prüfender Merkmale. Bei Nutzung der vorgestellten Methode 3D-Image-Stitching ist diese Steigung jedoch vernachlässigbar klein und die Kosten sind daher nahezu unabhängig vom Umfang der Prüfaufgabe. Ausgehend von einer Auslastung von 100% wird dabei durch die kürzere Prozesszeit nicht nur günstiger, sondern auch mehr geprüft, wodurch zugehörige Produktionsschritte mit Hilfe der Qualitätssicherung effektiver geregelt werden können.

Diese Beispielrechnung gilt unter der Voraussetzung, dass die Messgenauigkeit des roboterbasierten Messsystems für den betrachteten Anwendungsfall ausreicht. Falls eine höhere Genauigkeit gefordert ist, muss nach wie vor auf die noch immer wesentlich präziseren Koordinatenmessgeräte als Prüfmittel zurückgegriffen werden. Zusätzlich ist zu beachten, dass die konventionelle taktile Koordinatenmesstechnik etabliert und zugehörige Planungs- sowie

---

Prüfabläufe gut untersucht sind. Vor dem Einsatz neuartiger Verfahren wie 3D-Image-Stitching ist daher eine Analyse im Hinblick auf die notwendige Aus- und Weiterbildung der Prüftechniker zur Nutzung der Robotertechnik zweckmäßig.



## 8 Zusammenfassung und Ausblick

Roboterbasierte Messsysteme werden in der Fertigungsmesstechnik als flexible Prüfmittel eingesetzt. Dabei bleiben Anwendungsfälle der Offline-Prüfung bisher außen vor, da die hierfür notwendige Messgenauigkeit nicht erreicht wird. Stattdessen werden kostenintensive Koordinatenmessgeräte verwendet, die im Vergleich eine wesentlich höhere Prüfzeit aufweisen. Die vorgestellte Methode zielt daher auf die Befähigung roboterbasierter Messsysteme für den Einsatz in der Offline-Messtechnik, um flexibler und kostengünstiger prüfen zu können. Bei der Betrachtung des Messsystems, bestehend aus Industrieroboter und 3D-Sensor, fällt die Diskrepanz zwischen beiden Komponenten hinsichtlich der Präzision auf. Während der Sensor im Mikrometer-Bereich arbeitet, erreicht der Roboter lediglich Pose-Genauigkeiten im Millimeter-Bereich. Der Ansatz zur Steigerung der Genauigkeit besteht daher im Entfernen des Roboters als handhabende Kinematik aus der Messkette, damit dieser die Messunsicherheit nicht negativ beeinflusst. Die als 3D-Image-Stitching benannte Methode ist im Rahmen der vorliegenden Arbeit in den Gesamtprozess des Prüfens von Karosserieteilen eingebettet.

Der Lösungsansatz nutzt den 3D-Sensor, welchen die Anwendung ohnehin erfordert, für eine globale und datengetriebene Registrierung. Dazu werden zwischen den Prüfmerkmalen zusätzliche Aufnahmen erzeugt, sodass sich jeweils zwei benachbarte Punktwolken teilweise überlagern. Anschließend wird dieser gemeinsame Bereich genutzt, um deren räumliche Beziehung zu berechnen. Die fehlerbehafteten Positionsdaten der Robotersteuerung haben dadurch keinen Einfluss auf die Messunsicherheit der Prüfaufgabe. Da im betrachteten Anwendungsfall der Prüfung von Karosserieelementen homogene Oberflächen der Prüfobjekte auftreten, müssen für eine datengetriebene Verknüpfung zusätzliche Informationen eingebracht werden. Es wird daher eine externe Projektion aufgebracht und in räumliche Strukturen umgewandelt. Zur anschließenden Ausrichtung der Punktwolken kommt ein modifizierter ICP-Algorithmus zum

Einsatz. Nachdem die räumliche Verschiebung zwischen je zwei benachbarten Punktwolken berechnet ist, werden sämtliche Aufnahmen in ein globales Koordinatensystem überführt.

Durch die zusätzlich notwendigen Aufnahmen für eine datengetriebene Registrierung würde die zeitaufwändige, manuelle Planung des Prüfprogramms weiter erschwert. Die zugehörigen Arbeitsschritte wurden daher mit Hilfe stochastischer Bahnplanungsmethoden automatisiert. Im ersten Schritt werden für die Oberfläche des Prüfobjekts Aufnahmeposes generiert, welche die Voraussetzungen für eine erfolgreiche Datenaufnahme erfüllen. Im zweiten Schritt wird eine Teilmenge dieser Posen ausgewählt, welche sich gegenseitig überlagern und mit minimalem Messfehler verknüpft werden können. Die Verbindung der Posen zu einer kollisionsfreien Roboterbahn erfolgt im dritten und letzten Schritt. Die Berechnungszeit hängt wesentlich von der Effizienz des eingesetzten Kollisionsdetektors ab, ist jedoch wesentlich kürzer als eine manuelle Planung. Diese Reduzierung des Expertenaufwands für Inbetriebnahme und Programmierung des Systems führt zu einer gesteigerten Wandelbarkeit des Prüfmittels und damit zu einem erweiterten Einsatzfeld im Rahmen der Qualitätssicherung.

Da sich die Registrierung allein aus den aufgenommenen Punktwolken berechnet, beeinflussen Pose- und Wiederholgenauigkeit des Roboters das Messergebnis nicht. Durch die sequenzielle Verknüpfung der Punktwolken pflanzt sich jedoch der Messfehler des 3D-Sensors fort, sodass dieser mit steigender Größe des Prüfobjekts zunimmt. Der entstehende Messfehler wird daher abgeschätzt und durch eine geeignete Auswahl der zu verknüpfenden Aufnahmen minimiert. Der vorgestellte Ansatz fokussiert dabei die Prüfung einzelner geometrischer Merkmale an einem Prüfobjekt. Das Konzept verhindert daher mehrdeutige Lösungen durch geeignete Reduktion der Aufnahmeposes und Beschränkung der Verknüpfung auf jeweils zwei Nachbarn in einem Registrierungsbaum. Sobald die Prüfaufgabe hingegen eine vollständige Erfassung des Prüfobjekts erfordert, müssen für jede Aufnahme sämtliche Nachbarn betrachtet werden. Die entstehenden Punktwolken überlagern sich dabei mehrfach und bilden teilweise geschlossene Kreise. Für diesen Anwendungsfall müsste die Methode erweitert werden, um Punktwolken nicht paarweise, sondern im Gesamtverbund simultan zueinander auszurichten.

---

Die Validierung anhand einer prototypischen Umsetzung zeigt die Funktionsfähigkeit und Machbarkeit der Methode. Eine Realisierung im industriellen Maßstab in Bezug auf die Komplexität und Größe der Prüfobjekte wird hierdurch nicht ersetzt und muss das Ziel weitergehender Betrachtungen sein. Die vorgestellten Ergebnisse veranschaulichen jedoch das Potential von 3D-Image-Stitching, mit dem roboterbasierte Messsysteme flexibler und genauer eingesetzt werden können.

Ein Projektor auf LED-Basis wurde für eine einfache und kostengünstige Methode zur Mustergenerierung gewählt. Für Prüfstände ohne Abschirmung von Umgebungslicht (z. B. bei sehr großen Prüfobjekten) können alternativ diffraktive Optiken mit Laser als Lichtquelle eingesetzt werden. Diese erzeugen potentiell stabilere, intensivere und damit leichter erkennbare Muster. Die Investitionskosten steigen dadurch an, sind jedoch in Relation zum Gesamtsystem noch immer gering. Eine weitere Möglichkeit zur Weiterentwicklung ist die automatische Ausrichtung des Projektors durch einen zusätzlichen Industrieroboter. Die zugehörige Bahnplanung wird hierbei durch die zu koordinierenden Bewegungen beider Kinematiken komplexer, dem Anwender erspart es jedoch die manuelle Ausrichtung der Projektoren und damit einen weiteren manuellen Arbeitsschritt.

Für die Inline-Messtechnik ist 3D-Image-Stitching vorteilhaft, wenn die durch Kalibrierung des Roboters erreichbare Messgenauigkeit nicht ausreicht und externe Tracking-Systeme aufgrund hoher Investitionskosten nicht eingesetzt werden können. Durch die zusätzlichen Aufnahmen ist jedoch mit einer im direkten Vergleich längeren Prüfzeit zu rechnen. Im betrachteten Anwendungsfall der Offline-Prüfung hingegen ist eine hohe Messgenauigkeit entscheidend, wozu roboterbasierte Messsysteme durch 3D-Image-Stitching befähigt werden. In Kombination mit der automatisierten Erstellung des Prüfprogramms steht dem Anwender damit eine flexible, schnelle und kostengünstige Alternative zum konventionellen Koordinatenmessgerät zur Verfügung.



## Literatur

ABB AUTOMATION GMBH 2014a

ABB Automation GmbH, Hrsg. (2014a): IRB 140 Industrieroboter. Datenblatt. URL: <http://new.abb.com/products/robotics/de/industrieroboter/irb-140> (besucht am 05.01.2017).

ABB AUTOMATION GMBH 2014b

ABB Automation GmbH, Hrsg. (2014b): IRB 4600 Industrieroboter. Datenblatt. URL: <http://new.abb.com/products/robotics/de/industrieroboter/irb-4600> (besucht am 05.01.2017).

ABELE & REINHART 2011

Abele, E.; Reinhart, G.: Zukunft der Produktion. Herausforderungen, Forschungsfelder, Chancen. München: Carl Hanser. 2011. ISBN: 978-3-44642-595-8. DOI: 10.3139/9783446428058.

AMATO et al. 1998a

Amato, N. M.; Bayazit, O. B.; Dale, L. K.; Jones, C.; Vallejo, D.: Choosing good distance metrics and local planners for probabilistic roadmap methods. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Leuven, Belgium. 1998, S. 630–637. DOI: 10.1109/ROBOT.1998.677043.

AMATO et al. 1998b

Amato, N. M.; Bayazit, O. B.; Dale, L. K.; Jones, C.; Vallejo, D.: OBPRM: An Obstacle-Based PRM for 3D Workspaces. In: The algorithmic foundations of robotics. The algorithmic perspective. Hrsg. von Agarwal, P. K.; Kavraki, L. E.; Mason, M. T. Natick, Massachusetts: A K Peters. 1998. ISBN: 1-56881-081-4.

## Literatur

---

APPLEGATE et al. 2011

Applegate, D. L.; Bixby, R. E.; Chvatal, V.; Cook, W. J.: The Traveling Salesman Problem. A Computational Study. Princeton: Princeton University Press. 2011. ISBN: 978-0-69112-993-8.

ARUN et al. 1987

Arun, K. S.; Huang, T. S.; Blostein, S. D.: Least-Squares Fitting of Two 3-D Point Sets. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9 (1987) (5), S. 698–700. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1987.4767965.

BÄR 2001

Bär, G.: Geometrie. Eine Einführung für Ingenieure und Naturwissenschaftler. 2. Aufl. Mathematik für Ingenieure und Naturwissenschaftler. Wiesbaden: Vieweg + Teubner. 2001. ISBN: 3-32289-136-4. DOI: 10.1007/978-3-322-89136-5.

BARAFF & WITKIN 1992

Baraff, D.; Witkin, A.: Dynamic simulation of non-penetrating flexible bodies. Computer Graphics 26 (1992) (2), S. 303–308.

BARRAQUAND et al. 2000

Barraquand, J.; Kavraki, L. E.; Motwani, R.; Latombe, J.-C.; Li, T.-Y.; Raghavan, P.: A Random Sampling Scheme for Path Planning. In: Robotics Research. Hrsg. von Giralt, G.; Hirzinger, G. London: Springer. 2000, S. 249–264. ISBN: 978-1-44711-254-9. DOI: 10.1007/978-1-4471-0765-1\_28.

BENNAMOUN & MAMIC 2002

Bennamoun, M.; Mamic, G.: Object recognition. Fundamentals and case studies. Advances in pattern recognition. London: Springer. 2002. ISBN: 978-1-85233-398-0. DOI: 10.1007/978-1-4471-3722-1.

BERTAGNOLLI 2006

Bertagnolli, F.: Robotergestützte automatische Digitalisierung von Werkstückgeometrien mittels optischer Streifenprojektion. Dissertation. Messtechnik und Sensorik. Aachen: Shaker. 2006. ISBN: 3-83224-948-6.

BESL & MCKAY 1992

Besl, P. J.; McKay, N. D.: A Method for Registration of 3-D Shapes. IEEE Tran-

sactions on Pattern Analysis and Machine Intelligence 14 (1992) (2), S. 239–256. ISSN: 0162-8828. DOI: 10.1109/34.121791.

BEUTELSPACHER 2014

Beutelspacher, A.: Lineare Algebra. Eine Einführung in die Wissenschaft der Vektoren, Abbildungen und Matrizen. 8. Aufl. Wiesbaden: Springer. 2014. ISBN: 3-65802-413-5.

BIBER & STRASSER 2003

Biber, P.; Strasser, W.: The Normal Distributions Transform: A New Approach to Laser Scan Matching. In: International Conference on Intelligent Robots and Systems. IEEE Computer Society. Las Vegas, Nevada. 2003, S. 2743–2748. DOI: 10.1109/IROS.2003.1249285.

BOHLIN 1999

Bohlin, R.: Motion Planning for Industrial Robots. Thesis for the Degree of Licentiate of Engineering. Göteborg, Schweden: Chalmers University of Technology and Göteborg University. 1999.

BOHLIN & KAVRAKI 2000

Bohlin, R.; Kavraki, L. E.: Path Planning Using Lazy PRM. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. San Francisco, California. 2000. DOI: 10.1109/ROBOT.2000.844107.

BONGARDT 2004

Bongardt, T.: Methode zur Kompensation betriebsabhängiger Einflüsse auf die Absolutgenauigkeit von Industrierobotern. Dissertation. Forschungsberichte IWB 184. München: Utz. 2004. ISBN: 978-3-83160-332-9.

BOOR et al. 1999

Boor, V.; Overmars, M. H.; van der Stappen, A. F.: The Gaussian sampling strategy for probabilistic roadmap planners. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Detroit, Michigan. 1999, S. 1018–1023. DOI: 10.1109/ROBOT.1999.772447.

BOUAZIZ et al. 2013

Bouaziz, S.; Tagliasacchi, A.; Pauly, M.: Sparse Iterative Closest Point. Computer

## Literatur

---

Graphics Forum 32 (2013) (5), S. 113–123. ISSN: 0167-7055. DOI: 10.1111 /cgf.12178.

BRANDSTÄDT 1994

Brandstädt, A.: Graphen und Algorithmen. Leitfäden und Monographien der Informatik. Stuttgart: B.G. Teubner. 1994. ISBN: 978-3-51902-131-5.

BRANICKY et al. 2001

Branicky, M. S.; LaValle, S. M.; Olson, K.; Yang, L.: Quasi-Randomized Path Planning. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Seoul, South Korea. 2001, S. 1481–1487. DOI: 10.1109/ROBOT.2001.932820.

BREDIES & LORENZ 2011

Bredies, K.; Lorenz, D.: Mathematische Bildverarbeitung. Einführung in Grundlagen und moderne Theorie. 1. Aufl. Wiesbaden: Vieweg + Teubner. 2011. ISBN: 978-3-83481-037-3.

BREITBACH 2015

Breitbach, T.: Effizienzsteigerung der Zellenmodellierung für Industrieroboter durch hybride Simulationsmodelle. Dissertation. Werkzeugmaschinen 42. Aachen: Apprimus. 2015. ISBN: 978-3-86359-363-6.

BRÜGGEMANN & BREMER 2012

Brüggemann, H.; Bremer, P.: Grundlagen Qualitätsmanagement. Wiesbaden: Vieweg + Teubner. 2012. ISBN: 978-3-8348-1309-1. DOI: 10.1007/978-3-8348-8301-8.

BUCHMANN et al. 2015

Buchmann, T.; Baumgartl, J.; Henrich, D.; Westfechtel, B.: Robots and their Variability – A Societal Challenge and a Potential Solution. In: 5th International Workshop on Product Line Approaches in Software Engineering (PLEASE). IEEE Computer Society. Florence, Italy. 2015, S. 27–30. DOI: 10.1109/PLEASE.2015.15.

BUNGARTZ et al. 2009

Bungartz, H.-J.; Buchholz, M.; Pflüger, D.; Zimmer, S.: Modellbildung und

Simulation. Eine anwendungsorientierte Einführung. eXamen.press. Berlin: Springer. 2009. ISBN: 978-3540798095. DOI: 10.1007/978-3-540-79810-1.

BÜSING 2010

Büsing, C.: Graphen- und Netzwerkoptimierung. 1. Aufl. Heidelberg: Spektrum Akademischer Verlag. 2010. ISBN: 978-3-82742-422-8.

CARL ZEISS OPTOTECHNIK GMBH 2014

Carl Zeiss Optotechnik GmbH, Hrsg. (2014): COMET Automated - Der Neue innovative Sensor für die automatisierte 3D-Digitalisierung. Neubeuern. URL: <http://optotechnik.zeiss.com/produkte/3d-digitalisierung/streifenlichtprojektion> (besucht am 05.01.2017).

CHEN & MEDIONI 1991

Chen, Y.; Medioni, G.: Object Modeling by Registration of Multiple Range Images. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Sacramento, California. 1991, S. 2724–2729. DOI: 10.1109/ROBOT.1991.132043.

CHETVERIKOV et al. 2002

Chetverikov, D.; Svirko, D.; Stepanov, D.; Krsek, P.: The Trimmed Iterative Closest Point Algorithm. In: 16th International Conference on Pattern Recognition. IEEE Computer Society. Quebec City, Canada. 2002, S. 545–548. ISBN: 0-76951-695-5. DOI: 10.1109/ICPR.2002.1047997.

CHO et al. 2005

Cho, M.-W.; Lee, H.; Yoon, G.-S.; Choi, J.: A feature-based inspection planning system for coordinate measuring machines. The International Journal of Advanced Manufacturing Technology 26 (2005) (9-10), S. 1078–1087. ISSN: 0268-3768. DOI: 10.1007/s00170-004-2077-8.

CHOSSET et al. 2005

Choset, H. M.; Lynch, K.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L. E.; Thrun, S.: Principles of robot motion. Theory, algorithms, and implementation. Intelligent robotics and autonomous agents. Cambridge, Mass.: MIT Press. 2005. ISBN: 978-0-26203-327-5.

## Literatur

---

COHEN et al. 1995

Cohen, J.; Ming, L.; Dinesh, M.; Madhav, P.: I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In: Proceedings of the 1995 Symposium on Interactive 3D Graphics. ACM. 1995, S. 189–196.

CRAIG 2005

Craig, J. J.: Introduction to robotics. Mechanics and control. 3. Aufl. Upper Saddle River, New Jersey: Pearson/Prentice Hall. 2005. ISBN: 0-13123-629-6.

DEMANT et al. 2011

Demant, C.; Streicher-Abel, B.; Springhoff, A.: Industrielle Bildverarbeitung. Wie optische Qualitätskontrolle wirklich funktioniert. Berlin: Springer. 2011. ISBN: 3-64213-096-8.

DIESTEL 2006

Diestel, R.: Graphentheorie. 3. Aufl. Springer-Lehrbuch. Berlin: Springer. 2006. ISBN: 3-54021-391-0.

DIN EN ISO 9283

DIN Deutsches Institut für Normung e.V., Hrsg. (1999): DIN EN ISO 9283 - Industrieroboter - Leistungskenngrößen und zugehörige Prüfmethode. Berlin: Beuth.

DIN EN ISO 8373

DIN Deutsches Institut für Normung e.V., Hrsg. (2010): DIN EN ISO 8373 - Roboter und Robotikgeräte - Wörterbuch. Berlin: Beuth.

DRUON et al. 2006

Druon, S.; Aldon, M. J.; Crosnier, A.: Color Constrained ICP for Registration of Large Unstructured 3D Color Data Sets. In: International Conference on Information Acquisition (ICIA). IEEE Computer Society. Weihai, China. 2006, S. 249–255. DOI: 10.1109/ICIA.2006.306004.

DUNN & PARBERRY 2011

Dunn, F.; Parberry, I.: 3D math primer for graphics and game development. 2. Aufl. Boca Raton, Florida: A K Peters/CRC Press. 2011. ISBN: 978-1-56881-723-1.

ECKEY et al. 2002

Eckey, H.-F.; Kosfeld, R.; Rengers, M.: *Multivariate Statistik. Grundlagen - Methoden - Beispiele.* ger. Eckey, Hans-Friedrich (author.) Kosfeld, Reinhold (author.) Rengers, Martina (author.) Wiesbaden: Gabler Verlag, 2002. ISBN: 978-3-409-11969-6. DOI: 10.1007/978-3-322-84476-7. URL: <http://dx.doi.org/10.1007/978-3-322-84476-7>.

EGGERT et al. 1997

Eggert, D. W.; Lorusso, A.; Fisher, R. B.: Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications* 9 (1997) (5-6), S. 272–290. ISSN: 0932-8092. DOI: 10.1007/s001380050048.

ERICSON 2005

Ericson, C.: *Real-time collision detection.* Morgan Kaufmann series in interactive 3D technology. Amsterdam: Elsevier Morgan Kaufmann, 2005. ISBN: 1-55860-732-3.

ESTÉPAR et al. 2004

Estépar, R. S. J.; Brun, A.; Westin, C.-F.: Robust Generalized Total Least Squares Iterative Closest Point Registration. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Hrsg. von Hutchison, D.; Kanade, T.; Kittler, J.; Kleinberg, J. M.; Mattern, F.; Mitchell, J. C.; Naor, M.; Nierstrasz, O.; Pandu Rangan, C.; Steffen, B.; Sudan, M.; Terzopoulos, D.; Tygar, D.; Vardi, M. Y.; Weikum, G.; Barillot, C.; Haynor, D. R.; Hellier, P. *Lecture notes in computer science.* Berlin: Springer, 2004, S. 234–241. ISBN: 978-3-54022-976-6. DOI: 10.1007/978-3-540-30135-6\_29.

FANUC DEUTSCHLAND GMBH 2017

FANUC Deutschland GmbH, Hrsg. (2017): *M-710iC/70 Industrieroboter.* Datenblatt. URL: <http://www.fanuc.eu/de/de/roboter/roboterfilter-seite/m-710-serie/m-710ic-70> (besucht am 05. 01. 2017).

FELDMANN et al. 2013

Feldmann, K.; Schöppner, V.; Spur, G., Hrsg. (2013): *Handbuch Fügen, Handhaben und Montieren.* München: Carl Hanser, 2013. ISBN: 978-3-44642-827-0. DOI: 10.3139/9783446436565.

## Literatur

---

FELDMAR & AYACHE 1994

Feldmar, J.; Ayache, N.: Rigid, Affine and Locally Affine Registration of Free-Form Surfaces. Research Report. INRIA. 1994.

GECKS & HENRICH 2009

Gecks, T.; Henrich, D.: Sensor-based Online Planning of Time-optimized Paths in Dynamic Environments. In: German Workshop on Robotics. Braunschweig, Germany. 2009.

GERAERTS & OVERMARS 2004a

Geraerts, R.; Overmars, M. H.: A Comparative Study of Probabilistic Roadmap Planners. In: Algorithmic Foundations of Robotics V. Hrsg. von Boissonnat, J.-D.; Burdick, J.; Goldberg, K.; Hutchinson, S. Springer Tracts in Advanced Robotics. Berlin: Springer. 2004, S. 43–57. ISBN: 978-3-64207-341-0. DOI: 10.1007/978-3-540-45058-0\_4.

GERAERTS & OVERMARS 2004b

Geraerts, R.; Overmars, M. H.: Sampling Techniques for Probabilistic Roadmap Planners. In: Conference on Intelligent Autonomous Systems (IAS). IOS Press. 2004, S. 600–609.

GERKE 2015

Gerke, W.: Technische Assistenzsysteme. Vom Industrieroboter zum Roboterasistenten. Berlin: De Gruyter. 2015. ISBN: 978-3-11034-370-0.

GODIN et al. 1994

Godin, G.; Rioux, M.; Baribeau, R.: Three-dimensional registration using range and intensity information. In: Photonics for Industrial Applications. Hrsg. von El-Hakim, S. F. SPIE Proceedings. Boston, Massachusetts: SPIE. 1994, S. 279–290. DOI: 10.1117/12.189139.

GOM MBH 2010

GOM mbH, Hrsg. (2010): Anwendungsbeispiel - Automatisierte Roboterzelle zur Inspektion von Blechteilen. Die Zukunft der Messtechnik im Fahrzeugbau. Braunschweig. URL: [http://www.gom.com/fileadmin/user\\_upload/industries/automated\\_metrology\\_DE.pdf](http://www.gom.com/fileadmin/user_upload/industries/automated_metrology_DE.pdf) (besucht am 05.01.2017).

GRIMM 2004

Grimm, T.: User's guide to rapid prototyping. Dearborn, Michigan: Society of Manufacturing Engineers. 2004. ISBN: 978-0-87263-697-2.

GRIMSON et al. 1996

Grimson, W. L.; Ettinger, G. J.; White, S. J.; Lozano-Perez, T.; Wells, W. M.; Kikinis, R.: An Automatic Registration Method for Frameless Stereotaxy, Image Guided Surgery, and Enhanced Reality Visualization. IEEE Transactions on Medical Imaging 15 (1996) (2), S. 129–140. DOI: 10.1109/42.491415.

GUPTA 1998

Gupta, K.: Overview and State of the Art. In: Practical motion planning in robotics. Current approaches and future directions. Hrsg. von Gupta, K.; Pobil, A. P. del. Chichester: John Wiley & Sons. 1998, S. 3–8. ISBN: 0-47198-163-X.

HANSEN et al. 2007

Hansen, M. F.; Blas, M. R.; Larsen, R.: Mahalanobis distance based iterative closest point. In: Medical Imaging. Hrsg. von Pluim, J. P. W.; Reinhardt, J. M. SPIE Proceedings. San Diego, CA: SPIE. 2007. DOI: 10.1117/12.708205.

HATWIG et al. 2012

Hatwig, J.; Minnerup, P.; Zaeh, M. F.; Reinhart, G.: An automated path planning system for a robot with a laser scanner for remote laser cutting and welding. In: International Conference on Mechatronics and Automation (ICMA). IEEE Computer Society. Chengdu, China. 2012, S. 1323–1328.

HAUN 2013

Haun, M.: Handbuch Robotik. Programmieren und Einsatz intelligenter Roboter. 2. Aufl. VDI-Buch. Berlin: Springer. 2013. ISBN: 978-3-64239-858-2.

HORN 1987

Horn, B. K. P.: Closed-form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America 4 (1987) (4), S. 629. ISSN: 1084-7529. DOI: 10.1364/JOSAA.4.000629.

HORN et al. 1988

Horn, B. K. P.; Hilden, H. M.; Negahdaripour, S.: Closed-form solution of abso-

## Literatur

---

lute orientation using orthonormal matrices. *Journal of the Optical Society of America* 5 (1988) (7), S. 1127. ISSN: 1084-7529. DOI: 10.1364/JOSAA.5.001127.

HOROWITZ et al. 1998

Horowitz, E.; Sahni, S.; Rajasekran, S.: *Fundamentals of Computer Algorithms*. 2. Aufl. Computer software engineering series. Potomac, Maryland: Computer Science Press. 1998. ISBN: 978-8173716126.

HORSCH 2015

Horsch, J.: *Kostenrechnung*. Wiesbaden: Springer. 2015. ISBN: 978-3-65807-311-4. DOI: 10.1007/978-3-658-07312-1.

HSU et al. 2003

Hsu, D.; Jiang, T.; Reif, J.; Sun, Z.: The Bridge Test for Sampling Narrow Passages with Probabilistic Roadmap Planners. In: *International Conference on Robotics and Automation (ICRA)*. IEEE Computer Society. Taipei, Taiwan. 2003, S. 4420–4426. DOI: 10.1109/ROBOT.2003.1242285.

HSU et al. 2006

Hsu, D.; Latombe, J.-C.; Kurniawati, H.: On the Probabilistic Foundations of Probabilistic Roadmap Planning. *The International Journal of Robotics Research* 25 (2006) (7), S. 627–643. ISSN: 0278-3649. DOI: 10.1177/0278364906067174.

HUBBARD 1993

Hubbard, P. M.: Interactive collision detection. In: *Symposium on Research Frontiers in Virtual Reality*. IEEE Computer Society. San Jose, California. 1993, S. 24–31. DOI: 10.1109/VRAIS.1993.378267.

HUSSMANN 2015

Hußmann, S.: *Diskrete Mathematik erleben. Anwendungsbasierte und verstehensorientierte Zugänge*. 2. Aufl. Wiesbaden: Springer. 2015. ISBN: 978-3-65806-992-6.

HUSTY et al. 1997

Husty, M.; Karger, A.; Sachs, H.; Steinhilper, W.: *Kinematik und Robotik*. Berlin: Springer. 1997. ISBN: 978-3-64263-822-0.

IDS IMAGING DEVELOPMENT SYSTEMS GMBH 2017

IDS Imaging Development Systems GmbH, Hrsg. (2017): N10-1210-18 Stereo-3D-Kamera. Datenblatt. URL: [https://de.ids-imaging.com/IDS/datasheet\\_pdf.php?sku=AB00376&set=3D](https://de.ids-imaging.com/IDS/datasheet_pdf.php?sku=AB00376&set=3D) (besucht am 05. 01. 2017).

IFR STATISTICAL DEPARTMENT 2016

IFR Statistical Department, Hrsg. (2016): World robotics 2016. Industrial robots.

ISO/IEC 9075

ISO/IEC JTC 1/SC 32 Data management and interchange, Hrsg. (2011): ISO/IEC 9075 - Information technology - Database languages - SQL.

JÄHNE 2005

Jähne, B.: Digitale Bildverarbeitung. 6. Aufl. Berlin: Springer. 2005. ISBN: 3-54024-999-0. DOI: 10.1007/b138991.

JASTROW 2014

Jastrow, R.: Optische Messstechnik als Schlüsseltechnologie für die Fahrzeugfertigung der Zukunft. In: Koordinatenmesstechnik 2014. Schlüsseltechnologie für die Produktion von morgen. Hrsg. von VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik. VDI-Berichte 2243. VDI-Fachtagung Koordinatenmesstechnik. Düsseldorf: VDI-Verlag. 2014, S. 129–133. ISBN: 978-3-18092-243-0.

JOHNSON & BING KANG 1999

Johnson, A. E.; Bing Kang, S.: Registration and integration of textured 3D data. *Image and Vision Computing* 17 (1999) (2), S. 135–147. ISSN: 0262-8856. DOI: 10.1016/S0262-8856(98)00117-6.

JOST & HÜGLI 2002

Jost, T.; Hügli, H.: A Multi-Resolution Scheme ICP Algorithm for Fast Shape Registration. In: First International Symposium on 3D Data Processing Visualization and Transmission. IEEE Computer Society. Padova, Italy. 2002, S. 540–543. DOI: 10.1109/TDPVT.2002.1024114.

JOUNG et al. 2009

Joung, J. H.; An, K. H.; Kang, J. W.; Chung, M. J.; Yu, W.: 3D Environment Reconstruction Using Modified Color ICP Algorithm by Fusion of a Camera

and a 3D Laser Range Finder. In: International Conference on Intelligent Robots and Systems (IROS). IEEE Computer Society. St. Louis, Missouri. 2009, S. 3082–3088. DOI: 10.1109/IROS.2009.5354500.

KARPFINGER 2014

Karpfinger, C.: Höhere Mathematik in Rezepten. Berlin: Springer. 2014. ISBN: 978-3-64237-865-2. DOI: 10.1007/978-3-642-37866-9.

KAVRAKI et al. 1996

Kavraki, L. E.; Svestka, P.; Latombe, J.-C.; Overmars, M. H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12 (1996) (4), S. 566–580. ISSN: 1042-296X. DOI: 10.1109/70.508439.

KAVRAKI et al. 1998

Kavraki, L. E.; Kolountzakis, M. N.; Latombe, J.-C.: Analysis of probabilistic roadmaps for path planning. IEEE Transactions on Robotics and Automation 14 (1998) (1), S. 166–171. ISSN: 1042-296X. DOI: 10.1109/70.660866.

KAVRAKI & LATOMBE 1998

Kavraki, L. E.; Latombe, J.-C.: Probabilistic Roadmaps for Robot Path Planning. In: Practical motion planning in robotics. Current approaches and future directions. Hrsg. von Gupta, K.; Pobil, A. P. del. Chichester: John Wiley & Sons. 1998, S. 33–53. ISBN: 0-47198-163-X.

KEFERSTEIN & MARXER 2015

Keferstein, C. P.; Marxer, M.: Fertigungsmesstechnik. Wiesbaden: Springer. 2015. ISBN: 978-3-83482-582-7. DOI: 10.1007/978-3-8348-2583-4.

KIMMELMANN 1991

Kimmelman, W.: Robotergestützte Messtechnik: Flexible Bereitstellung und Bedienung von Mess- und Prüfvorrichtungen. Dissertation. Düsseldorf: VDI-Verlag. 1991. ISBN: 978-3-18401-189-5.

KLEIN 2005

Klein, R.: Algorithmische Geometrie. Grundlagen, Methoden, Anwendungen. 2. Aufl. eXamen.press. Berlin: Springer. 2005. ISBN: 978-3-54027-619-7.

KLEPPMANN 2013

Kleppmann, W.: Taschenbuch Versuchsplanung. Produkte und Prozesse optimieren. 8. Aufl. Hanser eLibrary. München: Carl Hanser. 2013. ISBN: 978-3-44643-752-4.

KOCKARA et al. 2007

Kockara, S.; Halic, T.; Iqbal, K.; Bayrak, C.; Rowe, R.: Collision detection: A survey. In: International Conference on Systems, Man and Cybernetics (ISIC). IEEE Computer Society. Montreal, Canada. 2007, S. 4046–4051. DOI: 10.1109/ICSMC.2007.4414258.

KRUMKE & NOLTEMEIER 2012

Krumke, S. O.; Noltemeier, H.: Graphentheoretische Konzepte und Algorithmen. 3. Aufl. Leitfäden der Informatik. Wiesbaden: Vieweg + Teubner. 2012. ISBN: 3-83482-264-7.

KUFFNER & LAVALLE 2000

Kuffner, J. J.; LaValle, S. M.: RRT-Connect: An Efficient Approach to Single-Query Path Planning. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. San Francisco, California. 2000, S. 995–1001. DOI: 10.1109/ROBOT.2000.844730.

LAMBRECHT et al. 2013

Lambrecht, J.; Walzel, H.; Krüger, J.: Robust finger gesture recognition on handheld devices for spatial programming of industrial robots. In: International Symposium on Robot and Human Interactive Communication (RO-MAN). IEEE Computer Society. Gyeongju, South Korea. 2013, S. 99–106. DOI: 10.1109/ROMAN.2013.6628462.

LATOMBE 1991

Latombe, J.-C.: Robot Motion Planning. 1. Aufl. The Springer International Series in Engineering and Computer Science 124. New York: Springer. 1991. ISBN: 978-0-79239-206-4. DOI: 10.1007/978-1-4615-4022-9.

LAVALLE 1998

LaValle, S. M.: Rapidly-Exploring Random Trees: A New Tool for Path Planning. Department of Computer Science. Technical Report. Iowa State University. 1998.

## Literatur

---

LAVALLE 2011

LaValle, S. M.: Motion Planning. The Essentials. IEEE Robotics & Automation Magazine 18 (2011) (1), S. 79–89. ISSN: 1070-9932. DOI: 10.1109 / MRA.2011. 940276.

LAVALLE & KUFFNER 2001

LaValle, S. M.; Kuffner, J. J.: Randomized Kinodynamic Planning. The International Journal of Robotics Research 20 (2001) (5), S. 378–400. ISSN: 0278-3649. DOI: 10.1177/02783640122067453.

LAVALLE 2006

LaValle, S. M.: Planning algorithms. Cambridge, New York: Cambridge University Press. 2006. ISBN: 978-0-52186-205-9.

LAVALLE et al. 2004

LaValle, S. M.; Branicky, M. S.; Lindemann, S. R.: On the Relationship between Classical Grid Search and Probabilistic Roadmaps. The International Journal of Robotics Research 23 (2004) (7-8), S. 673–692. ISSN: 0278-3649. DOI: 10.1177/ 0278364904045481.

LEE & PARK 2000

Lee, K. H.; Park, H.-p.: Automated inspection planning of free-form shape parts by laser scanning. Robotics and Computer-Integrated Manufacturing 16 (2000) (4), S. 201–210. ISSN: 0736-5845. DOI: 10.1016/S0736-5845(99)00060-5.

LETTENBAUER 2002

Lettenbauer, H.: System zum Einsatz von Industrierobotern in der fertigungsnahen Qualitätsprüfung. Dissertation. Berichte aus der Robotik. Aachen: Shaker. 2002. ISBN: 3-83220-941-7.

LEVOY et al. 2000

Levoy, M.; Ginsberg, J.; Shade, J.; Fulk, D.; Pulli, K.; Curless, B.; Rusinkiewicz, S.; Koller, D.; Pereira, L.; Ginzton, M.; Anderson, S.; Davis, J.: The digital Michelangelo project. In: 27th annual conference on Computer graphics and interactive techniques. Hrsg. von Brown, J. R.; Akeley, K. New Orleans, Louisiana. 2000, S. 131–144. DOI: 10.1145/344779.344849.

M. C. LIN & CANNY 1991

Lin, M. C.; Canny, J. F.: A fast algorithm for incremental distance calculation. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Sacramento, California. 1991, S. 1008–1014. DOI: 10.1109/ROBOT.1991.131723.

M. LIN & GOTTSCHALK 1998

Lin, M.; Gottschalk, S.: Collision detection between geometric models: A survey. In: IMA Conference on Mathematics of Surfaces. Institute of Mathematics and its Applications. 1998, S. 602–608.

LINDEMANN & LAVALLE 2005

Lindemann, S. R.; LaValle, S. M.: Current Issues in Sampling-Based Motion Planning. In: The Eleventh International Symposium Robotics Research. Hrsg. von Siciliano, B.; Khatib, O.; Dario, P.; Chatila, R. Springer Tracts in Advanced Robotics. Berlin: Springer. 2005, S. 36–54. ISBN: 978-3-54023-214-8. DOI: 10.1007/11008941\_5.

LÖFFLER-MANG 2012

Löffler-Mang, M.: Optische Sensorik. Lasertechnik, Experimente, Light Barriers. 1. Aufl. Fertigung. Wiesbaden: Vieweg + Teubner. 2012. ISBN: 978-3-83481-449-4. DOI: 10.1007/978-3-8348-8308-7.

LUDWIG 2011

Ludwig, R.: Robotergestützte Qualitätsprüfung in der Automobilfertigung. Dissertation. Berichte aus dem Institut für Konstruktions- und Fertigungstechnik 18. Aachen: Shaker. 2011. ISBN: 978-3-83229-787-9.

LUX & REINHART 2015

Lux, G.; Reinhart, G.: An approach for the automated self-calibration of robot-based inspection systems. In: 7th International Conference on Cybernetics and Intelligent Systems (CIS) and Conference on Robotics, Automation and Mechatronics (RAM). IEEE Computer Society. Siem Reap, Cambodia. 2015, S. 106–111. DOI: 10.1109/ICCIS.2015.7274605.

MAHMUD et al. 2011

Mahmud, M.; Joannic, D.; Roy, M.; Isheil, A.; Fontaine, J.-F.: 3D part inspection

path planning of a laser scanner with control on the uncertainty. *Computer-Aided Design* 43 (2011) (4), S. 345–355. ISSN: 0010-4485. DOI: 10.1016/j.cad.2010.12.014.

MAIER-HEIN et al. 2012

Maier-Hein, L.; Franz, A. M.; dos Santos, T. R.; Schmidt, M.; Fangerau, M.; Meinger, H.-P.; Fitzpatrick, J. M.: Convergent Iterative Closest-Point Algorithm to Accomodate Anisotropic and Inhomogenous Localization Error. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012) (8), S. 1520–1532. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2011.248.

MASUDA & YOKOYA 1995

Masuda, T.; Yokoya, N.: A Robust Method for Registration and Segmentation of Multiple Range Images. *Computer Vision and Image Understanding* 61 (1995) (3), S. 295–307. ISSN: 1077-3142. DOI: 10.1006/cviu.1995.1024.

MASUDA et al. 1996

Masuda, T.; Sakaue, K.; Yokoya, N.: Registration and Integration of Multiple Range Images for 3-D Model Construction. In: 13th International Conference on Pattern Recognition. IEEE Computer Society, Vienna, Austria. 1996, S. 879–883. DOI: 10.1109/ICPR.1996.546150.

MATOUŠEK 2009

Matoušek, J.: Geometric Discrepancy. An Illustrated Guide. *Algorithms and combinatorics* 13. New York: Springer. 2009. ISBN: 3-64203-942-1.

MINGUEZ et al. 2005

Minguez, J.; Lamiroux, F.; Montesano, L.: Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society, Barcelona, Spain. 2005, S. 3557–3563. DOI: 10.1109/ROBOT.2005.1570661.

NAUMANN et al. 2014

Naumann, H.; Schröder, G.; Löffler-Mang, M.: *Handbuch Bauelemente der Optik. Grundlagen, Werkstoffe, Geräte, Messtechnik*. 7. Aufl. Carl Hanser. 2014. ISBN: 978-3-44642-625-2. DOI: 10.3139/9783446441156.

NEUGEBAUER 1997

Neugebauer, P. J.: Geometrical Cloning of 3D Objects via Simultaneous Registration of Multiple Range Images. In: International Conference on Shape Modeling and Applications. IEEE Computer Society. Aizu-Wakamatsu, Japan. 1997, S. 130–139. DOI: 10.1109/SMA.1997.634890.

P. NEWMAN & KIN HO 2005

Newman, P.; Kin Ho: SLAM-Loop Closing with Visually Salient Features. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Barcelona, Spain. 2005, S. 635–642. DOI: 10.1109/ROBOT.2005.1570189.

T. S. NEWMAN & JAIN 1995

Newman, T. S.; Jain, A. K.: A Survey of Automated Visual Inspection. *Computer Vision and Image Understanding* 61 (1995) (2), S. 231–262. ISSN: 1077-3142. DOI: 10.1006/cviu.1995.1017.

OSRAM OPTO SEMICONDUCTORS GMBH 2016

OSRAM Opto Semiconductors GmbH, Hrsg. (2016): OSOLON Black Series (850 nm) 90° - SFH 4715S. Datenblatt. URL: [http://www.osram-os.com/Graphics/XPic4/00204341\\_0.pdf/SFH%204715S.pdf](http://www.osram-os.com/Graphics/XPic4/00204341_0.pdf/SFH%204715S.pdf) (besucht am 05.01.2017).

PAN et al. 2012

Pan, Z.; Polden, J.; Larkin, N.; van Duin, S.; Norrish, J.: Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing* 28 (2012) (2), S. 87–94. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2011.08.004.

PFEIFER & SCHMITT 2010

Pfeifer, T.; Schmitt, R.: *Fertigungsmesstechnik*. 3. Aufl. München: Oldenbourg. 2010. ISBN: 3-48659-202-5.

PFEIFFER & HOLZL 1995

Pfeiffer, F.; Holzl, J.: Parameter identification for industrial robots. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Nagoya, Japan. 1995, S. 1468–1476. DOI: 10.1109/ROBOT.1995.525483.

POTT et al. 2013

Pott, A.; Naumann, M.; Hägele, M.; Verl, A.: Industrieroboter und Handhabungsgeräte. In: Handbuch Fügen, Handhaben und Montieren. Hrsg. von Feldmann, K.; Schöppner, V.; Spur, G. München: Carl Hanser. 2013, S. 316–336. ISBN: 978-3-44642-827-0.

PUENTE LEÓN & KIENCKE 2012

Puente León, F.; Kiencke, U.: Messtechnik. Systemtheorie für Ingenieure und Informatiker. 9. Aufl. SpringerLink. Berlin: Springer. 2012. ISBN: 978-3-64230-074-5.

PULLI 1999

Pulli, K.: Multiview registration for large data sets. In: Second International Conference on 3-D Digital Imaging and Modeling. IEEE Computer Society. Ottawa, Canada. 1999, S. 160–168. DOI: 10.1109/IM.1999.805346.

REED & ALLEN 2000

Reed, M. K.; Allen, P. K.: Constraint-based sensor planning for scene modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) (12), S. 1460–1467. ISSN: 0162-8828. DOI: 10.1109/34.895979.

REINELT 1994

Reinelt, G.: The Traveling Salesman. Computational solutions for TSP applications. Lecture notes in computer science 840. Berlin: Springer. 1994. ISBN: 3-54058-334-3.

REINHART & TEKOUO 2009

Reinhart, G.; Tekouo, W.: Automatic programming of robot-mounted 3D optical scanning devices to easily measure parts in high-variant assembly. CIRP Annals - Manufacturing Technology 58 (2009) (1), S. 25–28. ISSN: 0007-8506.

REINHART et al. 1998

Reinhart, G.; Gräser, R.-G.; Klingel, R.: Qualification of Standard Industrial Robots to Cope with Sophisticated Assembly Tasks. CIRP Annals - Manufacturing Technology 47 (1998) (1), S. 1–4. ISSN: 0007-8506. DOI: 10.1016/S0007-8506(07)62772-3.

RIDENE & GOULETTE 2009

Ridene, T.; Goulette, F.: Registration of Fixed-and-mobile- Based Terrestrial Laser Data Sets with DSM. In: International Symposium on Computational Intelligence in Robotics and Automation (CIRA). IEEE Computer Society. Daejeon, South Korea. 2009, S. 375–380. DOI: 10.1109/CIRA.2009.5423176.

RÖSCH 2015

Rösch, O.: Steigerung der Arbeitsgenauigkeit bei der Fräsbearbeitung metallischer Werkstoffe mit Industrierobotern. Dissertation. Forschungsberichte IWB 302. München: Utz. 2015. ISBN: 978-3-83164-486-5.

ROSIN & ELLIS 1995

Rosin, P. L.; Ellis, T. J.: Image Difference Threshold Strategies and Shadow Detection. In: Proceedings of the British Conference on Machine Vision. Hrsg. von Pycock, D. Surrey, United Kingdom: BMVA Press. 1995, S. 347–356. ISBN: 0-95218-982-8. DOI: 10.5244/C.9.35.

ROTTMANN 2003

Rottmann, H.: C# .NET mit Methode. Professionelle Software entwickeln mit C# und .NET: Grundlagen, Windows, ADO.NET, ASP.NET und Remoting. Wiesbaden: Vieweg + Teubner. 2003. ISBN: 978-3-52805-845-6. DOI: 10.1007/978-3-322-83096-8.

RUSINKIEWICZ & LEVOY 2001

Rusinkiewicz, S.; Levoy, M.: Efficient Variants of the ICP Algorithm. In: Third International Conference on 3-D Digital Imaging and Modeling. IEEE Computer Society. Los Alamitos, California. 2001, S. 145–152.

SAHA et al. 2003

Saha, M.; Sanchez-Ante, G.; Latombe, J.-C.: Planning multi-goal tours for robot arms. In: International Conference on Robotics and Automation (ICRA). IEEE Computer Society. Taipei, Taiwan. 2003, S. 3797–3803. DOI: 10.1109/ROBOT.2003.1242179.

SÁNCHEZ & LATOMBE 2003

Sánchez, G.; Latombe, J.-C.: A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking. In: Robotics Research. Hrsg. von

## Literatur

---

Jarvis, R. A.; Zelinsky, A. Springer Tracts in Advanced Robotics. Berlin: Springer. 2003, S. 403–417. ISBN: 978-3-54000-550-6. DOI: 10.1007/3-540-36460-9\_27.

SCHMITT 2000

Schmitt, R.: Aufbau flexibler Mess- und Prüfstationen für die automatisierte Montage. Dissertation. Berichte aus der Produktionstechnik 26. Aachen: Shaker. 2000. ISBN: 3-82657-892-9.

SCHMITT et al. 2010

Schmitt, R.; Schönberg, A.; Damm, B.: Indoor-GPS based robots as a key technology for versatile production. In: ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics). German Conference on Robotics. Berlin: VDE Verlag. 2010, S. 199–205. ISBN: 978-3-80073-273-9.

SCHRÖER 1993

Schröer, K.: Identifikation von Kalibrationsparametern kinematischer Ketten. Dissertation. Produktionstechnik - Berlin 126. München: Carl Hanser. 1993. ISBN: 3-44617-650-0.

SCOTT et al. 2003

Scott, W. R.; Roth, G.; Rivest, J.-F.: View planning for automated three-dimensional object reconstruction and inspection. ACM Computing Surveys 35 (2003) (1), S. 64–96. ISSN: 0360-0300. DOI: 10.1145/641865.641868.

SEGAL et al. 2009

Segal, A. V.; Haehnel, D.; Thrun, S.: Generalized-ICP. Robotics: Science and Systems 2 (2009) (4).

SHARP et al. 2002

Sharp, G. C.; Lee, S. W.; Wehe, D. K.: ICP Registration using Invariant Features. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) (1), S. 90–102. ISSN: 0162-8828. DOI: 10.1109/34.982886.

SHI & XI 2008

Shi, Q.; Xi, N.: Automated Data Processing for a Rapid 3D Surface Inspection System. In: International Conference on Robotics and Automation (ICRA). IEEE

Computer Society. Pasadena, California. 2008, S. 3939–3944. DOI: 10.1109 / ROBOT.2008.4543816.

SHIAKOLAS et al. 2002

Shiakolas, P. S.; Conrad, K. L.; Yih, T. C.: On the accuracy, repeatability, and degree of influence of kinematics parameters for industrial robots. *International Journal of Modelling and Simulation* 22 (2002) (4), S. 245–254. ISSN: 0228-6203.

SICILIANO & KHATIB 2008

Siciliano, B.; Khatib, O., Hrsg. (2008): *Springer Handbook of Robotics*. Berlin: Springer. 2008. ISBN: 978-3-54023-957-4.

SIMON 1996

Simon, D. A.: *Fast and Accurate Shape-Based Registration*. Dissertation. Pittsburgh, Pennsylvania: Carnegie Mellon University. 1996.

SONKA et al. 1993

Sonka, M.; Hlavac, V.; Boyle, R.: *Image Processing, Analysis and Machine Vision*. Boston, Massachusetts: Springer. 1993. ISBN: 978-0-41245-570-4. DOI: 10.1007/978-1-4899-3216-7.

SPILLNER 2015

Spillner, R.: *Einsatz und Planung von Roboterassistenz zur Berücksichtigung von Leistungswandlungen in der Produktion*. Dissertation. Forschungsberichte IWB 296. München: Utz. 2015. ISBN: 978-3-83164-450-6.

SURMANN et al. 2003

Surmann, H.; Nüchter, A.; Hertzberg, J.: An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems* 45 (2003) (3-4), S. 181–198. ISSN: 0921-8890. DOI: 10.1016/j.robot.2003.09.004.

SÜSSE & RODNER 2014

Süße, H.; Rodner, E.: *Bildverarbeitung und Objekterkennung*. Wiesbaden: Springer. 2014. ISBN: 978-3-83482-605-3. DOI: 10.1007/978-3-8348-2606-0.

## Literatur

---

SZELISKI 2011

Szeliski, R.: Computer vision. Algorithms and applications. Texts in computer science. London: Springer. 2011. ISBN: 1-84882-935-3.

TARABANIS et al. 1995

Tarabanis, K. A.; Allen, P. K.; Tsai, R. Y.: A survey of sensor planning in computer vision. *IEEE Transactions on Robotics and Automation* 11 (1995) (1), S. 86–104. ISSN: 1042-296X. DOI: 10.1109/70.345940.

TEKOUO MOUTCHIHU 2012

Tekouo Moutchiho, W. B.: A New Programming Approach for Robot-based Flexible Inspection Systems. Dissertation. Forschungsberichte IWB. München: Utz. 2012. ISBN: 978-3-83164-247-2.

THYSSEN et al. 2016

Thyssen, L.; Seim, P.; Störkle, D. D.; Kuhlenkötter, B.: On the increase of geometric accuracy with the help of stiffening elements for robot-based incremental sheet metal forming. In: Proceedings of the 19th International ESAFORM Conference on Material Forming. AIP Conference Proceedings. Nantes, France. 2016. DOI: 10.1063/1.4963461.

TIPLER et al. 2015

Tipler, P. A.; Mosca, G.; Wagner, J.: Physik. Berlin: Springer. 2015. ISBN: 978-3-64254-165-0. DOI: 10.1007/978-3-642-54166-7.

TRUCCO et al. 1999

Trucco, E.; Fusiello, A.; Roberto, V.: Robust motion and correspondence of noisy 3-D point sets with missing data. *Pattern Recognition Letters* 20 (1999) (9), S. 889–898. ISSN: 0167-8655. DOI: 10.1016/S0167-8655(99)00055-0.

TRUCCO et al. 1997

Trucco, E.; Umasuthan, M.; Wallace, A. M.; Roberto, V.: Model-based planning of optimal sensor placements for inspection. *IEEE Transactions on Robotics and Automation* 13 (1997) (2), S. 182–194. ISSN: 1042-296X. DOI: 10.1109/70.563641.

TURK & LEVOY 1994

Turk, G.; Levoy, M.: Zippered Polygon Meshes from Range Images. In: 21st

Annual Conference on Computer Graphics and Interactive Techniques. Hrsg. von Schweitzer, D.; Glassner, A.; Keeler, M. 1994, S. 311–318. DOI: 10.1145/192161.192241.

ULRICH et al. 2014

Ulrich, M.; Lux, G.; Piprek, T.; Reinhart, G.: Analysis and Visualisation of the Positioning Accuracy and Underlying Effects of Industrial Robots. In: WGP Congress 2014. Progress in Production Engineering. Hrsg. von Merklein, M.; Franke, J.; Hagenah, H. Advanced materials research 1018. Trans Tech Publications. 2014, S. 15–22. ISBN: 3-03835-245-4.

ULRICH et al. 2015

Ulrich, M.; Forstner, A.; Reinhart, G.: High-accuracy 3D image stitching for robot-based inspection systems. In: International Conference on Image Processing (ICIP). IEEE Computer Society. Quebec City, Canada. 2015, S. 1011–1015. DOI: 10.1109/ICIP.2015.7350952.

ULRICH et al. 2016

Ulrich, M.; Lux, G.; Jürgensen, L.; Reinhart, G.: Automated and Cycle Time Optimized Path Planning for Robot-Based Inspection Systems. Procedia CIRP 44 (2016), S. 377–382. ISSN: 2212-8271. DOI: 10.1016/j.procir.2016.02.021.

VDI 2861

VDI-Gesellschaft Produktionstechnik, Hrsg. (1988): VDI 2861 - Kenngrößen für Industrieroboter. Düsseldorf: VDI-Verlag.

VDI 2634

VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, Hrsg. (2002): VDI 2634 - Optische 3-D-Messsysteme. Düsseldorf: VDI-Verlag.

VOGEL et al. 2009

Vogel, O.; Arnold, I.; Chughtai, A.; Ihler, E.; Kehrer, T.; Mehlig, U.; Zdun, U.: Software-Architektur. Grundlagen - Konzepte - Praxis. 2. Aufl. Heidelberg: Spektrum Akademischer Verlag. 2009. ISBN: 978-3-82741-933-0. DOI: 10.1007/978-3-8274-2267-5.

## Literatur

---

VOGL 2009

Vogl, W.: Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern. Dissertation. Forschungsberichte IWB 228. München: Utz. 2009. ISBN: 978-3-83160-869-0.

VOLKMANN 1996

Volkman, L.: Fundamente der Graphentheorie. Springer Lehrbuch Mathematik. Wien: Springer. 1996. ISBN: 978-3-21182-774-1.

VÖLLNER 2010

Völlner, G.: Rührreißschweißen mit Schwerlast-Industrierobotern. Dissertation. Forschungsberichte IWB 235. München: Utz. 2010. ISBN: 978-3-83160-955-0.

WALKER et al. 1991

Walker, M. W.; Shao, L.; Volz, R. A.: Estimating 3-D location parameters using dual number quaternions. CVGIP: Image Understanding 54 (1991) (3), S. 358–367. ISSN: 1049-9660. DOI: 10.1016/1049-9660(91)90036-O.

WEBER 2009

Weber, W.: Industrieroboter. Methoden der Steuerung und Regelung. 2. Aufl. München: Carl Hanser. 2009. ISBN: 978-3-44641-031-2.

WECKENMANN 2012

Weckenmann, A.: Koordinatenmesstechnik. Flexible Strategien für funktions- und fertigungsgerechtes Prüfen. 2. Aufl. München: Carl Hanser. 2012. ISBN: 978-3-44640-739-8.

WEIDNER 2014

Weidner, G. E.: Qualitätsmanagement. München: Carl Hanser. 2014. ISBN: 978-3-44643-839-2. DOI: 10.3139/9783446438507.

WEIK 1997

Weik, S.: Registration of 3-D partial surface models using luminance and depth information. In: International Conference on Recent Advances in 3-D Digital Imaging and Modeling. IEEE Computer Society. Ottawa., Canada. 1997, S. 93–100. DOI: 10.1109/IM.1997.603853.

WIEST 2001

Wiest, U.: Kinematische Kalibrierung von Industrierobotern. Dissertation. Berichte aus der Automatisierungstechnik. Aachen: Shaker. 2001. ISBN: 978-3-82658-609-5.

WU et al. 2015

Wu, Q.; Zou, W.; Xu, D.: Viewpoint Planning for Freeform Surface Inspection Using Plane Structured Light Scanners. *International Journal of Automation and Computing* (2015). ISSN: 1476-8186.

YAU & MENQ 1991

Yau, H.-T.; Menq, C.-H.: Path planning for automated dimensional inspection using coordinate measuring machines. In: *International Conference on Robotics and Automation (ICRA)*. IEEE Computer Society. Sacramento, California. 1991, S. 1934–1939. DOI: 10.1109/ROBOT.1991.131910.

ZHANG 1994

Zhang, Z.: Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *International Journal of Computer Vision* 12 (1994) (2), S. 119–152. ISSN: 0920-5691.

ZHAO et al. 2012

Zhao, H.; Kruth, J.-P.; van Gestel, N.; Boeckmans, B.; Bleys, P.: Automated dimensional inspection planning using the combination of laser scanner and tactile probe. *Measurement* 45 (2012) (5), S. 1057–1066. ISSN: 0263-2241. DOI: 10.1016/j.measurement.2012.01.037.

ZINSSER et al. 2003

Zinßer, T.; Schmidt, J.; Niemann, H.: A refined ICP algorithm for robust 3-D correspondence estimation. In: *International Conference on Image Processing (ICIP)*. IEEE Computer Society. Barcelona, Spain. 2003. DOI: 10.1109/ICIP.2003.1246775.



## **Anhang**

## A.1 Zuordnung der Parametersätze

Parametersatz	$k_{gauss}$	$k_{thresh}$	$n_{add}$	$\varepsilon_{ref}$ in $\mu\text{m}$	$u_{ref}$ in $\mu\text{m}$
1	$3 \times 3$	0,2	20	26,41	11,55
2	$3 \times 3$	0,2	50	40,55	26,14
3	$3 \times 3$	0,2	80	53	33,93
4	$3 \times 3$	0,2	110	73,77	53,39
5	$3 \times 3$	0,3	20	18,51	14,5
6	$3 \times 3$	0,3	50	11,36	4,61
7	$3 \times 3$	0,3	80	18,98	21,33
8	$3 \times 3$	0,3	110	25,58	35,49
9	$3 \times 3$	0,4	20	21,47	18,04
10	$3 \times 3$	0,4	50	12,37	7,29
11	$3 \times 3$	0,4	80	13,74	5,96
12	$3 \times 3$	0,4	110	23,46	26,89
13	$5 \times 5$	0,2	20	44,02	15,38
14	$5 \times 5$	0,2	50	14,15	7,42
15	$5 \times 5$	0,2	80	56,27	32,79
16	$5 \times 5$	0,2	110	70,72	43,29
17	$5 \times 5$	0,3	20	27,26	37,64
18	$5 \times 5$	0,3	50	8,9	4,53
19	$5 \times 5$	0,3	80	8,86	3,91
20	$5 \times 5$	0,3	110	19,23	33,31
21	$5 \times 5$	0,4	20	33,93	23,86
22	$5 \times 5$	0,4	50	22,05	10,38
23	$5 \times 5$	0,4	80	26,78	29,41
24	$5 \times 5$	0,4	110	28,75	32,2
25	$7 \times 7$	0,2	20	126,78	32,01
26	$7 \times 7$	0,2	50	37,78	14,87

Parametersatz	$k_{gauss}$	$k_{thresh}$	$n_{add}$	$\varepsilon_{ref}$ in $\mu\text{m}$	$u_{ref}$ in $\mu\text{m}$
27	$7 \times 7$	0,2	80	22,25	21,32
28	$7 \times 7$	0,2	110	65,16	39,61
29	$7 \times 7$	0,3	20	48,54	38,42
30	$7 \times 7$	0,3	50	12,4	5,33
31	$7 \times 7$	0,3	80	12,45	11
32	$7 \times 7$	0,3	110	17,17	24,54
33	$7 \times 7$	0,4	20	55,02	45,27
34	$7 \times 7$	0,4	50	54,98	53,35
35	$7 \times 7$	0,4	80	71,35	64,22
36	$7 \times 7$	0,4	110	81,39	68,94
37	$9 \times 9$	0,2	20	75,34	33,63
38	$9 \times 9$	0,2	50	46,6	22,29
39	$9 \times 9$	0,2	80	96,8	27,85
40	$9 \times 9$	0,2	110	102,01	22,73
41	$9 \times 9$	0,3	20	46,53	34,5
42	$9 \times 9$	0,3	50	23,62	17,9
43	$9 \times 9$	0,3	80	21,15	20,13
44	$9 \times 9$	0,3	110	22,84	24,47
45	$9 \times 9$	0,4	20	132,14	86,02
46	$9 \times 9$	0,4	50	58,97	33,13
47	$9 \times 9$	0,4	80	41,85	28,47
48	$9 \times 9$	0,4	110	44,84	34,77

### A.2 Betreute Studienarbeiten

Im Rahmen dieser Dissertation entstanden am Institut für Werkzeugmaschinen und Betriebswissenschaften (*iwb*) in den Jahren 2012 bis 2016 unter der wissenschaftlichen Anleitung des Autors die im Folgenden aufgeführten studentischen Arbeiten. Deren Ergebnisse flossen teilweise in die vorliegende Dissertation ein. Der Autor dankt allen beteiligten Studierenden herzlich für ihr Engagement bei der Unterstützung seiner wissenschaftlichen Arbeit.

- HASLINGER, PATRICK: Registrierung von Punktwolken durch 3D-Matching (2016)
- KUTTER, SEBASTIAN: Integration eines robotergestützten Messsystems basierend auf 3D-Image-Stitching (2016)
- VON RHEIN, LARA: Modellierung der Fehlereinflussfaktoren auf die Genauigkeit beim 3D-Image-Stitching (2016)
- FORSTNER, ANDREAS: Automatisierte Bahnplanung zur roboterbasierten Qualitätssicherung mittels 3D-Image-Stitching (2015)
- KUTTER, SEBASTIAN: Entwicklung eines laufzeitoptimierten Kollisionsdetektors für Robotertrajektorienplanung (2015)
- VON RHEIN, LARA: Analyse und Minimierung der Fehler beim 3D-Image-Stitching (2015)
- SCHMIDT, MANUEL: Trajektorienplanung für ein roboterbasiertes Messsystem (2015)
- BERTHOLD, JULIANA: Entwicklung eines Demonstrators zur Bauteilvermessung mittels optischer Referenzierung (2015)
- FORSTNER, ANDREAS: Hochgenaues 3D-Image-Stitching für ein roboterbasiertes Messsystem (2015)
- JÜRGENSEN, LEILA: Bahnplanung für einen Industrieroboter zur kollisionsfreien und taktzeitoptimierten Vermessung eines Karosserieteils (2014)

- 
- BURGHADT, MORITZ: Projektion von Markern auf homogenen Oberflächen für die Referenzierung durch Image Stitching (2014)
  - BAKER, THOMAS: Automatisierte Erstellung von Roboterposen für ein Inspektionssystem (2014)
  - HOLLSTEIN, KAI: 3D-Image-Stitching zur Referenzierung in der roboterbasierten Messtechnik (2014)
  - EBERLE, ROMAN: Bewertung verschiedener Sensorkonzepte für die roboterbasierte Qualitätssicherung (2014)
  - PIPREK, THOMAS: Untersuchung und Visualisierung von Fehlereinflüssen auf die Genauigkeit von Industrierobotern (2014)
  - WEIGAND, MAXIMILIAN: Entwicklung eines kameraoptischen Messverfahrens zur Anwendung in einem Mikromontagesystem (2013)
  - HAAS, MICHAEL: Konzeptionierung und Implementierung einer Serviceanwendung an einem Industrieroboter (2013)
  - HINTER, MANFRED: Einheitliche Schnittstelle zur Ansteuerung und Programmierung von Industrie-Kameras (2013)
  - PAUCKER, TOBIAS: Automatische Erstellung von verteilter Kommunikation bei der Programmierung von Robotersystemen (2013)