

Reihe 2

Fertigungstechnik

Nr. 697

M.Sc. Philipp Sommer (Hrsg.),
Stuttgart

iWindow – Intelligentes Maschinenfenster

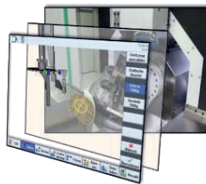


Abschlussbericht

Verbundforschungsprojekt

iWindow – Intelligentes Maschinenfenster

Ein im Rahmen des Förderprogramms *IKT 2020 – Forschung für Innovationen* unter der Bekanntmachung *Virtuelle Techniken für die Fabrik der Zukunft - Ein Beitrag zum Zukunftsprojekt Industrie 4.0* des Bundesministeriums für Bildung und Forschung (BMBF) durchgeführtes Verbundforschungsprojekt.



Am Verbundforschungsprojekt *iWindow* beteiligten sich folgende Firmen und Institutionen:

- DMG MORI AKTIENGESELLSCHAFT, Bielefeld
- Bosch Rexroth AG, Lohr am Main
- Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW), Universität Stuttgart
- ModuleWorks GmbH, Aachen
- TRUMPF GmbH + Co. KG, Ditzingen
- Höchstleistungsrechenzentrum Stuttgart (HLRS), Universität Stuttgart
- ISG Industrielle Steuerungstechnik GmbH, Stuttgart

Projektleitung: M.Sc. Philipp Sommer, ISW, Universität Stuttgart

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter den Förderkennzeichen 01IM14003A - G gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Institut für Steuerungstechnik der Werkzeugmaschinen
und Fertigungseinrichtungen der Universität Stuttgart

2018

Fortschritt-Berichte VDI

Reihe 2

Fertigungstechnik

M.Sc. Philipp Sommer (Hrsg.),
Stuttgart

Nr. 697

iWindow - Intelligentes
Maschinenfenster

VDI verlag

Sommer, Philipp

iWindow – Intelligentes Maschinenfenster

Fortschr.-Ber. VDI Reihe 2 Nr. 697. Düsseldorf: VDI Verlag 2018.

148 Seiten, 123 Bilder, 0 Tabellen.

ISBN 978-3-18-369702-1, ISSN 0178-9406,

€ 57,00/VDI-Mitgliederpreis € 51,30.

Für die Dokumentation: Augmented Reality – Virtual Reality – Visualisierung – Maschinenfenster – Mehrwertdienste – Assistenzsysteme – Simulation – Mensch-Maschine-Schnittstelle – Kamerakalibrierung – Digitaler Zwilling

Das Verbundforschungsprojekt iWindow: Intelligentes Maschinenfenster beschäftigte sich mit der visuellen Unterstützung von Maschinenbedienern an Werkzeugmaschinen. Diese konnten bisher nur auf wenige bis keine Systeme, die sie bei ihren täglichen Aufgaben direkt an der Werkzeugmaschine unterstützen, zurückgreifen. Das Forschungsprojekt verbindet reale und virtuelle Welt in der Werkzeugmaschine durch Technologien wie Virtual und Augmented Reality, digitaler Zwilling, Simulation und Mehrwertdienste. Durch Nutzung jeweils für die aktuelle Arbeitssituation passender Dienste, werden Mitarbeiter befähigt, sich an die steigende Individualisierung der Produkte und die flexiblere Produktion anzupassen. Kunden und Geschäftspartner werden durch die Möglichkeit eigene mehrwertgenerierende Dienste zu entwickeln und anderen Anwendern zur Verfügung zu stellen in den Wertschöpfungsprozess eingebunden. Diese Publikation beleuchtet die im Rahmen des Forschungsprojekts erarbeiteten Ergebnisse hinsichtlich für ein intelligentes Maschinenfenster benötigter Technologien und Entwicklungen.

Bibliographische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek

(German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie

(German National Bibliography); detailed bibliographic data is available via Internet at www.dnb.de.

© VDI Verlag GmbH · Düsseldorf 2018

Alle Rechte, auch das auszugswweisen Nachdruckes, der auszugswweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9406

ISBN 978-3-18-369702-1

Vorwort

Das Zukunftsprojekt *Industrie 4.0* der Bundesregierung hat die Schaffung neuer und Optimierung bestehender Formen intelligenter Produktions- und Automatisierungstechnik zum Ziel. Die Zukunft der Produktion wird durch eine starke Individualisierung der Produkte sowie eine hoch Flexibilisierung geprägt sein. Dazu wachsen die reale und die virtuelle Welt in der Fabrik von Morgen zusammen.¹ Wichtige Strukturelemente zukünftiger technischen Systemen sind innovative Benutzerschnittstellen, der digitale Schatten als virtuelles Echtzeitabbild der Realität sowie mehrwertgenerierende Softwaredienste. Auch eine App-Orientierung nach dem Prinzip von Smartphones und das Einbinden von Partnern und Kunden in den Wertschöpfungsprozess müssen adressiert werden.²

Das Verbundforschungsprojekt *iWindow: Intelligentes Maschinenfenster* beschäftigte sich mit der visuellen Unterstützung von Maschinenbedienern an Werkzeugmaschinen. Diese konnten bisher nur auf wenige bis keine Systeme, die sie bei ihren täglichen Aufgaben direkt an der Werkzeugmaschine unterstützen, zurückgreifen. Das Vorhaben wurde im Förderprogramm *IKT 2020 – Forschung für Innovationen* unter der Bekanntmachung *Virtuelle Techniken für die Fabrik der Zukunft - Ein Beitrag zum Zukunftsprojekt Industrie 4.0* des Bundesministeriums für Bildung und Forschung (BMBF) durchgeführt. Das Forschungsprojekt verbindet die reale und die virtuelle Welt in der Werkzeugmaschine durch Technologien wie Virtual und Augmented Reality, digitaler Zwilling, Simulation und Mehrwertdienste. Durch Nutzung jeweils für die aktuelle Arbeitssituation passender Dienste, werden Mitarbeiter befähigt sich an die steigende Individualisierung der Produkte und die flexiblere Produktion anzupassen. Kunden und Geschäftspartner werden durch die Möglichkeit eigene mehrwertgenerierende Dienste zu entwickeln und anderen Anwendern zur Verfügung zu stellen in den Wertschöpfungsprozess eingebunden. Diese Publikation beleuchtet die im Rahmen des Forschungsprojekts erarbeiteten Ergebnisse hinsichtlich für ein intelligentes Maschinenfenster benötigter Technologien und Entwicklungen.

Philipp Sommer

¹ Forschungsunion Wirtschaft und Wissenschaft (Hrsg.); acatech – Deutsche Akademie der Technikwissenschaften e.V. (Hrsg.): *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 : Deutschlands Zukunft als Produktionsstandort sichern*. Zwischenbericht des Arbeitskreises Industrie 4.0. Berlin, 2012

² Bauernhansl, Thomas ; Krüger, Jörg ; Reinhart, Gunther ; Schuh, Günther: *WGP-Standpunkt Industrie 4.0*. Berlin, 2016

Inhaltsverzeichnis

1	Motivation und Konzept	1
1.1	Motivation und Zielstellung.....	1
1.2	Konzept.....	6
2	Ergebnisse aus Demonstratorensicht	9
2.1	Einleitung	9
2.2	Kameraauswahl und -kalibrierung.....	10
2.3	Absortierassistent.....	12
2.4	Kamerabasierte Überlagerung an einem Roboter	14
2.5	Kamerabasierte Überlagerung an Werkzeugmaschinen.....	14
2.6	Perspektivische Überlagerung am transparenten LCD	16
2.7	Perspektivische Überlagerung am transparenten OLED Display	18
2.8	Finaler Demonstrator Werkzeugmaschine.....	20
2.8.1	Echtzeitvisualisierung im Bedienpanel.....	20
2.8.2	Perspektivische VR-Echtzeitvisualisierung an herkömmlichen Displays.....	20
2.8.3	Augmented Reality mittels transparenter Displays	22
3	Unterstützung beim Absortiervorgang in der Blechfertigung	25
3.1	Ausgangssituation und Zielsetzung	25
3.2	Absortierassistentz mittels kamerabasierter Entnahmedetektion	27
3.3	Erfassung des Absortierbereichs mit Kameras	28
3.4	Ermittlung der ebenen Bildszene auf Basis der Kamerabilder	29
3.5	Erfassung der Tafelgeometrie auf Basis des NC-Programms	31
3.6	Nachweis der Funktionsfähigkeit des Assistenzsystems	32
4	Dynamisch erweiterbares Framework für Mehrwertdienste.....	34
4.1	Motivation und Anforderungen	34
4.1.1	Anforderungen an die Gestaltung eines erweiterbaren Plug-In-Systems	37
4.1.2	Anforderungen an Sicherheit und Zugriffsrechte von Diensten	37
4.1.3	Anforderungen an Entwicklung und Vertrieb von Diensten.....	37
4.2	Mehrwertdienste.....	38
4.2.1	Basisdienste.....	38

4.2.2	Mehrwertdienste	39
4.2.3	Metadaten	40
4.2.4	Gestaltung unabhängiger Dienste	40
4.3	Kommunikation zwischen Diensten	42
4.4	Literaturverzeichnis	46
5	Datenmodell und Steuerungsanbindung über OPC UA	47
5.1	Einleitung	47
5.2	Ausgangslage und Problemstellung	47
5.3	Datenmodell	49
5.3.1	Grundlagen OPC UA Adressraum	49
5.3.2	Informationsmodelle für CNC-Steuerungen	50
5.3.3	OPC UA VDW Companion Specification	52
5.4	Steuerungsanbindung über OPC UA	54
5.4.1	OPC UA VDW Companion Specification – Evaluierung mit Prototypen	54
5.4.2	Echtzeitdaten über OPC UA	55
5.5	Ergebnisse und Ausblick	58
5.6	Literaturverzeichnis	59
6	Visualisierung des Maschineninnenraums mit VR & AR	61
6.1	Darstellungsarten	61
6.2	Virtuelles Modell des Maschineninnenraums	63
6.3	Synchronisierung mit aktuellen Maschinendaten	65
6.4	Kamerabasierte AR	67
6.4.1	Beschreibung der Kameraperspektive	67
6.4.2	Überlagerung des Kamerabilds mit virtuellen Inhalten	69
6.4.3	Verdeckungsrechnung	70
6.5	Berücksichtigung der Betrachterperspektive	71
6.5.1	Nachbildung des realen Displays durch ein virtuelles Display	72
6.5.2	Erfassung der Position des Betrachters mittels Head- bzw. Eye-Tracking	73
6.5.3	Berechnung der zur Darstellung benötigten perspektivischen Parameter	74
6.6	Literaturverzeichnis	76
7	Kalibrierung von Kameras in Werkzeugmaschinen	78

7.1	Einleitung und Zielsetzung.....	78
7.2	Grundlagen der Kamerakalibrierung.....	78
7.3	Anforderungen	86
7.3.1	Arbeitsräume der Werkzeugmaschinen	88
7.3.2	Bildaufnahme	89
7.3.3	Kalibrierungsprozess	90
7.4	Evaluierung der Methoden zur Kamerakalibrierung.....	90
7.4.1	Kalibrierverfahren in OpenCV	90
7.4.2	MATLAB Camera Calibration Toolkit	92
7.4.3	Kalibrierung von ToF-Kameras	93
7.4.4	Versuch der Kalibrierung auf Palette in Laserschneidmaschine (extrinsisch) ...	94
7.4.5	Versuch der Kalibrierung in der SW Maschine (extrinsisch)	96
7.5	Kalibrierung eines 1-Kamerasystems	97
7.6	Kalibrierung eines Multi-Kamerasystems.....	98
7.6.1	Vorgehen	100
7.6.2	Finden der LED Projektionen.....	101
7.6.3	Der MultiCamCalib Algorithmus	103
7.7	Literaturverzeichnis	104
8	Echtzeit-Maschinensimulation und der Nutzen von Mehrwertdiensten	106
8.1	Ziele.....	106
8.2	Echtzeit-Maschinensimulation.....	107
8.3	Überlagerungsberechnung	110
8.4	Kollisionsvermeidung.....	111
8.5	Mehrwertdienste.....	112
8.5.1	Bahnvisualisierung inklusive Überdeckungsberechnung.....	112
8.5.2	Werkstück- und Werkzeugkonturvisualisierung.....	112
8.5.3	Perspektivische Visualisierung	113
8.6	Ausblick.....	113
8.7	Literaturverzeichnis	114
9	Steuerungsgekoppelte Vorab-Überprüfung des Bearbeitungsprozesses	115
9.1	Steuerungsanbindung mittels OPC UA.....	116

9.2	Visualisierung.....	117
9.2.1	3D-Stereo.....	117
9.2.2	Perspektivische Darstellung als Fenster.....	117
9.2.3	Oberflächeneigenschaften	119
9.3	Modellierung und Simulation von Werkstücken, Werkzeugen und Magazinen	121
9.4	Technische Integration & Anwendung von Mehrwertinformationen	124
9.4.1	Abtragsimulation.....	124
9.4.2	Spurbildung	124
9.4.3	Kollisionserkennung	126
9.4.4	Schnelle Kommunikation.....	127
9.5	Literaturverzeichnis	128
10	3D-Rekonstruktion	129
10.1	Einleitung und Zielsetzung.....	129
10.2	Evaluierung der Methoden zur 3D-Rekonstruktion	130
10.2.1	Stereo-Kameras	130
10.2.2	ToF-Kameras.....	131
10.2.3	Structured Light Scanner.....	133
10.3	Framework für Computer Vision	135
10.4	3D-Rekonstruktion in der Werkzeugmaschine.....	138
10.5	Literaturverzeichnis	139

1 Motivation und Konzept

D. Do-Khac

1.1 Motivation und Zielstellung

Von der modernen industriellen Fertigung fordert der Weltmarkt eine weiter steigende Vielfalt an komplexer werdenden Produkten in immer kürzer werdenden Lieferfristen bei gleichzeitig steigender Forderung nach Flexibilität. Kurze Produktlebenszyklen bzw. Auffrischungs-Zyklen einer immer weiter wachsenden Produktvielfalt erhöhen die Erwartungshaltung der Märkte. Der globalisierte Wettbewerb stellt daher an die produzierenden Unternehmen höhere Anforderungen hinsichtlich ihrer Reaktionsschnelligkeit. Fertigungsbetriebe in Hochlohnländern sind umso stärker gefordert, da sie höheren Ansprüchen an Termintreue und gelieferter Qualität genügen müssen, was teils ihre Preisstellung rechtfertigt. Die Unternehmen sind gezwungen, die Arbeitsabläufe bei einem Auftragswechsel oder einem neuen Produktionsstart schnell und fehlerfrei zu beherrschen. Für den Kunden des fertigenden Unternehmens bestimmt die Lieferzeit oft maßgeblich die Markteintrittszeit, so dass der Produktlieferant sich oft keine Produktionsanlaufschleifen leisten kann. Aus Sicht eines Maschinenherstellers sind also verschiedene Möglichkeiten zu verfolgen, wie man den Maschinenbetreiber dabei unterstützen kann, die Produktwechsel und Anlaufphasen schneller und treffsicherer zu beherrschen. Gelingt es, mittels zusätzlicher Funktionen der Maschine dem Maschinenbetreiber in dem Ablauf bedeutende Mehrwerte zu bieten, so ist damit ein Alleinstellungsmerkmal, ein Wettbewerbsvorteil für die Werkzeugmaschine erzielt.

Bei Fertigung eines neuen Produktes oder eines Auftragswechsels sind unumgänglich manuell durchzuführende Schritte an der Werkzeugmaschine erforderlich. Die Werkzeugmaschine muss passend für einen Auftrag mit NC-Programmen versorgt werden, es muss ein passender Werkzeugsatz in der Maschine eingerichtet werden, und so muss beim Einfahren des ersten Teils der Maschinenbediener ein komplexes Spektrum an Aufgaben bewältigen, angedeutet in Abbildung 1-1. Der Großteil dieser Einrichte-Aufgaben wird trotz der vergangenen Fortschritte und anhaltenden Weiterentwicklung auch auf absehbare Zeit vom Bedienpersonal an der Maschine manuell durchgeführt werden. Angesichts des globalen Wettbewerbs und der steigenden Produktkomplexität muss das Bedienpersonal also komplexere Aufgaben in gleicher oder weniger Zeit bewältigen. Folgt man der Annahme, dass eine Vollautomatisierung der Maschinen-Einrichtung nicht eintreten wird, so gilt es, die Mitarbeiter an den Maschinen besser zu unterstützen.



Abbildung 1-1: Maschineneinrichter verantworten die korrekte Kombination von Fertigungsmitteln in der Maschine (Quelle: DMG MORI)

Es können verschiedene Aspekte betrachtet werden, um eine bessere Bediener-Unterstützung zu erzielen. Beim Rüsten einer Werkzeugmaschine für einen neuen Fertigungsauftrag sind einige Teilaufgaben zwar zunehmend digitalisiert, was jedoch auch dazu führt, dass verschiedene Computersysteme zu bedienen sind. Die Beschickung einer Maschine mit passenden NC-Programmen kann beispielsweise über ein Netzwerk erfolgen. Die Prüfung, ob das richtige Programm auch auf der Maschine angewählt ist, wird unter Nutzung eines Verwaltungssystems durchzuführen sein. Sofern die Einbringung der Werkzeugdatensätze in die Maschine digitalisiert ist, wird üblicherweise ein zusätzliches, separates Software-System eingesetzt, welches beispielsweise die Werkzeugdaten mit einem ERP-System synchronisiert, oder sich aus einer Datenbank speist, die wiederum Messprotokolle aus Werkzeug-Voreinstellgeräten verwaltet.

Programmanwahl und Werkzeugdatenbeschickung sind natürlich nur zwei von mehreren heute möglichen Beispielen, die ein eigenes Softwaresystem mitbringen. Und im Digitalisierungstrend der Industrie werden weiterhin zusätzliche neue Möglichkeiten angedacht und entwickelt, bislang nicht computergestützte Tätigkeiten zu digitalisieren. Momentan verstärkt dies einen Trend, Smartphones, Tablets, Industrie-PCs oder weitere Bildschirme im Umfeld von Werkzeugmaschinen einzusetzen. Jeder für sich versuchen diese Ansätze, den jeweils unterstützten Arbeitsablauf effizienter zu gestalten. Dies belastet das Personal gleichzeitig jedoch dadurch, dass die Aufmerksamkeit auf zusätzliche Bildschirme mit nicht unbedingt gleichbleibender Benutzerführung verteilt werden muss. Dieser Trend wird durch die aktuelle Verbreitung solcher Smart Devices (Tablets, Smartphones usw.) verstärkt, ist aber nicht grundsätzlich neu: Eine Restlaufzeit-Anzeige, Werkstückzähler, Simulationen und weitere Information darstellende Hilfestellungen erhöhen bereits heute die Anzahl der verteilten Anzeigeelemente, Beispiele zeigt Abbildung 1-2.



Abbildung 1-2: Zusätzliche Anzeigen als Informationskanäle zum Maschinenbediener (Quelle: DMG MORI)

Auch ohne konkret auf einen der vielen im Rahmen eines Produktionsanlaufs zu bewältigenden Schritte in der Maschinenvorbereitung einzugehen, ist ersichtlich, dass eine fortschreitende Digitalisierung hinsichtlich der Unterstützung des Personals in Bezug auf Anzeigen, Softwaresysteme und Bedienerschnittstellen konsolidiert werden muss. Ansonsten ist zu erwarten, dass die Effizienz-gewinne der digitalen Unterstützung teils wieder verloren gehen.

Ansätze zur digitalen Teilautomatisierung von Abläufen beziehungsweise der Ablaufunterstützung für das Bedienpersonal legen aktuell eine gewisse Abstraktion vom konkreten Arbeitsablauf zu Grunde. Typischerweise werden solche Softwaresysteme an Büro-PCs entwickelt und nehmen oft auch ein solches Nutzungsmodell an, lediglich mit eingeschränkten Bedienmöglichkeiten, wenn an der Maschine eine Computermouse nicht vorausgesetzt werden kann. Ist beispielsweise für einen Fertigungsauftrag das Werkstück definiert, im klassischen Vorgehen in Form einer Zeichnung, so ist das heute in einer digitalen Auftragsverwaltung eine dreidimensionale Darstellung des Geometriemodells des Werkstücks. Dies wird dann angezeigt, wenn eine abstrakte Folge von Menüs durchgegangen wird, die in Summe eine digitale Auftragsmappe repräsentieren. Während in Produkten verschiedener Verbrauchermärkte heute spezielle Brillen, Anzeigen oder Smartphone-Software den Einsatz von Augmented Reality ermöglicht, eine verschmolzene Darstellung der realen Welt mit digitalen Zusatzinformationen einer virtuellen Welt (Beispiel aus dem Automobilsektor zeigt Abbildung 1-3), so sind solche Ansätze in Produktions- und Fertigungsmaschinen aktuell die Ausnahme. Der Vorteil, digitale Zusatzinformationen und von Computern vorausgewertete Zusammenhänge in direkter Verbindung mit den realen Gegebenheiten darzustellen, kann im Allgemeinen nicht genutzt werden.

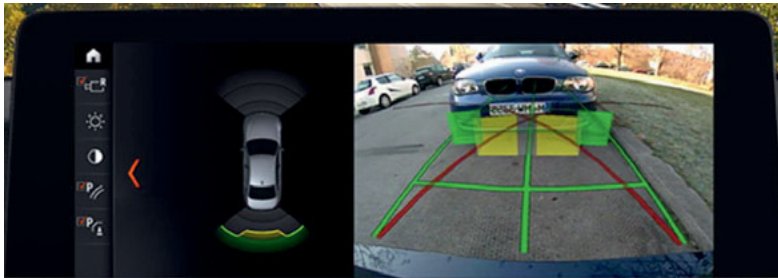


Abbildung 1-3: Digitale Zusatzinformation wird in die Ansicht der realen Welt eingeblendet (Quelle: Homepage der BMW Group)

Dies zeigt eine weitere Begrenzung der Möglichkeiten zur digitalen Unterstützung des Rüstpersonals auf. Während einerseits verschiedene Softwaresysteme an verschiedenen Zugangspunkten (also Bildschirmen mit Eingabemöglichkeiten) in wachsender Zahl als Nebeneffekt die Bedienkomplexität erhöhen, können neue Technologien zur direkteren Verknüpfungen zwischen Arbeitsgang und Unterstützungssystem heute nicht in praxistauglicher Breite eingesetzt werden. Dabei wäre dies ein vielversprechender Weg, durch eine Verringerung des Abstraktionsbedarfs beim Bediener diesen zu entlasten. Muss der Bediener nicht mehr durch menügeführte Bedienmasken navigieren, sondern bekommt eine direkte Darstellung in Augmented Reality Technik, so ist in der Softwarebedienung und im kognitiven Erfassen der Information aus der digitalen Welt eine höhere Effizienz erreicht.

Neben den Aspekten einer Unterstützung durch digitale Systeme im Sinne von zusätzlichen Informationskanälen zum Bediener hin, sind durch ein digitales Sichtfenster in die Maschine konkret an die Fertigungssituation im Arbeitsraum gekoppelte Vorteile denkbar. Während eine Simulation des Bearbeitungsvorgangs in der Zerspanung typischerweise zu einer Validierung des programmierten Zerspanungsprozesses führen soll, ist durch den Einsatz eines digitalen Maschinenfensters auch zum Zeitpunkt der tatsächlichen Bearbeitung ein Mehrwert erkennbar. Beim Einfahren eines neuen Bearbeitungsprozesses ist die Einsehbarkeit in den Maschinenarbeitsraum ein wichtiges Kriterium für den Bediener. Um einen guten Überblick über die Bearbeitungssituation zu gewähren, werden durchaus möglichst große Sichtfenster verbaut. Während des laufenden Prozesses ist jedoch nicht ungewöhnlich, dass durch Kühlschmiermittel die Sichtbarkeit des Werkstücks beeinträchtigt wird. Gerade bei Bearbeitungen mit hohen Drücken, durchaus über 10 Bar, ist die Sichtbarkeit des direkten Bearbeitungsprozesses am Werkzeug stark beeinträchtigt durch Spritzen des Kühlschmiermittels oder Nebelbildung. Selbst in der Trockenbearbeitung kommt es bei längeren Prozessen durch die Bildung von Spänehaufen zu beeinträchtigter Sichtbarkeit an der Werkzeugschneide. Ein digitales Sichtfenster mit der Möglichkeit, einen simulierten Ablauf darzustellen, bietet hier die Möglichkeit eine dreidimensionale Sicht des Prozesses

darzustellen. Dies könnte dann in Echtzeit, unbeeinträchtigt von Kühlschmiermittel oder Spanbildung die Sichtbarkeit des Prozesses verbessern.

Entsprechende Echtzeitfähigkeit vorausgesetzt, ist sogar eine weitere Einsatzmöglichkeit denkbar. So ist es bei Maschinen für die integrierte Fräs-Drehbearbeitung kostenintensiv, im Sinne der Arbeitssicherheit taugliche Sichtscheiben zu verbauen. Diese Maschinen zeichnen sich dadurch aus, dass auf einem Fräszentrum auch Drehbearbeitungen durchgeführt werden können. Daraus ergeben sich veränderte Sicherheitsbetrachtungen, die durchaus den Einsatz von Verbundsicherheitsglas zur Folge haben. Unter diesen Voraussetzungen ist denkbar, hier mit einer Blechumhausung zu arbeiten und auf dem digitalen Sichtfenster stattdessen eine simulierte Darstellung des gesamten Arbeitsraumes zu nutzen.

Als letzter hier adressierter Aspekt wird die zunehmende Angebotsvielfalt digitaler Softwareprodukte aufgezeigt. Der grundsätzliche Einsatz von Software ist in der spanenden Fertigung natürlich lange etabliert, schließlich wird seit über dreißig Jahren auf NC-gesteuerten Maschinen gefertigt, die ohne Programmierung kein Werkstück fertigen. Auch im weiteren Sinne als Unterstützungssysteme charakterisierbare Software, wie CAD/CAM-Systeme welche diese NC-Programmierung automatisieren sollen, sind seit über zwanzig Jahren verfügbar und im Einsatz. Dieser Software-Markt ist jedoch ursprünglich stark geprägt von einem Nischen-Charakter und proprietären Grundlagen. So zielt ein System zur Maschinendatenerfassung (MDE) oder ein CAM-System typischerweise auf eine bestimmte Maschinensteuerung, oder oft auch spezifisch auf einen Maschinenhersteller, wenn nicht sogar eine einzelne Baureihe eines Maschinenherstellers.

Mit dem Aufkommen von Smartphones und den mit ihnen auf sehr breiter Front etablierten Plattformen beobachtet man heute einen Trend, ein solches Entwicklungs-, Verteilungs- und Vermarktungskonzept auch im Bereich der Software für die Fertigung zur Anwendung zu bringen. Die von so einem Vorgehen erwarteten Vorteile sollen unter anderem dem Applikations-Entwickler eine breitere Verteilung seines Produkts ermöglichen, so dass sich seine Entwicklungsaufwände besser amortisieren. Folge wären ein größeres Angebot an Applikationen für einen größeren Interessenten-Markt, sowie eine geringere Einstiegshürde, so dass auch Anwendungen mit kleineren Mehrwerten eine Implementierung rechtfertigen können. Aus Sicht der Anlagenbetreiber hätte dies die positive Konsequenz, dass sie an einer besser gebündelten Stelle ein größeres Angebot an Software-Systemen zur Auswahl haben und sie können mit weniger Aufwand einen Überblick über die Möglichkeiten bekommen. Auch dieser Plattform-Ansatz lässt sich im aktuellen Fertigungsumfeld kaum Verfolgen.

1.2 Konzept

Um die in Kapitel 1.1 aufgezeigten Themen einer neuartigen Lösung zuzuführen verfolgt das Projekt einen Lösungsdemonstrator, auf den in Kapitel 2 im Detail eingegangen wird, der mehrere Lösungsansätze umsetzen und einer Bewertung zuführen soll. Dreh- und Angelpunkt ist das Lösungskonzept eines intelligenten Maschinenfensters. Spanende Fertigungsmaschinen verfügen ausnahmslos über Sichtfenster, die als trennende Einrichtung im Sinne des Personenschutzes im Maschinenbetrieb dienen, und gleichzeitig die Überwachung des Fertigungsprozesses in der Maschine ermöglichen. Dieses Sichtfenster mit einem digitalen, durchsichtigen Bildschirm zu versehen ermöglicht im ersten Kernaspekt eine Konsolidierung der Bedienschnittstellen für den Maschinenbediener. Bei Betrachtung einer typischen, nicht besonders hohen Ausprägung digitaler Unterstützungsprozesse muss der Maschinenbediener den Maschinenarbeitsraum beobachten, im Vorfeld eine Simulation am Büro-PC sichten, die Maschinenfunktionen über ein Human Machine Interface (HMI) bedienen und digital vorliegende Auftragsverwaltungsdaten auf einem Tablet-PC gegenprüfen. Diese vier digitalen Zugangspunkte lassen sich deutlich konsolidieren, wenn das intelligente Maschinenfenster genutzt wird – der Nutzen ist veranschaulicht in Abbildung 1-4, die detaillierte Kombination mehrerer Informationsebenen untersuchen Kapitel 4 und 6.



Abbildung 1-4: Das intelligente Maschinenfenster konsolidiert Informationskanäle zum Bediener

1: Simulation 2: HMI 3 Maschineninnenraum 4: Smart Device

A: Maschineninnenraum inkl. HMI, Simulation B: Smart Device

(Quelle: ISW)

Die Nutzbarmachung des Maschinensichtfensters als digitaler Bildschirm erlaubt, die spezifisch an der Maschine und in der Fertigungssituation benötigten digitalen Systeme auf dem Maschinensichtfenster gebündelt anzuzeigen. Eine Synchronisierung mit dem aktuellen Ist-Zustand der Steuerung stellt Kapitel 5 näher dar. Davon eher unabhängige

Softwaresysteme, wie beispielsweise eine Auftragsverwaltung, kann auf einem mitgeführten Smart Device (Tablet-PC, Smartphone o.ä.) verbleiben, oder sogar mit den Anzeigen im intelligenten Maschinenfenster verzahnt werden.

Neben dem bloßen Nutzen als zusätzliche digitale Darstellung und Eingabeschnittstelle soll ferner die kognitiv effizientere, verschmolzene Darstellung digitaler und realer Ansichten durch Augmented Reality umgesetzt werden. Hier gibt es diverse Wege, dem Bediener den Abgleich beispielsweise von Auftragsdaten zur erreichten Rüstsituation oder dem realen Fertigungsprozess zu erleichtern. Wenn hier simulierte Prozesse dem realen Prozess überlagert dargestellt werden, dann ist beispielsweise auf einen Blick und ohne langes Überlegen prüfbar, ob das gerade eingesetzte Werkzeug auch dem geplanten Werkzeug entspricht. Notwendig dazu ist die Kombination einer digitalen Simulation mit der Darstellung auf dem intelligenten Maschinenfenster, wobei die der Maschinensteuerung vorliegenden Informationen zum Ist-Zustand der Maschine ebenfalls in Echtzeit ausgewertet werden können. Veranschaulicht ist dieser Ansatz in Abbildung 1-5, zur Umsetzung benötigte Technologien zur Bilderfassung und Visualisierung sind beschrieben in Kapitel 6 und Kapitel 7.

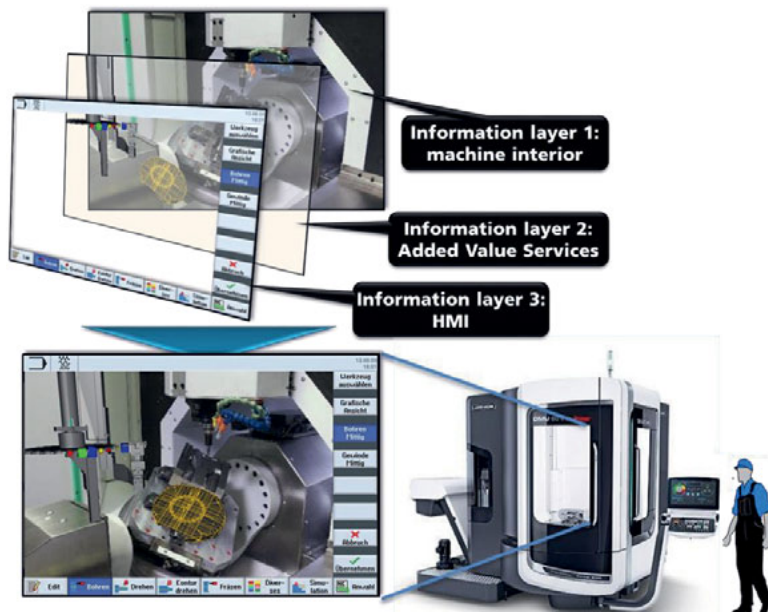


Abbildung 1-5: Das intelligente Maschinenfenster kombiniert mehrere Informations-ebenen (Quelle: ISW)

Die Mehrwerte einer separaten Simulation setzen heutzutage aber allgemein voraus, dass zumindest entsprechende Softwareschnittstellen verfügbar sind, wenn nicht gleich eine ganze

Plattform zur Verzahnung verschiedener Softwarekomponenten zur Verfügung steht. So liefern moderne Maschinensteuerungen zwar ausnahmslos Simulationssysteme mit. Eine zweckgebundene Anpassung, wie sie für die in Abbildung 1-5 angedeutete Möglichkeit einer Darstellung in Augmented Reality notwendig wäre, ist aber bei fast allen diesen Systemen nicht hinreichend weitgehend möglich. Dazu muss in Soft- und Hardware eine entsprechende Infrastruktur konzipiert und umgesetzt werden. Ein Framework für mehrere Mehrwertdienste erarbeitet Kapitel 4. Darauf aufsetzende digitale Mehrwertdienste beschreiben die Kapitel 8 und 9. Den größten Mehrwert hinsichtlich Nutzen- und Erkenntnisgewinn erbringt die Integration der verschiedenen Projekt-Teilaspekte in diesen Lösungsansatz.

Die konzipierten Lösungsansätze werden vornehmlich bei Produktionsbeobachtung, Rüstvorgängen und dem Einfahren neuer Werkstücke einen Mehrwert bringen. Folgerichtig fokussiert das Projekt für den Projektdemonstrator auf eine Maschine, die mit häufigen Produktwechseln und der Fertigung kleiner Losgrößen oft konfrontiert wird. Dies war häufig bei Maschinen der heute von CMX-Baureihen abgelösten DMG MORI ECOLINE der Fall, so dass im Projekt auf einer ecoMill 50 aufgebaut wird – die etwas größere Schwestermaschine ecoMill 70 zeigt Abbildung 1-6.



Abbildung 1-6: Fertigungszentren der Baureihen aus dem Produktprogramm ECOLINE finden sich in Fertigungsbetrieben mit kleinen Losgrößen und häufigen Produktwechseln (Quelle: DMG MORI)

Ziel des Projekts ist aufbauend auf diesen Randbedingungen die in den folgenden Kapiteln berichtete Definition und Integration der notwendigen Teiltechnologien in einen Gesamtdemonstrator, an dem sich Herausforderungen, Machbarkeit und Möglichkeiten zur Überführung in die industrielle Praxis beurteilen lassen.

2 Ergebnisse aus Demonstratorensicht

P. Sommer, M. Kiefer, R. Rahäuser, W. Schotte, A. Neyrinck, S. Müller, E. Berckmann

2.1 Einleitung

Auf Grund der visuellen Ausrichtung des Projektes wurde von Anfang an großer Wert auf das subjektive Erleben und die praktische Erfahrung mit den verwendeten Technologien gelegt. Durch dieses Ziel motiviert entstanden über die Projektlaufzeit hinweg mehrere Demonstratoren, die kontinuierlich weiterentwickelt wurden. Ein Teil dieser Demonstratoren wurde im Rahmen mehrerer Messeauftritte auch zur Präsentation von Projektergebnissen gegenüber einem breiten Fachpublikum genutzt. In den nachfolgenden Kapiteln werden die entstandenen Demonstratoren insbesondere anhand von Bildern ausführlicher erläutert. Dabei werden jeweils die im Rahmen dieser Demonstratoren (weiter-)entwickelten und eingesetzten Technologien angeführt und zur näheren Erläuterung der technischen Details auf weitere Kapitel dieser Publikation verwiesen.

Abbildung 2-1 stellt die Entwicklung der Demonstratoren ausgehend von der Projektidee hin zum finalen Demonstrator und zum bei der Firma TRUMPF entstandenen Produkt Absortierassistent dar. Grundlage für alle kamerabasierten Verfahren zur Überlagerung wie auch zur Objekterkennung und -rekonstruktion bilden die Auswahl und die Kalibrierung von Kameras. Hierzu wurde der in Kapitel 2.2 beschriebene Demonstrator beim Projektpartner HLRS entwickelt. Die aus diesem Demonstrator gewonnenen Erkenntnisse flossen direkt in die Entwicklung des Produktes Absortierassistent, dargestellt in Kapitel 2.3, ein. Bereits in einer frühen Projektphase wurden erste Erkenntnisse zur kamerabasierten Überlagerung einer industriellen Umgebung mit virtuellen Objekten gesammelt. Hierzu wurde ein Szenario aus realem Industrieroboter und virtuellen Materialflusseinrichtungen aufgebaut. Der entstandene Demonstrator wird in Kapitel 0 näher erläutert. Ausgehend von diesem Demonstrator und den Erkenntnissen aus Kameraauswahl und -kalibrierung wurden die Konzepte zur kamerabasierten Überlagerung auf eine kommerzielle Werkzeugmaschine und die zugehörige Benutzerschnittstelle (engl. human machine interface, HMI) übertragen, siehe Kapitel 2.5. Parallel zu den Entwicklungen zur kamerabasierten Überlagerung wurde auch die perspektivische Überlagerung der Realität mit virtuellen Inhalten an transparenten Displays untersucht. Hierzu wurden im ersten Schritt, bedingt durch die eingeschränkte Verfügbarkeit entsprechender alternativer Systeme, Untersuchungen mit einer transparenten Flüssigkristallanzeige (engl. liquid crystal display, LCD) durchgeführt. Der hierzu genutzte Aufbau wird in Kapitel 2.6 erläutert. Im weiten Projektverlauf konnte ein auf organischen Leuchtdioden (engl. organic light emitting diode, OLED) basierendes Display beschafft werden. Die erhofften besseren optischen Eigenschaften gegenüber einem LCD bestätigten sich anhand des in Kapitel 2.7 dargestellten Demonstrators. Abschließend wurden alle gewonnenen Erkenntnisse in Form eines finalen Demonstrators an einer Werkzeugmaschine

zusammengefasst. Hierbei wird nochmals untergliedert in die Visualisierung mit kamerabasierter Überlagerung des Maschineninnenraums am Bedienpanel (Kapitel 2.8.1), die perspektivische Visualisierung mittels virtueller Realität (engl. Virtual Reality, VR) an herkömmlichen Displays (Kapitel 2.8.2), sowie die Erweiterung des realen Maschineninnenraums durch virtuelle Inhalte (engl. Augmented Reality, AR) mittels transparenter Displays (Kapitel 2.8.3).

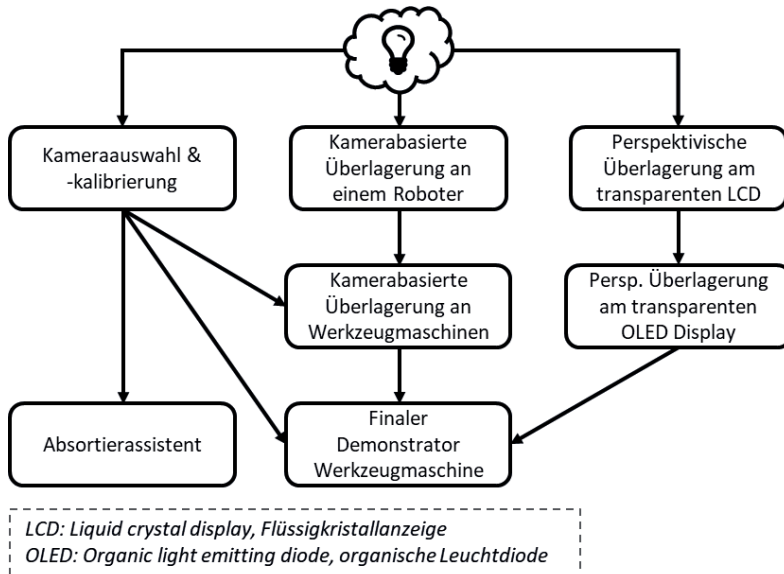


Abbildung 2-1: Übersicht über die Entwicklung der Demonstratoren (Quelle: ISW)

2.2 Kameraauswahl und -kalibrierung

Im Rahmen des Projekts wurde, wie in Kapitel 2.1 erwähnt, ein Prototyp aufgebaut werden, der als Entwicklungs- und Testumgebung für die Hardwareevaluierung und Softwareentwicklung für Visualisierung, Nutzer-Tracking und 3D-Rekonstruktion verwendet werden soll. Komponenten, die zur Auswahl stehen, sollten zunächst am Prototyp verbaut und evaluiert werden, bevor diese oder baugleiche Hardware im Demonstrator zum Einsatz kommt.

Im wesentlich ermöglicht der Prototyp drei Szenarien, die Visualisierung mit einem opaken Display (Kapitel 9), die Visualisierung mit einem transparenten Display inklusive des Nutzer-Trackings (Kapitel 6.5) sowie der Aufbau eines Systems zu Kamera-basierten 3D-Rekonstruktion (Kapitel 10) des Maschineninnenraums. Hierfür wurde aus Aluminiumprofilen ein Gestell aufgebaut, das das Setup unterschiedlicher Kamerakonfigurationen ermöglicht, die die unterschiedlichen Voraussetzungen in den Arbeitsmaschinen, die im Rahmen von

iWindow betrachtet werden, möglichst ähnlich abbildet. Das heißt beispielsweise, dass Kameras in Werkzeugmaschinen und entsprechend am Prototypen so angebracht werden müssen, dass sie außerhalb des Arbeitsraums der Werkzeugmaschine liegen, um von Kühlmitteln, Spänen oder Funken möglichst nicht getroffen bzw. beschädigt zu werden. Der Prototyp gibt einem somit die Möglichkeit unter anderem verschiedene Anordnungen von Kameras oder verschiedene Kombinationen von Kameratypen zu testen (vgl. Abbildung 2-2 und Abbildung 2-3).

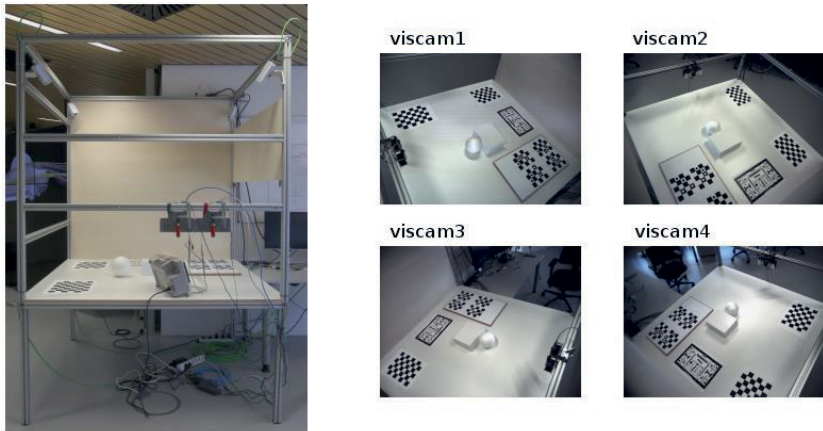
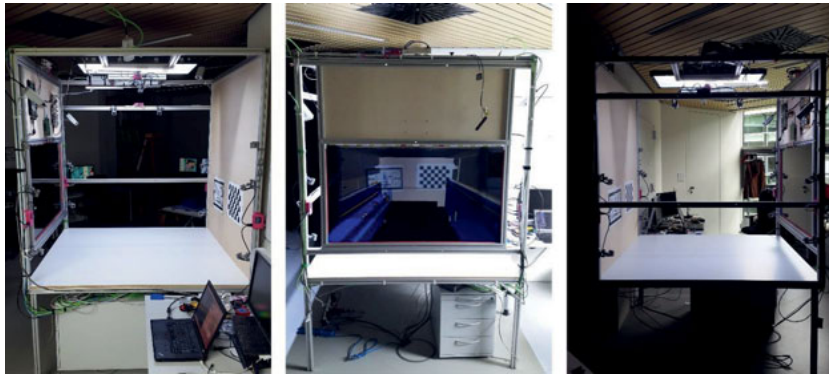


Abbildung 2-2: Erster Aufbau des Prototypen (Quelle: HLRS)

Im Laufe des Projekts wurden zahlreiche Displays, Kamerasysteme und Kamerakonfigurationen sowie Zubehör getestet, wie unter anderem einzelne und synchronisierte Time-of-Flight Kameras, IP-Kameras, verschiedene Kameramodule mit Rolling als auch Global Shuttern an Einplatinenrechnern, Stereokamerasystem, Beleuchtungsmodule, opake und transparente Displays.

Zum Ende des Projekts ist am Prototyp ein Kamerasystem bestehend aus 18 Kameras verbaut, das zur Entwicklung und Evaluation von Verfahren zur 3D-Rekonstruktion genutzt wurde. Dies sind 4 IP-Kameras, 2 Kamera paare in Stereokamerakonfiguration mit Rolling Shutter, 1 Kamera paar in Stereokamerakonfiguration mit Global Shutter und der Möglichkeit zur Aufnahme mit hoher Bildwiederholrate, sowie 8 weitere Kameramodule. Während die IP-Kameras direkt über Gigabit-Ethernet ausgelesen werden können, werden die 14 Kameramodule über USB 3.0 an sieben Hardkernel Odroid XU4 Einplatinencomputern betrieben. Die Odroid XU4 übernehmen die Bildaufnahme sowie eine Bildvorverarbeitung lokal und senden die vorverarbeiteten Bilder ebenfalls über Gigabit-Ethernet an den Host-Computer, der die weitere Bildverarbeitung und Kalibrierung der Kameras (Kapitel 7) sowie die 3D-Rekonstruktion übernimmt (Kapitel 10).



**Abbildung 2-3: Finaler Stand des Prototypen mit transparentem LC-Display
(Quelle: HLRS)**

Im Laufe des Projekts gab es wesentliche Weiterentwicklungen der Technologien im Bereich der transparenten Displays. Mit Projektbeginn war die Beschaffung von transparenten Displays auf LC-Basis geplant. Diese Technik wurde durch einige Hersteller abgekündigt und durch OLED Technologie ersetzt. Diese Displays haben eine deutliche höhere Transparenz und benötigen keine zusätzliche Hintergrundbeleuchtung. Transparente OLED-Displays bedeuten allerdings eine wesentlich teurere Beschaffung. Um am Demonstrator den Stand-der-Technik und damit ein OLED-Display installieren zu können, aber gleichzeitig auch die Entwicklung am Prototypen zu ermöglichen, wurde am Prototyp ein LC-Display verbaut, welches durch das HLRS zu einem transparenten Display umgebaut wurde. Das Bild dieses Umbaus ist zu den auf dem Markt verfügbaren transparenten Displays auf LC-Basis nochmal schlechter aber zumindest vergleichbar. Das Display stellt aber für die Entwicklung der Anwendungen unter Nutzung transparenter Displays eine gute Grundlage dar und konnte erfolgreich genutzt werden.

2.3 Absortierassistent

Beim Laserschneiden von Blechteilen werden aus einem Blech meist viele unterschiedliche Blechteile geschnitten. Der von TRUMPF entwickelte Absortierassistent unterstützt den Maschinenbediener beim anschließenden Absortieren der geschnittenen Blechteile. Hierzu wird der Bereich mit den geschnittenen Blechteilen mit mehreren Kameras beobachtet. Mithilfe einer geeigneten Kamerakalibrierung, einer Bildtransformation und einem Zusammenführen der einzelnen Bilder wird eine transformierte Bildszene des Blechs mit den geschnittenen Blechteilen erzeugt, welche mit der bekannten Sollgeometrie der Blechteile verglichen werden kann. Die Entnahme eines Blechteils wird mithilfe eines Differenzbildverfahrens und dem Abgleich mit der bekannten Sollgeometrie der Teile erkannt.

Die detaillierte Vorstellung des entwickelten Assistenzsystems erfolgt in Kapitel 3. Daneben wird in Kapitel 7 auf die Kamerakalibrierung eingegangen.

Die erste Vorstellung des kamerabasierten Assistenzsystems erfolgte auf der Fachmesse EuroBlech 2016 (vgl. Abbildung 2-4). Im Rahmen des BMBF-Verbundprojekts iWindow flossen Arbeitsergebnisse aus dem Konsortium mit ein. Die Studie demonstriert neue Lösungsansätze, um die Herausforderungen des Absortierens von Blechteilen an einer Laserschneidanlage besser zu beherrschen. Im Rahmen der Eröffnung des neuen TRUMPF Showrooms – Smart Factory Chicago – wurde die Weiterentwicklung des Demonstrators präsentiert (vgl. Abbildung 2-5). Der Showroom macht die Vernetzung und Digitalisierung der Blechverarbeitung live erlebbar. Hierzu wurde das Assistenzsystem zum Absortieren erstmals in einer seriennahen Umsetzung mit drei Kameras vorgestellt. Dieser Demonstrator basiert auf den grundlegenden Arbeiten der Studie und deren Weiterentwicklung hinsichtlich Arbeitsraumdimensionen und Praxistauglichkeit im Produktionsalltag.



Abbildung 2-4: Auf der EuroBLECH 2016 gezeigte Studie des Absortierassistenten mit einer Kamera. (Quelle: TRUMPF)



Abbildung 2-5: Im TRUMPF Showroom *Smart Factory Chicago* aufgestellter Absortierassistent mit drei Kameras und schräger Perspektive. (Quelle: TRUMPF)

2.4 Kamerabasierte Überlagerung an einem Roboter

Der erste Demonstrator zur kamerabasierten Erweiterung der Realität mit virtuellen Objekten basiert auf einem kleinen industriellen Roboter. Dieser Roboter agiert, wie im linken Bild in Abbildung 2-6 dargestellt, auf einer freien Fläche ohne weitere reale Objekte. Der Videostream einer diesen Roboter beobachtenden Kamera wird in Echtzeit auf ein Display übertragen und, wie im rechten Bild in Abbildung 2-6 zu sehen, mit Materialflusselementen erweitert. Bei diesen ausschließlich virtuell vorhandenen Materialflusselementen handelt es sich um den Greifer des Roboters, die Förderbänder und die Werkstücke. Um eine korrekte Darstellung der virtuellen Objekte in der realen Umgebung zu erreichen, werden die Perspektive sowie die optischen Eigenschaften der Kamera vermessen und nachgebildet (vgl. Kapitel 7 und 6.4.1). Die virtuellen Objekte werden dreidimensional in einem Computermodell (vgl. Kapitel 6.2) vorgehalten und durch Überlagerung in den Videostream integriert (vgl. Kapitel 6.4.2). Der reale Roboter wird über eine kommerzielle Robotersteuerung angesteuert und transportiert virtuelle Werkstücke von einem auf ein anderes Förderband. Im Computermodell wird synchron ein digitales Abbild des realen Roboters mitgeführt (vgl. Kapitel 6.3). Mittels einer bidirektionalen Datenanbindung werden die aktuellen Achswinkel des Roboters in die virtuelle Umgebung und Sensorsignale einer virtuellen Lichtschranke aus diesem Computermodell an den Roboter gespiegelt. Durch Verwendung des digitalen Abbilds des realen Roboters wird die Verdeckung virtueller Objekte durch den Roboter und umgekehrt korrekt berechnet und dargestellt (vgl. Kapitel 6.4.3).

Dieser Demonstrator wurde von den Projektpartnern ISW und ISG auf der Fachmesse SPS/IPC/Drives 2015 in Nürnberg ausgestellt.

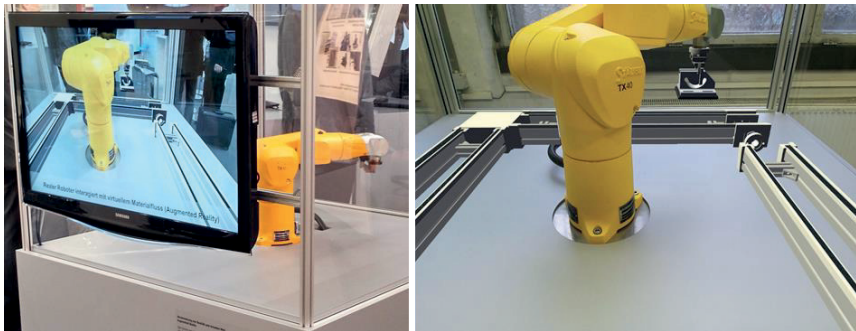


Abbildung 2-6: Kamerabasierte Überlagerung an einem Roboter; Links: Seitenansicht; Rechts: Screenshot der Überlagerung am Display (Quelle: ISW)

2.5 Kamerabasierte Überlagerung an Werkzeugmaschinen

Der zweite Demonstrator zur kamerabasierten Überlagerung von virtuellen Objekten basiert auf einem 5-Achs-Bearbeitungszentrum der Firma Schwäbische Werkzeugmaschinen GmbH

(SW). Diese wurde mit der in Abbildung 2-7 gezeigten Multitouch-Benutzeroberfläche (HMI) ausgestattet. Dabei handelt es sich um eine Neuentwicklung der Firma Bosch Rexroth, die neben den klassischen Aufgaben eines Bedienpanels an einer Werkzeugmaschine auch Mehrwertdienste wie eine NC-Simulation beinhaltet. Diese NC-Simulation simuliert virtuell den Bearbeitungsprozess parallel zur realen Bearbeitung der Werkzeugmaschine (vgl. Kapitel 8.2). Idee war es nun diese NC-Simulation als Erweiterung der Realität über einen Videostream zu legen und damit eine Alternative zu einem reinen Maschinenfenster zu schaffen. Um dies zu erreichen, wurde eine Kamera innerhalb des Maschinenraums verbaut. Für die korrekte Darstellung der virtuellen NC-Simulation innerhalb der vom Kamerabild erfassten realen Umgebung müssen die Perspektive und die optischen Eigenschaften der Kamera vermessen und entsprechend in der virtuellen Umgebung der NC Simulation angepasst werden (vgl. Kapitel 6.4, 7 und 8.3). Die NC-Simulation und Überlagerungsberechnung wurde von der Firma ModuleWorks entwickelt und in die HMI von Bosch Rexroth integriert. Damit die virtuelle NC-Simulation das Kamerabild nicht komplett überlagert, verfügt die NC-Simulation über die Möglichkeit der Anpassung der Transparenz einzelner virtueller Objekte. Abbildung 2-8 zeigt als Ergebnis dieser Arbeiten die Integration der kamerabasierend überlagerten NC-Simulation in die Multitouch-Benutzeroberfläche von Bosch Rexroth. Die Anbindung der NC-Simulation an die Daten der CNC-Steuerung erfolgt über die Kommunikationsschnittstelle OPC UA. Dazu bietet das CNC-System MTX von Bosch Rexroth eine spezielle Funktion zum Übertragen von Daten aus dem Echtzeitbereich des Steuerungskerns (vgl. Kapitel 5). Diese erlaubt die Ausführung der gezeigten Anwendung auf einem mobilen Endgerät wie einem Tablet (Abbildung 2-9). Hierbei erfolgt die Netzwerkverbindung des mobilen Endgeräts mit der Werkzeugmaschine und Kamera über eine drahtlose Funkverbindung per WLAN.

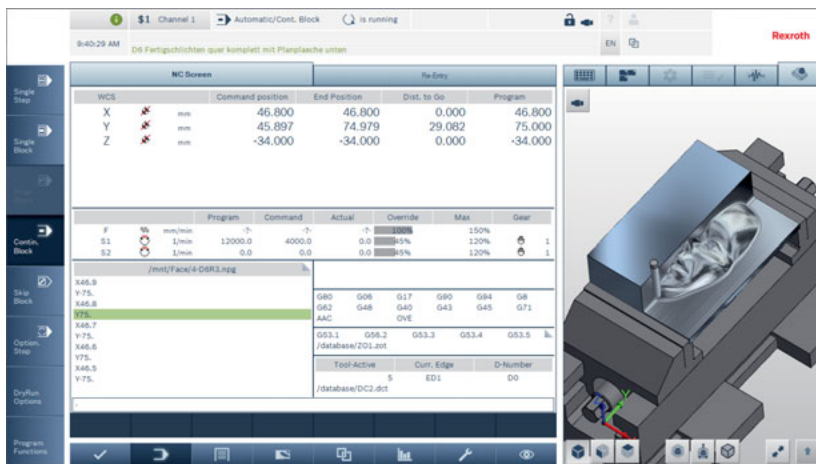


Abbildung 2-7: Rexroth IndraWorks Operation: HMI auf Multitouch-Bildschirm mit Bedienpanel als klassische HMI (Quelle: Bosch Rexroth AG)



Abbildung 2-8: Überlagerung von NC-Simulation und Kamerabild integriert in die Benutzeroberfläche IndraWorks Operation (Quelle: Bosch Rexroth AG)

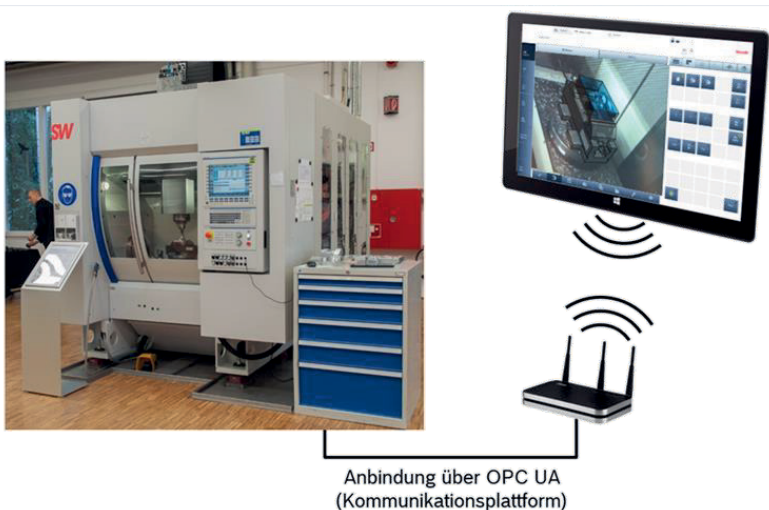


Abbildung 2-9: Multitouch-HMI mit kamerabasierter NC-Simulation auf einem mobilen Endgerät verbunden über WLAN und OPC UA (Quelle: Bosch Rexroth AG)

2.6 Perspektivische Überlagerung am transparenten LCD

Eine perspektivische Überlagerung des Maschineninnenraums durch Nutzung transparenter Displays ist die dritte Darstellungsart neben der reinen Visualisierung mittels Computermodell

und der kamerabasierten Überlagerung des Innenraums (vgl. Kapitel 6.1). Grundsätzlich kommen zwei Technologien für transparente Displays in Frage, Flüssigkristallanzeigen (engl. liquid crystal display, LCD) sowie auf organischen Leuchtdioden (engl. organic light emitting diode, OLED) basierende Displays (vgl. Kapitel 6.1). Da zu Beginn des Projekts transparente OLED Displays nur eingeschränkt verfügbar waren, wurden die ersten Versuche mit einem LCD durchgeführt. Ein herkömmliches LCD wurde zerlegt und zu einem transparenten Display umgebaut. Dazu wurden die rückwändige Verkleidung, die Elektronik und die zur Beleuchtung notwendige Hintergrundbeleuchtung (engl. Backlight) von der Flüssigkristallschicht abgenommen. Die Elektronik und die Flüssigkristallschicht wurden anschließend in einem neuen Gehäuse nebeneinander montiert und bilden das transparente LCD. Dieses wurde, wie in Abbildung 2-10 dargestellt, vor dem vorhandenen Fenster der Werkzeugmaschine positioniert. Das Backlight wurde im Maschineninnenraum an der, dem Fenster gegenüberliegenden, Wand platziert. Um die Perspektive des Betrachters berücksichtigen zu können, wurde oberhalb des Displays ein Tracking System angebracht (vgl. Kapitel 6.5).

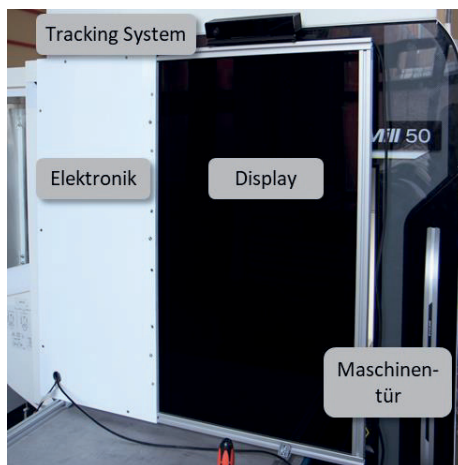


Abbildung 2-10: Aufbau mit transparentem LCD vor dem Maschinenfenster (Quelle: ISW)

Durch Nachbildung des Maschineninnenraums durch ein Computermodell (vgl. Kapitel 6.2) sowie Synchronisierung dieses Modells mit aktuellen Maschinendaten (vgl. Kapitel 6.3) wurde ein digitales Abbild des Innenraums geschaffen. Mittels perspektivischer Überlagerung können virtuelle Inhalte aus dem Computermodell in den realen Innenraum eingeblendet werden. Abbildung 2-11 zeigt Ergebnisse dieser perspektivischen Überlagerung aus Sicht des Betrachters aus drei unterschiedlichen Positionen (rechts, mittig, links zur Displaymitte). Als weiße Fläche deutlich zu sehen ist das Backlight im mittleren Bereich der Maschine. Eine Überlagerung zwischen realer und virtueller Maschine findet sich vor dem Backlight an der Spindel. Die Spindel selbst ist sowohl real wie auch virtuell vorhanden und daher doppelt zu

erkennen. Da die Geometrie der Spindel im Computermodell rein zylindrisch und nicht wie an der echten Maschine konisch geformt ist, kommt es zu einer kleinen Abweichung in der Überlagerung. In die reale Maschine war zur Zeit der Aufnahme kein Werkzeug eingespannt, dieses wurde rein virtuell integriert. Auf Grund der Synchronisierung des Modells mit der echten Maschine befindet sich das Werkzeug jedoch zu jeder Zeit an der korrekten Position. Deutlich zu erkennen sind auch die durch das Display entstehende Unschärfe des realen Innenraums sowie eine Verdunklung.

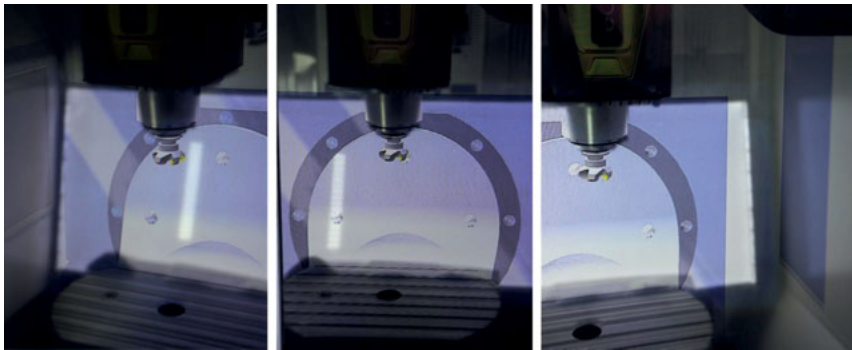


Abbildung 2-11: Perspektivische Überlagerung mit transparentem LCD aus drei Perspektiven (Quelle: ISW)

2.7 Perspektivische Überlagerung am transparenten OLED Display

Die zur Nutzung eines LCD notwendige Hintergrundbeleuchtung ist für den praktischen Einsatz in einer Werkzeugmaschine ungeeignet. Selbst bei ausreichender Beleuchtung des Innenraums kommt es zu Verschattungen und damit Einschränkungen der Darstellungsqualität. Daher wurde ein auf organischen Leuchtdioden (engl. organic light emitting diode, OLED) basierendes Display beschafft. Da OLEDs selbstleuchtende Eigenschaften besitzen, kann auf eine zusätzliche Hintergrundbeleuchtung verzichtet werden. Auch bessere optische Eigenschaften hinsichtlich Transparenz und Kontrast sprechen für ein OLED-Display.

Zu Präsentationszwecken wurde das transparente OLED-Display vor einer kleineren Werkzeugmaschine angebracht. Diese hauptsächlich für die Ausbildung konzipierte Werkzeugmaschine kann leichter transportiert werden und eignet sich daher besonders für Messeauftritte. Der in Abbildung 2-12 dargestellte Demonstrator wurde 2017 vom ISW auf den Fachmessen Hannover Messe Industrie und SPS/IPC/Drives in Nürnberg ausgestellt. In dieser Abbildung ist eine vollständige Überlagerung der Werkzeugmaschine mit ihrem digitalen Abbild zu sehen (engl. Virtual Reality, VR). Die Bewegungen der realen Maschine werden durch Synchronisierung mit Maschinendaten auf das digitale Abbild gespiegelt (vgl. Kapitel 6.3). Eine perspektivisch korrekte Darstellung entsprechend der aktuellen Position des

Betrachters wird durch ein oberhalb des Displays angebrachtes Tracking System und die Nachbildung der Perspektive des Betrachters erreicht (vgl. Kapitel 6.5).

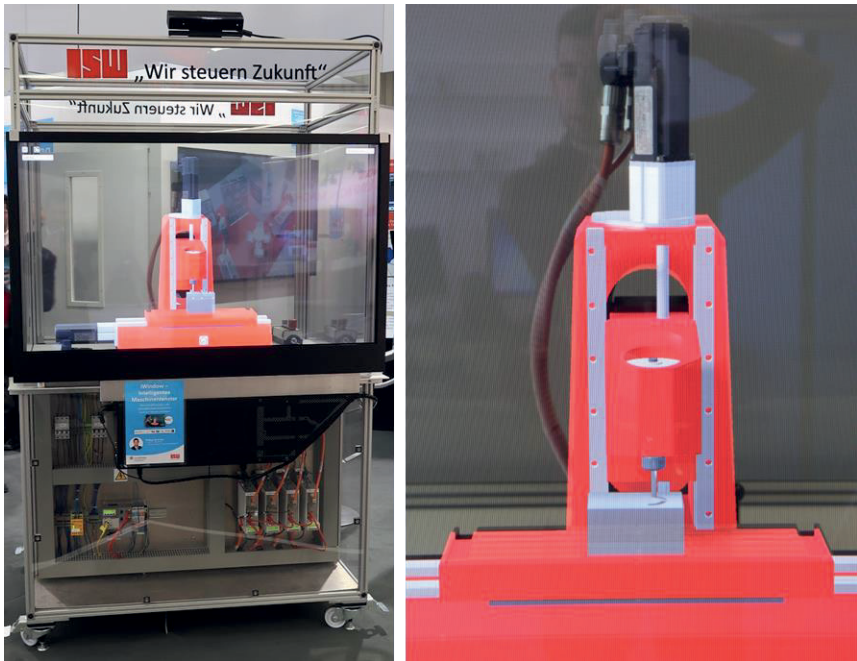


Abbildung 2-12: Vollständige Überlagerung mit virtuellem Abbild am OLED-Display (Quelle: ISW)

Durch Umschalten der Darstellungsart werden nur die in der Realität nicht vorhandenen Elemente eingeblendet, wodurch sich eine Erweiterung der Realität (engl. Augmented Reality, AR) ergibt. Abbildung 2-13 zeigt die reale Maschine bei der Bearbeitung eines virtuellen Werkstücks aus Sicht des Betrachters aus drei unterschiedlichen Positionen (rechts, mittig, links zur Displaymitte). Deutlich zu erkennen ist hierbei die korrekte Verdeckungsberechnung (vgl. Kapitel 6.4.3) zwischen Maschine, Werkstück und Werkzeug. Durch die integrierte Materialabtragssimulation (vgl. Kapitel 8.2) entspricht auch die aktuelle Geometrie des virtuellen Werkstücks einer realen Bearbeitung, unabhängig davon ob ein reales Werkstück vorhanden oder wie im Fall der Abbildungen nicht vorhanden ist. Bei real vorhandenem Werkstück dient die Abtragssimulation zur Verdeckungsberechnung. Ist kein echtes Werkstück eingespannt, kann durch das virtuelle Werkstück eine Vorabprüfung der Bearbeitung durchgeführt werden.

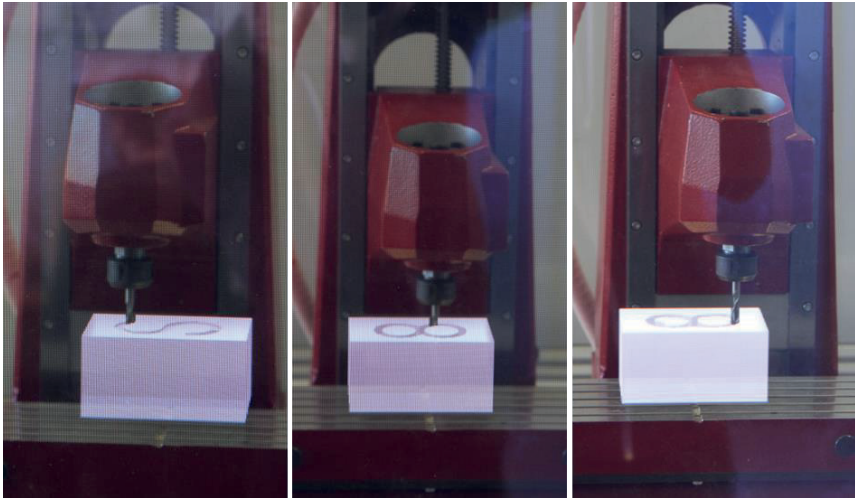


Abbildung 2-13: Perspektivische Überlagerung zur erweiterten Darstellung (AR) am transparenten OLED-Display (Quelle: ISW)

2.8 Finaler Demonstrator Werkzeugmaschine

Abschließend wurden die Ergebnisse der bisherigen Arbeiten in einem gemeinsamen Demonstrator zusammen geführt. Als Basis für diesen Demonstrator dient eine DMU 50 ecoMill Universalfräsmaschine der Firma DMG MORI. Um die verschiedenen Visualisierungsarten besser demonstrieren zu können, gliedert sich der finale Demonstrator in drei Teile, die im Folgenden vorgestellt werden.

2.8.1 Echtzeitvisualisierung im Bedienpanel

Die in Kapitel 2.5 bereits vorgestellte Multitouch-Oberfläche IndraWorks Operation von Bosch Rexroth kam ebenfalls auf dem Demonstrator DMU 50 ecoMill zum Einsatz. Diese beinhaltet ebenfalls die Möglichkeit einer Echtzeitvisualisierung in Form einer NC-Simulation. Dadurch wird der Bearbeitungsprozess der Maschine in Echtzeit auch auf dem Bedienpanel visualisiert. Dies wird ebenfalls durch eine Steuerungsanbindung per OPC UA erreicht, bei der Daten aus dem Echtzeitbereich durch das Bosch Rexroth CNC-Systems MTX bereitgestellt werden.

2.8.2 Perspektivische VR-Echtzeitvisualisierung an herkömmlichen Displays

Den zweiten Bestandteil des DMU50-Demonstrators stellt eine weitere an der rechten Seitenwand der Maschine angebrachte Version des intelligenten Maschinenfensters dar. Es besteht aus einem konventionellen 3D-Stereodisplay und einer Tracking-Kamera (Abbildung 2-14).

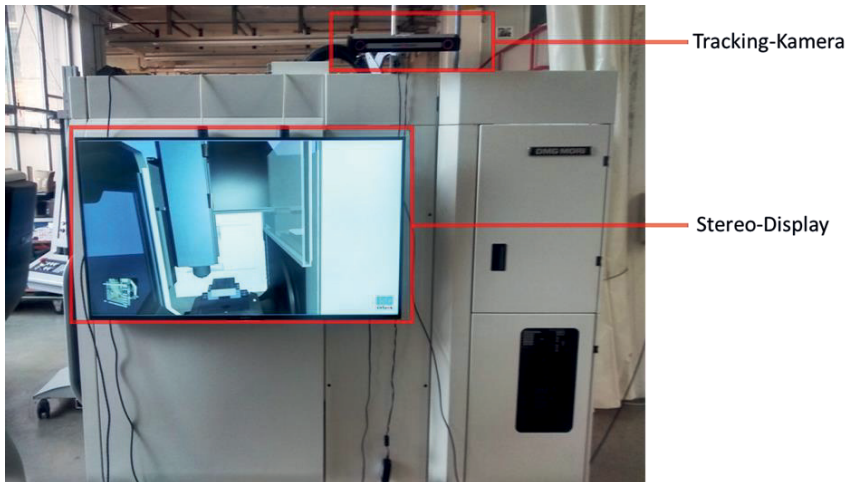


Abbildung 2-14: 3D-Stereo-Display und Head-Tracking-System an der iWindow-Werkzeugmaschine

Auf dem intelligenten Maschinenfenster ist ein virtuelles Modell des Maschineninnenraums zu sehen, welches sich anhand der Positionssollwerte der realen Steuerung zeitgleich zur realen Maschine bewegt. Im virtuellen Modell sind neben den statischen und bewegten Maschinenteilen sowohl die Werkzeuge in Spindel und Magazin enthalten, als auch die Spannmittel und Werkstücke. Im virtuellen Modell werden Werkzeugwechselvorgänge wie an der realen Maschine durchgeführt, indem das virtuelle Modell dieselben Steuersignale (z.B. Magazin einrücken/drehen, Werkzeug in Spindel spannen) erhält. Der Bearbeitungsfortschritt wird durch die Simulation des Materialabtrags detailliert in Echtzeit angezeigt. Um eine nahezu fotorealistische Darstellung des Maschineninnenraums zu ermöglichen, werden Oberflächen der 3D-Geometrien mit Texturen überlagert und metallische Oberflächen spiegelnd dargestellt.

Die virtuelle Darstellung des Maschineninnenraums wird weiterhin um computergenerierte Informationen, sogenannte Mehrwertdienste, angereichert. Wahlweise kann der bereits zurückgelegte Bearbeitungspfad als dreidimensionale Spur auf dem Werkstück angezeigt werden. Die Liniendarstellung der Spur kann für Analysezwecke abhängig von aktuellen Prozessparametern angepasst werden, beispielsweise durch Einfärben entsprechend der Bahngeschwindigkeit. Ebenfalls werden ansonsten unsichtbare Maschinenzustände im 3D-Modell durch farbliches Hervorheben von Maschinenteilen für den Werker erkennbar gemacht; im Beispiel ist dies die Einfärbung der Spindelaufnahme anhand des Spannzustands.

Um Kollisionen oder Fehler im NC-Teileprogramm frühzeitig während des Einrichtbetriebs zu erkennen, kann die Steuerung in den sogenannten „DryRun“-Modus geschaltet werden. So

werden die Sollwerte nur an das virtuelle Modell und nicht an die reale Maschine gesendet. Die Kollisionserkennung des virtuellen Modells berechnet auf Grundlage der CAD-Geometrien der Maschinenteile auftretende Kollisionen. Diese werden auf dem Display durch Einfärben einer Schattenkopie der Kollidierenden Körper zum Kollisionszeitpunkt angezeigt. Gleichzeitig wird über einen in das iWindow-Framework integrierten Mehrwertdienst eine Kollisionsmeldung auf dem frontseitigen AR-Display angezeigt.

Das Modell auf dem intelligenten Maschinenfenster kann wahlweise in 3D-Stereo angezeigt werden. Die Betrachter müssen dann Shutter-Brillen benutzen, um das virtuelle Modell als dreidimensionales Modell zu betrachten.

Durch das über dem intelligenten Maschinenfenster angebrachte Tracking-System wird die genaue Position eines Betrachters erfasst, um die Visualisierung des Maschineninnenraums auf dem Display perspektivisch, d.h. relativ zum Betrachter, darzustellen. Durch die perspektivische Darstellung entsteht für den Betrachter der Eindruck, durch ein in die Maschinenwand eingelassenes Fenster in die Maschine zu blicken.

Auf die Details der Erarbeitung der einzelnen Technologien und Mehrwertdienste wird in den nachfolgenden Kapiteln eingegangen:

- Die Bereitstellung des für die Steuerungsanbindung genutzten OPC UA-Servers in Kapitel 5.
- Die Anbindung des virtuellen Modells per OPC UA in Kapitel 9.1.
- Die Modellierung von Werkzeugen und des Werkzeughandlings in Kapitel 9.3.
- Die Bereitstellung der Abtrags-, Kollisions und Spurbildungsmehrwertdienste in Kapitel 8.
- Die Anwendung der Abtrag-, Kollisions- und Spurbildungssimulation in Kapitel 9.4.
- Die Methoden für die perspektivische Darstellung des Maschineninnenraums in Abhängigkeit von der Lage des Betrachters in den Kapiteln 6 und 9.2.

2.8.3 Augmented Reality mittels transparenter Displays

Das in Kapitel 2.7 bereits vorgestellte OLED-Display wurde, wie in Abbildung 2-15 links dargestellt, vor dem bestehenden Maschinenfenster des Demonstrators befestigt. Zur perspektivischen Darstellung der Inhalte, sowohl rein virtuell wie auch überlagernd, wurde oberhalb des Displays ein Tracking System angebracht (vgl. Kapitel 6.5). Dieser Aufbau entspricht großteils dem in Kapitel 2.6 vorgestellten Demonstrator mit LCD. Lediglich das OLED-Display, welches den ursprünglich verwendeten LCD ersetzt, wurde entgegen dem vorigen Aufbau fest mit der Maschinentür verbunden. Das im Rahmen des vorausgehenden Demonstrators bereits entwickelte Computermodell des Maschineninnenraums wurde ebenfalls übernommen. Abbildung 2-15 rechts zeigt die perspektivische Visualisierung des rein virtuellen Maschineninnenraums (VR) auf Basis dieses Computermodells. Diese Darstellungsart kann verwendet werden, wenn eine Sicht in den realen Innenraum

beispielsweise auf Grund von Kühlschmiermittel nicht möglich ist. Hierzu ist es notwendig, dass das dargestellte Computermodell zu jeder Zeit dem Zustand der tatsächlichen Maschine entspricht (vgl. Kapitel 6.3).

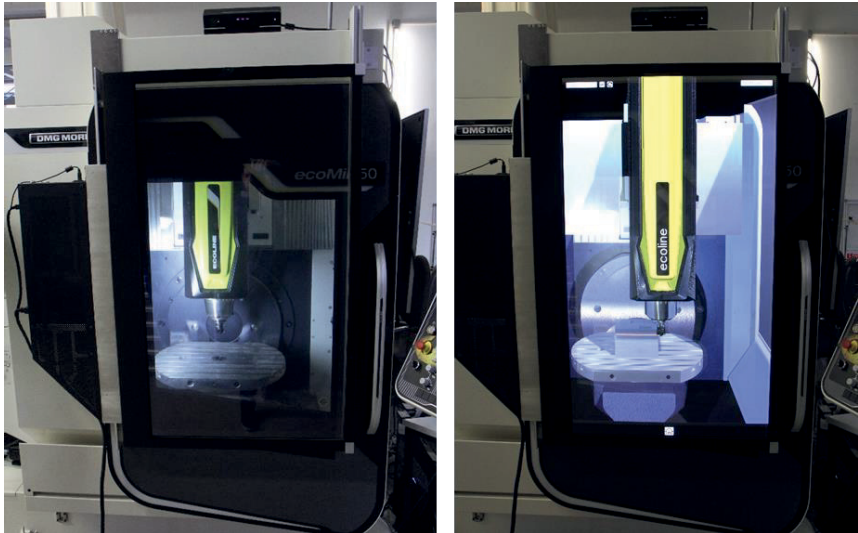


Abbildung 2-15: Links: Transparentes OLED-Display vor dem Maschinenfenster; Rechts: Perspektivische Darstellung des virtuellen Maschineninnenraums (Quelle: ISW)

Alternativ zur rein virtuellen Darstellung (VR) können auch ausschließlich in der Realität nicht vorhandenen Elemente zur Erweiterung des realen Innenraums eingeblendet werden (AR). Beispielsweise kann, wie in Abbildung 2-16 aus drei unterschiedlichen Betrachterpositionen (rechts, mittig, links zur Displaymitte) dargestellt, ein virtuelles Werkstück in den realen Innenraum eingeblendet und durch die reale Maschine bearbeitet werden. Entsprechend der Ergebnisse des Demonstrators aus Kapitel 2.7 kommen auch hier die Synchronisierung des digitalen Abbilds mit Maschinendaten (vgl. Kapitel 6.3), eine integrierte Materialabtragssimulation (vgl. Kapitel 8.2) sowie eine Verdeckungsberechnung zwischen realem Werkzeug und virtuellem Werkstück (vgl. Kapitel 6.4.3) zum Einsatz.

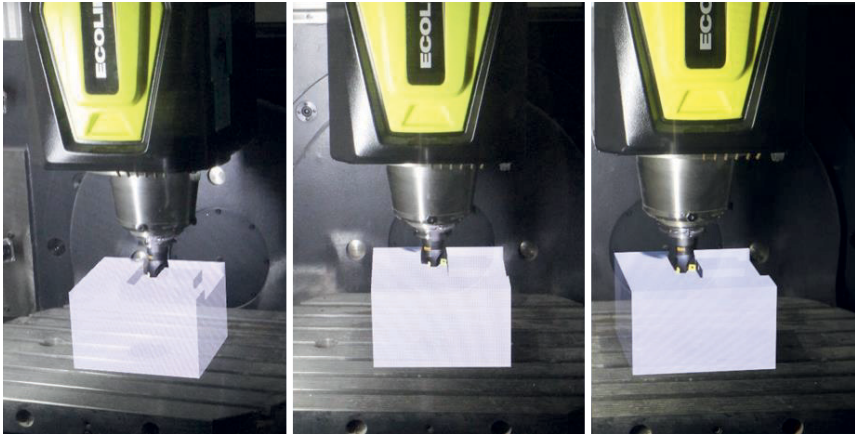


Abbildung 2-16: Perspektivische Darstellung eines virtuellen Werkstücks am finalen Demonstrator mittels transparentem OLED Display (Quelle: ISW)

Durch Anbindung der Visualisierung an das Framework für Mehrwertdienste (vgl. Kapitel 4) werden zusätzliche durch entsprechende Dienste generierte Inhalte ebenfalls auf dem Maschinenfenster eingeblendet. Abbildung 2-17 zeigt die Visualisierung der zurückliegenden Werkzeugbahn sowohl in der überlagernden (AR, links) wie auch in der rein virtuellen (VR, rechts) Darstellung. Da sämtliche die jeweilige Darstellungsart betreffende Funktionen, z.B. Überlagerung und Verdeckungsrechnung, von der Visualisierungskomponente übernommen werden, müssen diese nicht bei der Entwicklung eines Mehrwertdienstes berücksichtigt werden. Derselbe Mehrwertdienst kann sowohl für die rein virtuelle Darstellung, die kamerabasierte Überlagerung und die Überlagerung am transparenten Display verwendet werden.

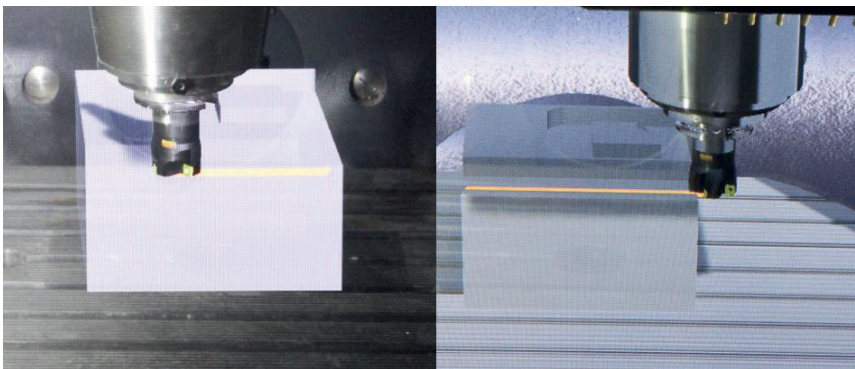


Abbildung 2-17: Visualisierung der Werkzeugbahn mittels Mehrwertdienst; Links: Augmented Reality; Rechts: Virtual Reality (Quelle: ISW)

3 Unterstützung beim Absortiertvorgang in der Blechfertigung

M. Kiefer, R. Rahäuser

3.1 Ausgangssituation und Zielsetzung

Bei 2D-Laserschneidanlagen werden mit einem in einer Ebene bewegtem Laser Bauteile aus einem Blech geschnitten, welches auf einer Palette im Arbeitsraum gelagert ist. Nach dem Schneidvorgang wird die Palette aus dem Arbeitsraum der Maschine gefahren und die Teile entnommen.

Eine eigens durchgeführte Studie zeigt, dass die Entnahme der lasergeschnittenen Bauteile fast ausschließlich manuell vom Bediener erfolgt. Zur Sortierung der Teile hat der Bediener in der Regel die erforderlichen Informationen für jedes geschnittene Teil in Form eines ausgedruckten Begleitpapiers vorliegen. Auf diesem sind für jedes Teil beispielsweise die Teilegeometrie, der folgende Bearbeitungsprozesse und der Kundenauftrag aufgelistet (vgl. Abbildung 3-1).

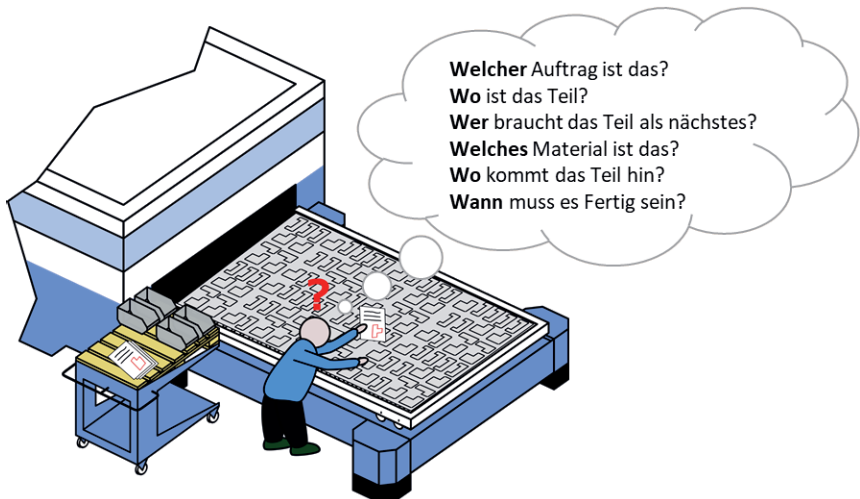


Abbildung 3-1: Schematische Darstellung des manuellen Absortiertvorgangs bei einer 2D-Laserschneidanlage (Quelle: TRUMPF)

Die manuelle Vereinzelung und Sortierung der Teile ist hierbei relativ mühsam und zeitaufwändig und besitzt dadurch einen hohen Zeitanteil am gesamten Herstellungsprozess. Durch die hohe Flexibilität des Laserschneidens kann bereits auf einem einzelnen Blech eine enorme Teilevielfalt entstehen. Dies macht einen ständigen Abgleich mit den Begleitpapieren

notwendig. Fehler sind unvermeidbar, die oftmals erst in Folgeprozessen bemerkt werden und dann umso aufwändiger gelöst werden müssen. Mit sinkenden Losgrößen und einer steigenden Zahl an Einzelaufträgen verschärft sich das Problem und führt in der Folge zu erheblichen Wirtschaftlichkeitsnachteilen.

Das Ziel ist es, ein Assistenzsystem zu entwickeln, welches dem Bediener ergonomisch in Echtzeit relevante Informationen zu den Teilen bereitstellt. Damit soll das Sortieren sowie das Vereinzeln der Teile unterstützt und die entnommenen Teile automatisch korrekt verbucht werden. Dadurch kann auf die Sortierung mittels ausgedruckter Begleitpapiere verzichtet und die beschriebenen Nachteile vermieden werden. Der Bediener soll dabei durch das Assistenzsystem unterstützt werden, jedoch sollen seine individuellen Ablaufpräferenzen voll erhalten bleiben. Feste Handgriffe sollen nicht vorgeschrieben, sondern lediglich die aktuell benötigte Information verständlich und ergonomisch bereitgestellt werden.

Eine durchgeführte Studie in mehreren Betrieben der Blechbearbeitung zeigt das enorme Potential durch die Einführung eines Assistenzsystems (vgl. Abbildung 3-2). Dabei wurde das Absortieren für zwei typische Auftragsszenarien untersucht. Je nach Situation können pro Auftrag Zeiteinsparungen von etwa 30 % bis 50 % erzielt werden. Daneben wird die Anzahl falsch verbuchter Teile durch das automatische Verbuchen mit dem Absortierassistent vermieden.

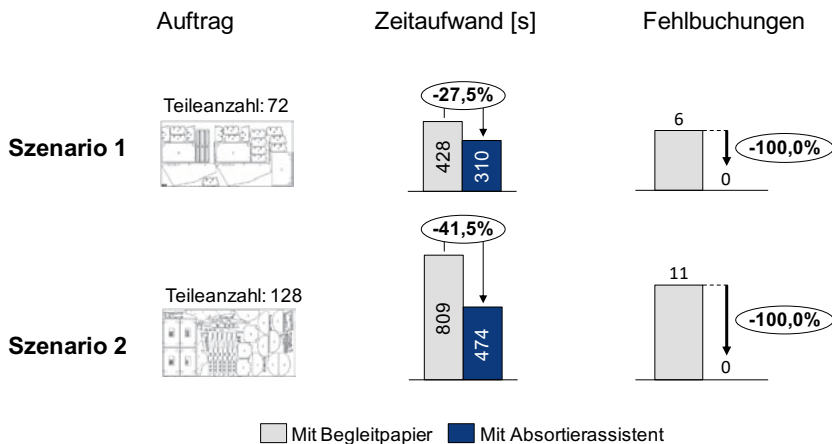


Abbildung 3-2: Repräsentative Studie zur erzielbaren Effizienz durch ein Absortierassistenten anstelle des Absortierens mithilfe ausgedruckter Begleitpapiere. (Quelle: TRUMPF)

3.2 Absortierassistentz mittels kamerabasierter Entnahmedetektion

Als Lösungskonzept zur Unterstützung des Absortiervorgangs wird eine kamerabasierte Entnahmedetektion der Teile verfolgt. Das Lösungskonzept ist in Abbildung 3-3 schematisch dargestellt.

Zur Detektion der entnommenen Teile wird der Absortierbereich hinter der Maschine mit Kameras beobachtet und aus den Kamerabildern durch eine Bildtransformation Bilder in der Ebene rekonstruiert. Bild 0 bis Bild 4 verdeutlicht die Aufnahme mehrere Bilder während dem Absortiervorgang. Das jeweils binarisierte Differenzbild vor Bild 0 und nach Bild 4 der Teilentnahme wird mit der dazugehörigen Tafelbelegung, also der Sollgeometrie und der Lage der Teile, korreliert. Dieser Abgleich liefert eindeutig räumlich die Bauteilidentifikation jedes entnommen Teils der Tafel.

Auf diese Weise kann durch den Vergleich der hintereinander aufgenommenen Bilder durch den Vergleich der Differenzbilder mit der Tafelbelegung die entnommen Teile während dem Absortiervorgang zugeordnet und verbucht werden.

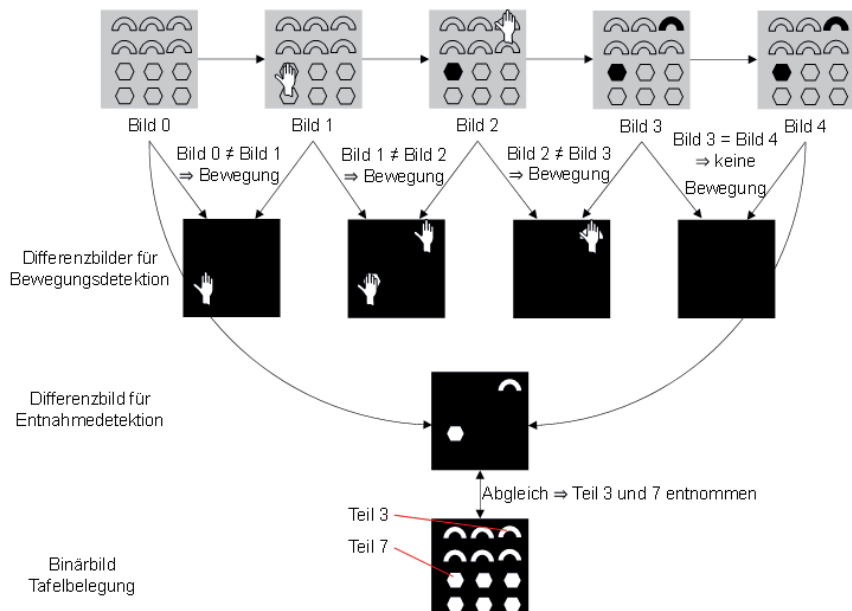


Abbildung 3-3: Konzept der Entnahmedetektion mithilfe dem bildbasierten Vergleich der Tafelgeometrie mit dem Kamerabild der geschnittenen Teile auf dem Blech. (Quelle: TRUMPF)

Im Rahmen des iWindow-Projekts wurde zusammen mit den Projektpartnern ein Großteil der Methoden des dargestellten Lösungskonzepts erarbeitet und untersucht. Die Arbeitsschwerpunkte waren dabei folgende:

- Erfassung des Absortierbereichs mit Kameras
- Ermittlung der Bilder in der Ebene aus den Kamerabildern
- Erfassung der Tafelgeometrie auf Basis des NC-Programms

Auf die Ergebnisse der drei Punkte wird im Folgenden näher eingegangen. Anschließend wird auf den Nachweis der Funktionsfähigkeit des Assistenzsystems eingegangen.

3.3 Erfassung des Absortierbereichs mit Kameras

Zur Erfassung der geschnittenen Teile auf der Palette hinter der Maschine dient ein Kamerasystem bestehend aus drei Flächenkameras. Die Kameras werden aufgrund der Umgebungsbedingungen an die Rückwand der Lasermaschine angebracht. Daraus resultiert eine schräge Perspektive auf die Szene (vgl. Abbildung 3-4).

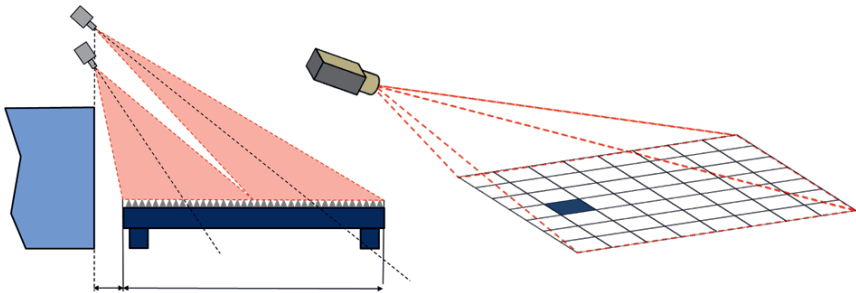


Abbildung 3-4: Bildaufnahme der Palette aus schräger Perspektive (Quelle: TRUMPF).

Der relevante Bereich der Erfassung entspricht der Ebene der Palettenauflage, auf denen die aus dem Blech geschnittenen Teile liegen. Eine qualitative Untersuchung ergab, dass die wirtschaftlich sinnvollste Lösung drei Kamera-Module beinhaltet. Die dadurch erreichte Flächensegmentierung ist in Abbildung 3-5 veranschaulicht. Die Kameras C2 und C3 erfassen in etwa 50 % der weiter entfernten Fläche und die mittig platzierte Kamera C1 das nähere liegende Feld.

Die Zusammenführung der Kamerabilder zu der resultierenden Szene wird im nachfolgenden Kapitel beschrieben. Das Konzept lässt sich für die Arbeitsraumdimensionen von 3mx1.5m und 4mx2m adaptieren. Damit werden die am weitesten verbreiteten Maschinenausprägungen abgedeckt.

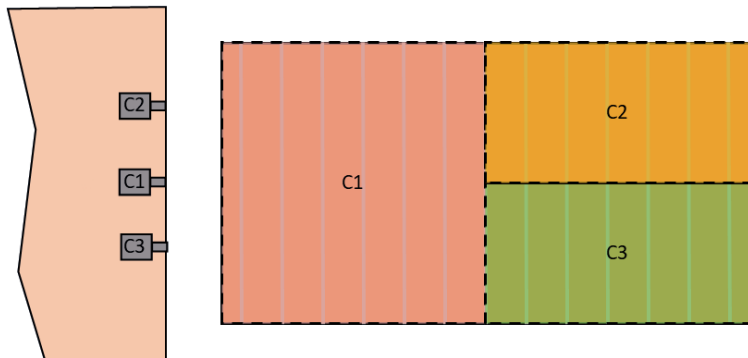


Abbildung 3-5: Flächensegmentierung zu Kamerasystem. (Quelle: TRUMPF)

3.4 Ermittlung der ebenen Bildszene auf Basis der Kamerabilder

Um über das Differenzbildverfahren die entnommen Teile im Absortierbereich erkennen zu können muss aus den drei, schräg aufgenommenen Kamerabildern ein einzelnes Bild in der Ebene errechnet werden. Hierzu muss das Eingangsbild jeder Kamera in die 2D-Ebene der Palettenauflage transformiert und dann zu einem Bild zusammengeführt werden.

Um die Kamerabilder transformieren zu können, muss eine Kamerakalibrierung erfolgen. Hierfür können spezielle Kalibriermuster (Targets) verwendet werden. Das sogenannte Circle-Pattern als Target aus Blech erwies sich vorteilhaft gegenüber gewöhnlichen Schachbrettmustern. Zum einen kann ein Kreismuster gegenüber einem Schachbrettmuster direkt mit der Laserschneidanlage aus einem Blech geschnitten werden (vgl. Abbildung 3-6). Dadurch kann auf einfache Weise ein Pattern erstellt werden und der Service-Techniker muss zur Kamerakalibrierung beim Kunden nicht immer ein Pattern mit sich führen. Zum anderen erwies sich aus der schrägen Perspektive das Kreismuster als vorteilhaft, da der Einfluss der sogenannten Tonnenverzerrung nahezu vollständig kompensiert werden kann. Dieser Aspekt wird in Kapitel 7 näher erläutert.

Das Ergebnis der Kalibrierung kann als wichtigster Meilenstein betrachtet werden. Die Präzision der Kalibrierung ist Grundlage für eine hochgenaue Bildtransformation.

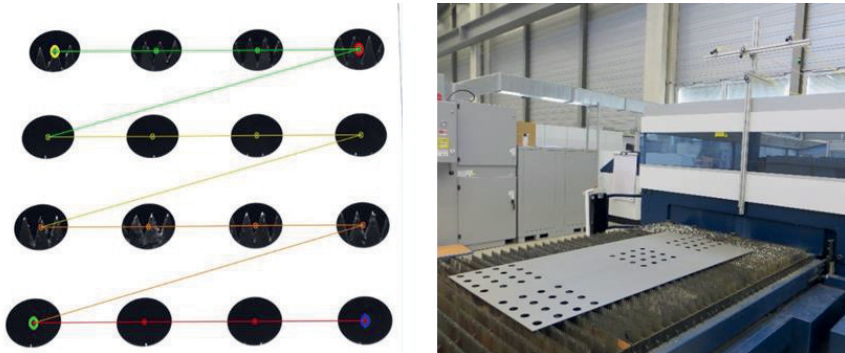


Abbildung 3-6: Aus einem Baustahlblech geschnittene Circle-Pattern zur Kamera-kalibrierung. (Quelle: TRUMPF)

Wie am Anfang des Kapitels beschrieben, müssen in einem weiteren Schritt die drei in die Ebene transformierten Kamerabilder noch zur gemeinsamen Bildszene zusammengeführt werden (vgl. Abbildung 3-7). Die Zusammenführung der Kamerabilder zu einer einzelnen Szene wird auch als *Stitching* bezeichnet und muss im betrachteten Anwendungsfall zeitsynchron und in Echtzeit erfolgen, da die Teileentnahme im Absortierbereich robust aus dem Vergleich der aufgenommenen Szenen erkannt werden muss. Hierzu werden die einzeln generierten, transformierten Bilder aus Kamera 1, 2 und 3 direkt in einen gemeinsamen Bildspeicher überführt. An den Nahtstellen kann die Qualität der Kalibrierung überprüft werden. Hier konnte ein maximaler Fehler von einem Pixel erreicht werden, sodass dieses Resultat ausreichend hohe Qualität bietet für die konkrete Anwendung. Das Beispiel in Abbildung 3-8 zeigt reale Bildaufnahmen der drei Kameras und das aus dem *Stitching* resultierende Bild der gesamten Szene.

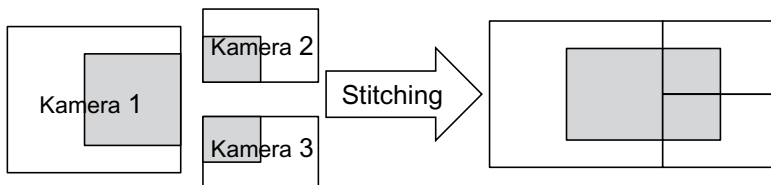


Abbildung 3-7: Schemaskizze zum *Stitching* der resultierenden Bildszene aus den transformierten Kamerabildern. (Quelle: TRUMPF)

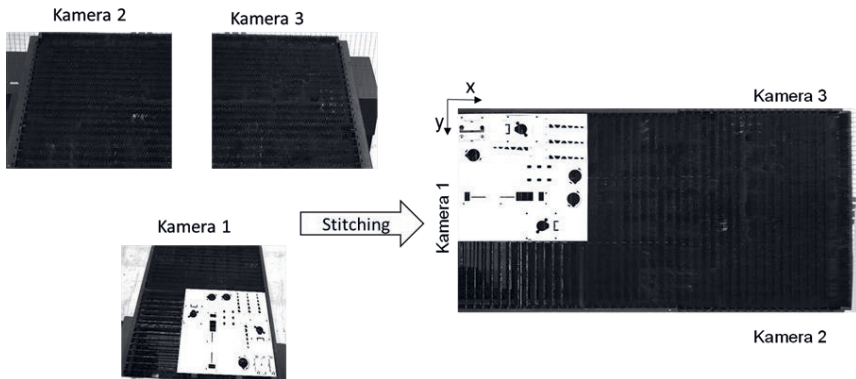


Abbildung 3-8: Reale Bildaufnahmen Kamera 1, 2 und 3 (links) und das zusammengeführte, transformierte Bild (rechts). (Quelle: TRUMPF)

3.5 Erfassung der Tafelgeometrie auf Basis des NC-Programms

Durch den Vergleich der zeitlich hintereinander aufgenommenen Bildszenen können die Differenzbilder erzeugt werden. Wie in Abbildung 3-3 gezeigt ist, wird nach der Ermittlung des Differenzbilds durch den Abgleich des Differenzbilds mit der Tafelgeometrie, also der Lage und Geometrie der Teile, das entnommene Teil erkannt.

Die Tafelgeometrie wird über das NC-Programm generiert. In diesem sind zum einen maschinenspezifische Zyklen-Befehle zum Laserschneiden enthalten, beispielsweise Zyklen zum Zünden und Ausschalten des Lasers. Daneben sind im NC-Programm gemäß der Norm DIN 66025 respektive ISO 6983 die Verfahrensanweisungen der NC-Achsen zur Bewegung des Laserschneidkopfs enthalten. Über die Verfahrensanweisungen ist die resultierende Tafelgeometrie definiert. Um diese aus dem NC-Programm zu generieren wurde daher ein entsprechender NC-Parser entwickelt, welcher die Verfahrensanweisungen und die maschinenspezifischen Zyklen korrekt interpretieren kann und die Tafelgeometrie erzeugt.

Der zweidimensionale Bearbeitungsvorgang beim Laserschneiden ermöglicht die direkte Überführung von Werkzeugbahn zur Bauteilgeometrie. Die sogenannte Bahnkorrektur als Bahnversatz gemäß der laserspezifischen Schnittspaltbreite ist die geometrische Differenz zwischen resultierender Bauteilkontur und Werkzeugbahn des Lasers. Die im Vergleich zu den geschnittenen Bauteilen sehr kleine Schnittspaltbreite von typischerweise 150 µm erlaubt die Vereinfachung, dass für die Ermittlung der Tafelgeometrie die Werkzeugbahn bei eingeschaltetem Laserstrahl als Werkstückkontur angenommen werden kann.

Als Ausgangsformat des entwickelten NC-Parsers dient das Austauschformat XML. Darin werden die Bauteilhierarchie, Lage und Bauteilgeometrie generisch abgebildet. Diskrete

Punkte in Form von Polygonzügen beschreiben dabei alle bauteilspezifischen Konturen (vgl. Abbildung 3-9).

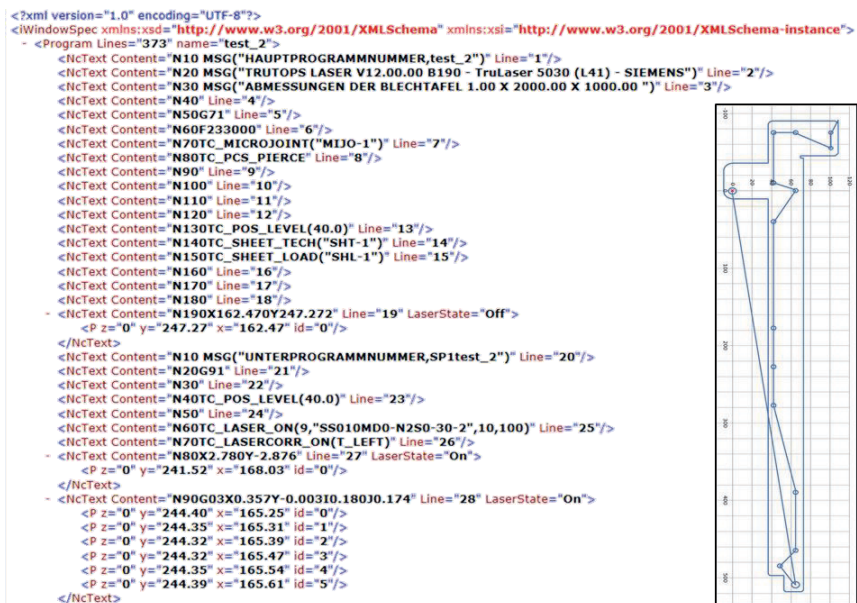


Abbildung 3-9: Auszug aus XML Spezifikation (links) und Darstellung der Tafelgeometrie (rechts). (Quelle: TRUMPF)

3.6 Nachweis der Funktionsfähigkeit des Assistenzsystems

Die Erprobung der kamerabasierten Detektion der Teileentnahme zeigt eine hohe Robustheit auch bei sich ändernden Beleuchtungsbedingungen. Hierbei ist der Umgang mit der Änderungsgeschwindigkeit von externen Beleuchtungsbedingungen die größte Herausforderung (vgl. Kapitel 7).

Die Detektion der Teileentnahme ist beispielhaft in Abbildung 3-10 dargestellt. Das Eingangsbild Bild A entspricht der Aufnahme vor der Entnahme eines oder mehrere Teile. Bild B entspricht der Aufnahme nach erfolgter Entnahme. Durch mathematische Subtraktion (Differenzbild) wird der Bildanteil mit den Änderungen eindeutig sichtbar. Durch anschließende Korrelation von Differenzbild mit der korrespondierenden Tafelgeometrie können entnommene Teile eindeutig erkannt werden. Damit können selbst baugleiche Teile eindeutig durch ihre Lage auf der Tafel erkannt werden.

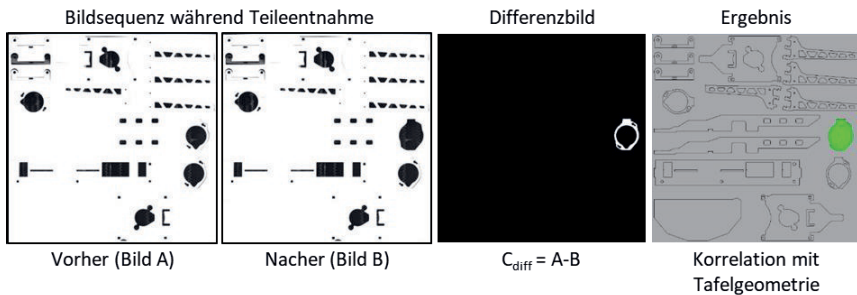


Abbildung 3-10: Bildsequenzen zur Algorithmik der Entnahmedetektion. (Quelle: TRUMPF)

Diese Technologie hat gegenüber klassischen Ansätzen mit Pattern-Matching Algorithmen erhebliche Vorteile und erlaubt selbst unter beschränkter Auflösung der Szene eine robuste Lösung. Da dieses Konzept auf Fremdlicht als Beleuchtungsgrundlage beruht, hat dieses einen erheblichen Einfluss welcher durch mehrere Maßnahmen begrenzt werden kann. Mit einer adaptiven Belichtungssteuerung und eine gezielte Überbelichtung kann der Einfluss stark reduziert werden. Weitere Optimierung der erarbeiteten Methoden wird als sinnvoll erachtet und wird auch nach Abschluss des Forschungsprojekts weiterverfolgt.

Die Benutzerschnittstelle für Bediener war nicht Bestandteil dieses Forschungsprojekts und wird deshalb hier nicht näher als Ergebnis vorgestellt. Auf einer webbasierten Oberfläche stellt sie letztlich dem Bediener alle notwendigen Informationen und Interaktionen zur Verfügung, um die jeweils entnommenen Teile auf die zur Verfügung stehenden Ablageflächen zu sortieren, ohne dass hierzu die auftragsspezifischen Begleitpapiere notwendig sind, wie es in der Ausgangssituation beschrieben wird.

4 Dynamisch erweiterbares Framework für Mehrwertdienste

P. Sommer, A. Verl

4.1 Motivation und Anforderungen

Wie bereits in Kapitel 1.2 erläutert, gliedern sich die Informationen des intelligenten Maschinenfensters in drei Informationsebenen (dargestellt in Abbildung 4-1):

- Ebene 1: Maschineninnenraum
- Ebene 2: Mehrwertdienste
- Ebene 3: HMI

Die Mehrwertinformationen der Ebene 2 werden hierbei nicht unveränderlich zum Zeitpunkt der Entwicklung der Maschine vordefiniert, sondern je nach Bedarf durch Mehrwertdienste (MWDe) generiert. Dieses Kapitel befasst sich mit der Gestaltung dieser Mehrwertdienste und dem zugehörigen Framework in dem die Dienste ausgeführt werden.

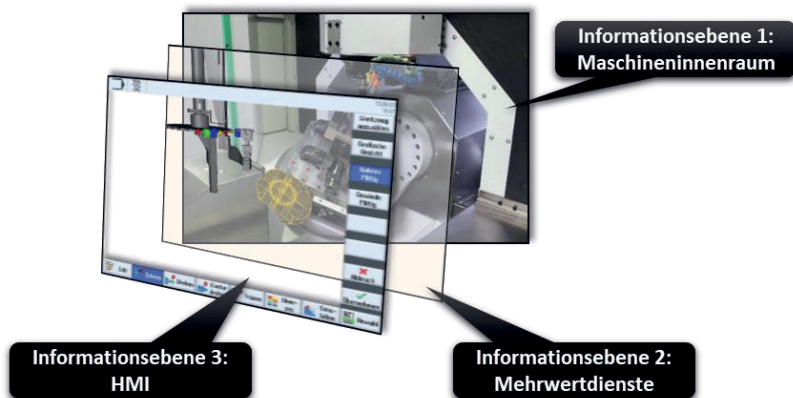


Abbildung 4-1: Informationsebenen des intelligenten Maschinenfensters (Quelle: ISW)

Ein Mehrwertdienst erweitert das intelligente Maschinenfenster um zusätzliche Funktionalität und unterstützt den Bediener der Werkzeugmaschine bei seiner Arbeit. Die Aufgaben eines Maschinenbedieners umfassen Vorbereitung, Einrichtung, Überwachung und Optimierung der Bearbeitung [1, 2, 3]. Hierbei ergeben sich mehrere Problemstellungen für die der Maschinenbediener heute wenig bis keine Hilfestellung erfährt.

Zur Fertigung neuer Werkstücke (Erstfertigung, Prototypen, Einzelstücke, Nachproduktion) gibt es zwei Vorgehensweisen. Das benötigte Maschinenprogramm (G-Code) wird entweder mittels CAD und CAM in der Arbeitsvorbereitung erstellt oder durch den Maschinenbediener anhand einer Zeichnung direkt an der Maschine programmiert.

Im ersten Fall erhält der Maschinenbediener das Programm in Form einer „Black Box“ über deren Inhalt er nur wenige Informationen besitzt. Beispielsweise sind Werkzeugbahnen, Verfahrwege und Arbeitsschritte nicht offensichtlich. Mögliche Kollisionen (z.B. mit Spannmitteln) können vorab nicht abgeleitet und überprüft werden. Auch eine Prüfung des verwendeten Rohteils gegenüber dem Zielwerkstück ist nicht möglich.

Die Programmierung direkt an der Maschine erfordert einen Transfer der geometrischen Angaben in der Zeichnung in Programmcode. Diese Aufgabe ist nicht intuitiv und fehleranfällig. Hilfreich wäre hierbei beispielsweise eine Abstrahierung der Bewegungen, die Visualisierung von Kollisionen, Schnitttiefen und Vorschub. Auch Unterstützung zur Optimierung von Maschinenprogrammen (z.B. Eilgangbewegungen) ist denkbar.

Es bedarf folglich neue an die jeweilige Aufgabe angepasste Assistenzfunktionen für den Maschinenbediener. Solche Assistenzfunktionen sind heute bereits vorhanden, jedoch meist nur in CAD-, CAM- oder Simulations-Tools in der Büroumgebung (vergleiche Abbildung 4-2, links). In wenigen Fällen, in denen unterstützende Funktionalität bereits in Maschinen integriert wurde, erfolgte dies üblicherweise in Form von fest einprogrammierter Anwendersoftware durch den Maschinenbauer (siehe Abbildung 4-2, rechts). Eine nachträgliche Erweiterung der Funktionalität und eine Anpassung auf die aktuellen Bedürfnisse des Maschinenbedieners sind nicht möglich.

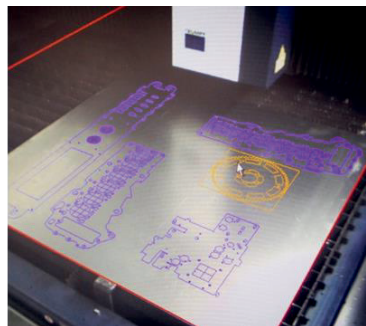
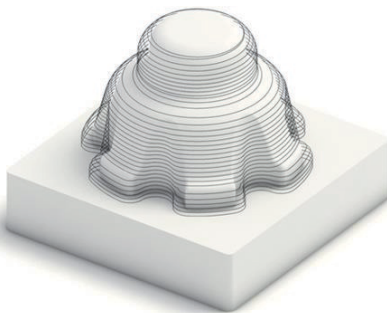


Abbildung 4-2: Links: Bahnvisualisierung (Quelle: ModuleWorks); Rechts: Platzierung von Schnittkonturen (Quelle: TRUMPF)

Die zukunftsfähige Gestaltung von innovativen Mensch-Maschine-Schnittstellen erfordert jedoch eine solche nachträgliche Erweiterbarkeit der Funktionalität sowie eine individuelle Anpassung unter der Verwendung von mehrwertgenerierenden Softwarediensten [4]. Hierbei dient das Smartphone als Vorbild. Wird eine neue Funktionalität oder Hilfe benötigt, öffnet der Nutzer den Appstore, sucht sich eine passende App aus, lädt und installiert diese unkompliziert auf sein Endgerät und kann diese dann anschließend sofort nutzen. Dieses Konzept der App-Orientierung lässt sich auf die Benutzerschnittstelle von

Werkzeugmaschinen übertragen (siehe Abbildung 4-3). Durch nachträgliche Installation und Nutzung von Mehrwertdiensten direkt an der Maschine lässt sich die Benutzerschnittstelle auf die aktuellen Bedürfnisse und Anforderungen des Maschinenbedieners anpassen. Des Weiteren führt die nachträgliche Erweiterbarkeit dazu, dass die verfügbaren Assistenzfunktionen auch nach der Auslieferung der Maschine aktuell gehalten werden können und sich Updatezyklen verringern. Durch die offene Gestaltung eines solchen Mehrwertdienstsystems, welche es sowohl Kunden wie auch externe Softwareanbieter erlaubt, Dienste zu entwickeln und anderen über einen Store zur Verfügung zu stellen, können diese ebenfalls an der Wertschöpfung mitwirken. Dadurch steigen einerseits die Auswahl und Vielfalt der verfügbaren Dienste, andererseits reduzieren sich Kosten bei Maschinenbauern durch die Verlagerung der Entwicklung neuer Services zu Kunden, Lieferanten und Serviceprovidern (Prosumer-Modell) [4].



Abbildung 4-3: Vorbild Smartphone (Quelle: ISW)

Aus dieser Zielstellung ergeben sich verschiedene Anforderungen an die Gestaltung der Mehrwertdienste und dem zugehörigen Framework zur Ausführung der Dienste. Diese Anforderungen lassen sich, wie in Abbildung 4-4 dargestellt, in drei Kategorien gliedern:

- Gestaltung eines erweiterbaren Plug-In-Systems
- Sicherheit und Zugriffsrechte von Diensten
- Entwicklung und Vertrieb von Diensten



Abbildung 4-4: Anforderungskategorien (Quelle: ISW)

4.1.1 Anforderungen an die Gestaltung eines erweiterbaren Plug-In-Systems

Wie bereits erläutert besteht Bedarf die Funktionalität der Benutzerschnittstelle und damit des intelligenten Maschinenfensters nachträglich mit mehrwertgenerierenden Softwarediensten erweitern zu können. Diese App-Orientierung führt zu einer Art Plug-In-System, in das sich neue Mehrwertdienste zur Laufzeit des Systems durch Download und Installation integrieren und anschließend nutzen lassen. Da solche Dienste zukünftig nicht nur vom Hersteller der Maschine, sondern auch von Kunden, Lieferanten und Serviceprovidern zur Verfügung gestellt werden, können nicht alle Dienste durch Einspielen eines einzelnen Softwareupdates aktualisiert werden. Einzelne Updates je Dienst erfordern einen Mechanismus zum Verteilen und Einspielen dieser, unabhängig von anderen Diensten und vom Basissystem des Maschinenfensters. Diese Anforderungen führen zur Verwendung einer Service-orientierten Architektur.

4.1.2 Anforderungen an Sicherheit und Zugriffsrechte von Diensten

Die Entwicklung der Dienste durch verschiedene Anbieter führt zu Anforderungen hinsichtlich der Betriebs- (Safety) und der Datensicherheit (Security). Ein Mehrwertdienst muss trotz Fehler oder Abstürze anderer Dienste lauffähig bleiben und darf nicht durch diese beeinträchtigt werden. Dies führt zu voneinander unabhängigen Diensten, die mittels Sandboxing voneinander isoliert werden müssen, und zugehörigen Kommunikationsmechanismen. Mittels Sandboxing kann verhindert werden, dass Fehler innerhalb einer Anwendung zu Auswirkungen oder Abstürzen in anderen Anwendungen führen und somit die Betriebssicherheit gesteigert werden [5]. Des Weiteren beschreibt Sandboxing die Ausführung einer Anwendung in einer begrenzten Umgebung ohne Zugriff auf das System und auf Ressourcen auf die nicht ausdrücklich Zugriff gewährt wurde [6]. Dies ermöglicht im Sinne der Datensicherheit einerseits die Trennung der Datenräume von Anwendungen und andererseits die Kontrolle von Zugriffen auf Systemressourcen. Darüber hinaus müssen auch die Zugriffsrechte auf Daten anderer Dienste, wie beispielsweise Maschinendaten, geregelt werden. Hierzu werden entsprechende Mechanismen in der Kommunikation zwischen Diensten benötigt.

4.1.3 Anforderungen an Entwicklung und Vertrieb von Diensten

Auch hinsichtlich Entwicklung und Vertrieb von Diensten ergeben sich Anforderungen durch die Möglichkeit verschiedener Anbieter der Mehrwertdienste. Da verschiedene Entwickler auch verschiedene Präferenzen hinsichtlich Systemumgebung und Programmiersprache haben, müssen hier möglichst viele Alternativen unterstützt werden. Virtualisierung beschreibt die Abstraktion von physikalischen Ressourcen und macht damit ein Anwendungsprogramm von der Zielplattform (z.B. Hardware, Betriebssystem, Ressourcen) unabhängig. Durch Virtualisierung kann folglich eine Anwendung unabhängig des Zielsystems entwickelt werden. Weiterhin muss auch der verwendete Kommunikationsmechanismus mit

verschiedenen Programmiersprachen nutzbar sein, um Entwickler nicht einzuschränken. Damit externe Entwickler ebenfalls die Möglichkeit haben ihre Dienste einem großen Kundenkreis zur Verfügung zu stellen und dieser wiederum sehr einfach auf ein möglichst großes Angebot von Diensten zugreifen kann, ist eine zentrale Vertriebsplattform erforderlich. Der Appstore heutiger Smartphones dient als Vorbild für die Anforderung an einen Mehrwertdienststore für das intelligente Maschinenfenster.

4.2 Mehrwertdienste

Abgeleitet aus der Motivation und den Anforderungen ergibt sich eine Service-orientierten Architektur, wie in Abbildung 4-5 dargestellt. Dabei werden die Softwaredienste die im intelligenten Maschinenfenster zum Einsatz kommen in Basis- und Mehrwertdienste unterschieden. Die Eigenschaften der beiden Dienstarten werden nachfolgend ausführlicher erläutert. Eine Visualisierungskomponente bietet in abstrahierter Form alle zur Darstellung grafischer Inhalte benötigten Funktionalitäten an. Bei dieser Komponente handelt es sich im eigentlichen Sinne ebenfalls um einen Basisdienst, welcher auf Grund seiner besonderen Funktion jedoch separat im Diagramm aufgeführt ist. Um Dienste zur Laufzeit des Systems hinzufügen, ausführen, beenden und aktualisieren zu können, kommt eine Komponente zur Dienstverwaltung zum Einsatz. Um aus der Verbindung mehrerer Dienste einen Mehrwert generieren zu können, bedarf es einer Kommunikation zwischen diesen Diensten.

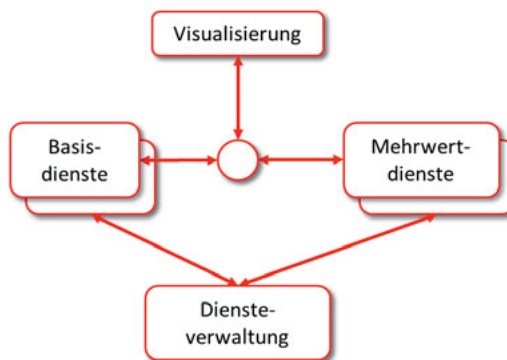


Abbildung 4-5: Systemarchitektur der Mehrwertdienste (Quelle: ISW)

4.2.1 Basisdienste

Wiederverwendbare Basisfunktionen werden in Basisdiensten gekapselt. Dadurch können einerseits häufiger benötigte Funktionalitäten ausgelagert und zentral vorgehalten werden. Andererseits führt die Segmentierung dieser Funktionalitäten in Basisdienste auch dazu, dass einzelne dieser Dienste unabhängig von anderen aktualisiert oder auf die jeweilige Maschine angepasst werden können. Basisdienste bringen auf Grund ihrer allgemeinen Funktionalität eigenständig keinen Mehrwert in das Gesamtsystem ein. Erst durch die Verwendung der

Funktionalität des Basisdienstes in einem Mehrwertdienst wird der tatsächliche Mehrwert für den Nutzer generiert.

Beispiele für Basisdienste sind Dienste zur Anbindung an die Maschinensteuerung (siehe Kapitel 5), zur Verwaltung und Kalibrierung von Kameras (siehe Kapitel 7.5), zum Parsen von Maschinenprogrammen (NC-Programme in G-Code; siehe Kapitel 3.5), zur Positionserkennung des Betrachters (siehe Kapitel 6.5.2) und zur Maschinensimulation (siehe Kapitel 8 und 9). Wie bereits genannt ist im eigentlichen Sinne auch die Visualisierungskomponente des intelligenten Maschinenfensters ein Basisdienst, da sie in abstrahierter Form alle zur Darstellung grafischer Inhalte benötigten Funktionalitäten wiederverwendbar anbietet.

4.2.2 Mehrwertdienste

Zur Generierung von Mehrwert für den Nutzer kommen Mehrwertdienste zum Einsatz. Diese Mehrwertdienste nutzen idealerweise einen oder mehrere Basisdienste und fügen zusätzliche vom zu erzielenden Mehrwert abhängige Funktionalität hinzu. Mit Hilfe dieser Dienste lässt sich die Funktionalität des intelligenten Maschinenfensters und damit der Werkzeugmaschine nachträglich erweitern.

Das Zusammenspiel aus einem Mehrwertdienst und mehreren Basisdiensten wird nachfolgen am Beispiel eines Dienstes zur Visualisierung der zurückliegenden Werkzeugbahn veranschaulicht. Wie in Abbildung 4-6 dargestellt, nutzt der Mehrwertdienst zur Bahnvisualisierung einerseits den Basisdienst zur Steuerungsanbindung und andererseits die Visualisierungskomponente. Der Basisdienst zur Steuerungsanbindung dient als Datenschnittstelle zur Maschinensteuerung und liefert unter anderem die aktuelle Position des Werkzeugmittelpunkts (Tool Center Point, TCP) sowie weitere Maschinendaten wie Vorschubgeschwindigkeit, aktive Bewegungsart oder Spindellast. Der Mehrwertdienst zur Bahnvisualisierung puffert die vergangenen TCP-Positionen in einer Liste. Aus den Punkten in dieser Liste wird eine dem zurückgelegten Weg des TCP entsprechende Linie generiert und diese mit fortschreitender Bewegung aktualisiert. Darüber hinaus kann diese Linie anhand weiterer Maschinendaten eingefärbt werden. Beispielsweise kann die Linie abhängig von der Geschwindigkeit eine andere Farbe annehmen oder durch zwei definierte Farben zwischen Eilgang und Bearbeitungsgeschwindigkeit unterschieden werden. Auf diese Weise entsteht eine einfach ersichtliche Zuordnung dieser bisher rein quantitativen Daten zur geometrischen Maschinenbewegung. Die abschließende Darstellung der vom Mehrwertdienst erzeugten Linien am Maschinenfenster erfolgt über die Basiskomponente zur Visualisierung. Unter den durch diesen Dienst zur Verfügung gestellten allgemeinen Visualisierungsfunktionen finden sich auch Funktionalitäten zum Zeichnen von Linien. Alle weiteren die korrekte Darstellung betreffenden Aufgaben wie perspektivische Darstellung und Verdeckung (siehe Kapitel 6) übernimmt die Visualisierungs-komponente eigenständig.

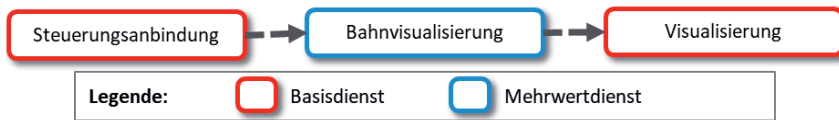


Abbildung 4-6: Exemplarisches Zusammenspiel mehrerer Dienste zur Schaffung von Mehrwert (Quelle: ISW)

4.2.3 Metadaten

Zur Beschreibung der Eigenschaften von Diensten werden weitere Informationen in Form von Metadaten gespeichert. Damit besteht ein Dienst aus der funktionalen Anwendung und den zugehörigen Metainformationen, dargestellt in Abbildung 4-7. Zu den Metadaten eines Dienstes zählen beispielsweise die Bezeichnung des Dienstes, eine verständliche textuelle Funktionsbeschreibung, Informationen zum Entwickler, Versionsinformationen und zur Ausführung erforderliche Zugriffsrechte. Die Beschreibung der Metadaten erfolgt in der sowohl menschen- wie auch maschinenlesbaren Auszeichnungssprache XML.



Abbildung 4-7: Zusammensetzung eines Dienstes (Quelle: ISW)

Anhand der zur Ausführung erforderlichen Zugriffsrechte ist bereits vor der Installation und Nutzung eines Dienstes bekannt, auf welche Basisdienste und Daten dieser Zugriff erhält. Diesen Zugriffsrechten muss der Nutzer, entsprechend der Verwendung von Smartphone-Apps, im Vorfeld ausdrücklich zustimmen. Beispielsweise kann mittels dieser Rechte geregelt werden, ob ein Dienst mittels Basisdienst zur Steuerungsanbindung aktuelle Maschinendaten lesen, einzelne Bewegungen der Maschine oder ganze Maschinenprogramme kommandieren und mittels Basisdienst zur Visualisierung virtuelle Objekte am Maschinenfenster darstellen darf. Nicht vorab angemeldeter und damit auch nicht ausdrücklich gestatteter Zugriff auf weitere Dienste und deren Daten wird im Kommunikationsmechanismus unterbunden (siehe Kapitel 4.3).

4.2.4 Gestaltung unabhängiger Dienste

Im Rahmen der Anforderungen wurde bereits erläutert, dass eine Service-orientierten Architektur sowie Sandboxing für einzelnen Dienste erforderlich ist. Mittels Sandboxing kann sowohl die Betriebs- (Safety) wie auch die Datensicherheit (Security) gewährleistet werden. Um es externen Entwicklern zu ermöglichen neue Dienste zu erstellen, ist es notwendig mittels Virtualisierung verschiedenen Systemumgebung und Programmiersprache zur Entwicklung zu unterstützen. Resultierend aus diesen Anforderungen sieht der für das

intelligente Maschinenfenster gewählte Lösungsansatz eine Virtualisierung und Kapselung der einzelnen Dienste in sogenannten Containern vor. Die Firma Docker, die die gleichnamige Container-Plattform anbietet, beschreibt Container wie folgend:

“A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.” [7]

Dienste für das intelligente Maschinenfenster können als eigenständige Anwendung, nahezu ohne Einschränkungen entwickelt werden. Jede dieser Anwendungen wird anschließend in einem Container gekapselt. Dabei beinhaltet der Container alle zur Ausführung der Anwendung notwendigen Daten wie beispielsweise Laufzeitumgebung, Bibliotheken oder Einstellungen, vergleiche Abbildung 4-8. Die Verwendung verschiedener Laufzeitumgebungen, z.B. das Java Runtime Environment oder die Common Language Runtime im .NET Framework, und Programmiersprachen innerhalb eines Containers wird unterstützt. Ein Container isoliert die enthaltene Anwendung von anderen Container und vom zugrundeliegenden Hostsystem.

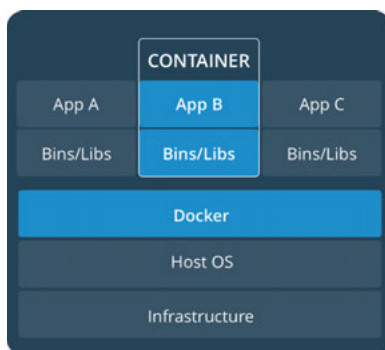


Abbildung 4-8: Container-Infrastruktur [7]

Um die in Container gekapselten Dienste zur Laufzeit des intelligenten Maschinenfensters installieren und ausführen zu können bedarf es einer Komponente zur Verwaltung der Container. Diese Verwaltungskomponente wird im Rahmen dieses Projekts als Mehrwertdienst Manager bezeichnet, dargestellt in Abbildung 4-9. Auf dem lokalen System werden Container in einem Repository als Images vorgehalten. Vor der erstmaligen Ausführung wird in der Docker Runtime aus einem solchen Image eine Instanz, der eigentliche Container, abgeleitet. Diese Instanz kann nun gestartet und bei Bedarf auch wieder gestoppt werden. Neben der Durchführung dieser Aktionen ermöglicht die Verwaltungskomponente auch die Vergabe von Zugriffsrechten auf das System und auf Systemressourcen (z.B. Netzwerk, Speicher, usw.). Realisiert werden diese Funktionen mittels eine Programmierschnittstelle (API) der Docker Container-Plattform.

Zur Realisierung einer zentralen Vertriebsplattform in Form eines Mehrwertdienststores wird ein weiteres, globales Repository verwendet. Dieses Repository wird auf dem Server des Mehrwertdienststores bereitgestellt und beinhaltet die Images aller im Store verfügbaren Mehrwertdienste. Durch Aufspielen neuer Dienste in den Mehrwertdienststore ergibt sich für externe Entwickler die Möglichkeit ihre Dienste einem großen Kundenkreis zur Verfügung zu stellen. Soll ein neuer Mehrwertdienst auf dem intelligenten Maschinenfenster genutzt werden, kann dieser mittels der Website der Vertriebsplattform gefunden, vom globalen auf das lokale Repository herunter-geladen und anschließend genutzt werden.

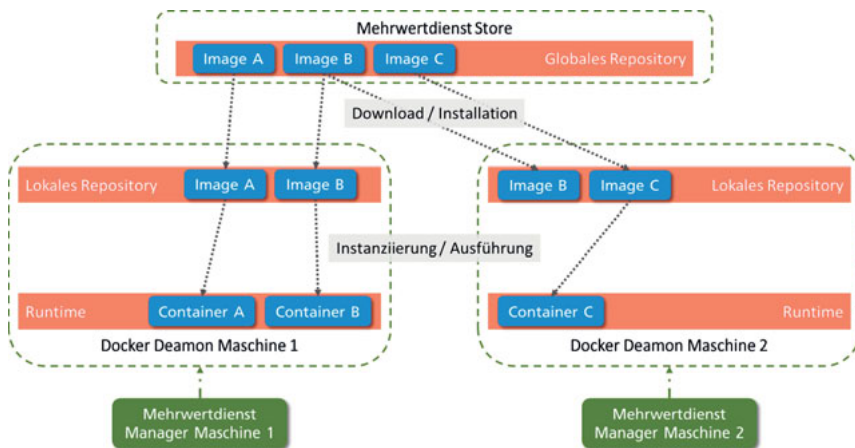


Abbildung 4-9: Verwaltung von Containern (Quelle: ISW)

4.3 Kommunikation zwischen Diensten

Um die in Container gekapselten Dienste miteinander zu verbinden bedarf es einer Kommunikation zwischen diesen. Zur Realisierung dieser Kommunikation kommen mehrere Kommunikationsmodelle in Frage, wobei im Folgenden nur das (mehrstufige) Client/Server-Modell und das (zentralisierten) Peer-to-Peer-Modell anhand einiger relevanter Eigenschaften einander gegenübergestellt werden. In Abbildung 4-10 ist je ein Beispiel beider Kommunikationsmodelle mit den Basisdiensten Steuerungsanbindung und Visualisierung, sowie drei beliebigen Mehrwertdiensten dargestellt.

Im Client/Server-Modell erfolgt eine üblicherweise synchrone d.h. blockierende Kommunikation zwischen einem Client und einem Server. Der Server stellt hierbei eine Funktionalität zur Verfügung, die von einem oder mehreren Clients genutzt werden kann. Im mehrstufigen Client/Server-Modell kann sich ein Server zur Erbringung seiner Funktionalität wiederum der Funktionalität anderer Server bedienen. [8] Dieses Modell führt dazu, dass für jede Verknüpfung zwischen zwei Diensten eine neue Verbindung aufgebaut wird (1 zu 1). Um

eine solche direkte Verbindung aufzubauen muss das Ziel d.h. der Empfänger der Daten bekannt und erreichbar sein. Des Weiteren sind mit jedem Verbindungsaufbau eine neue Authentifizierung und damit auch eine Autorisierung des Datenzugriffs notwendig. Werden Daten von einem Dienst an viele andere weitergegeben, führt dies zum mehrfachen Senden derselben Daten über verschiedene Verbindungen.

Dem gegenüber steht das Peer-to-Peer-Modell, bei dem mehrere Dienste gleichberechtigt in einem Netzwerk miteinander kommunizieren. Ein Dienst kann gleichermaßen Daten senden wie auch empfangen und damit sowohl die Rolle eines Clients wie auch eines Servers einnehmen. Auf diese Weise wird die klassische Client/Server-Rollenverteilung aufgelöst. Der Datenaustausch erfolgt hierbei üblicherweise asynchron, was zu Vorteilen hinsichtlich Parallelisierbarkeit und Entkopplung der Kommunikationspartner führt. Das asynchrone Versenden von Daten ähnelt dem Prinzip der postalischen Zustellungen von Briefen, weswegen man in diesem Fall üblicherweise vom Versenden von Nachrichten (engl. Messaging) spricht. Grundsätzlich entfällt im Peer-to-Peer-Modell die zentrale Struktur, oft kommt jedoch zum Verwalten und Auffinden von Ressourcen ein zentraler Vermittler, der Broker, zum Einsatz. [8] Da in diesem Kommunikationsmodell jeder Dienst nur eine einzige Datenverbindung, über die alle anderen Dienste angesprochen werden können (1 zu n), zum Broker herstellt, erfolgen auch nur eine einmalige Authentifizierung und Autorisierung. Die nachträgliche Anbindung an weitere Dienste und damit zusätzliche Daten erfolgt über diese Verbindung und erfordert daher weder einen neuen Verbindungsaufbau noch die erneute Vergabe von Zugriffsrechten. Das Ziel von ausgehenden bzw. die Quelle ankommender Daten muss auf Grund der losen Kopplung der Dienste nicht bekannt sein. Während es beim Client/Server-Modell zu einem Abbruch der direkten Verbindung zwischen zwei Diensten kommen kann, kann im Peer-to-Peer-Modell lediglich eine Verbindung zwischen Dienst und Broker abreißen. Nicht erreichbaren Datenquellen und -senken führen nicht zu Problemen, da entsprechende Nachrichten entweder gepuffert oder verworfen werden. Beim Senden von Daten von einem Dienst an viele andere, werden die Daten nur einmalig zum Broker verschickt und von dort vergleichbar mit einem Broadcast an die Empfänger verteilt.

Da für eine Service-orientierte Architektur mit loser Kopplung die Vorteile des Peer-to-Peer-Modells überwiegen, wurde dies Modell für das Framework für Mehrwertdienste ausgewählt.

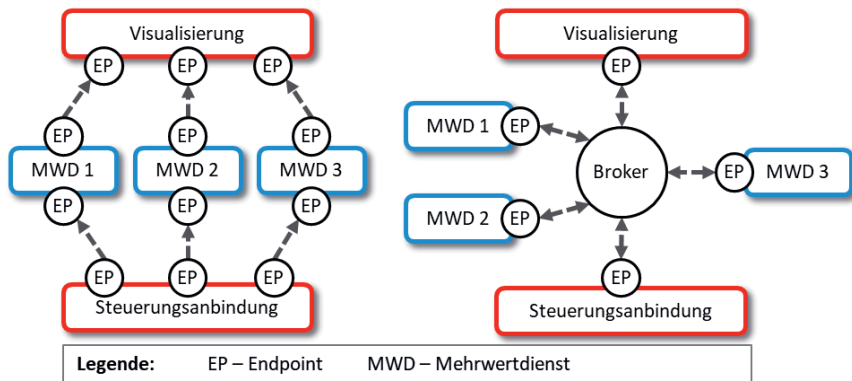


Abbildung 4-10: Kommunikationsmodelle; Links: Client/Server-Modell; Rechts: Peer-to-Peer-Modell (Quelle: ISW)

Nachrichten im Peer-to-Peer-Modell können auf verschiedene Arten ausgetauscht werden. Im Folgenden werden die in Abbildung 4-11 dargestellten Messaging-Arten Request/Reply und Publish/Subscribe (Pub/Sub) erläutert.

Das Request/Reply-Modell orientiert sich an der klassischen Kommunikation im Client/Server-Modell, erhält jedoch die Vorteile des Peer-to-Peer-Modells. Ein Dienst versendet hierbei eine Anfrage (Request) in Form einer einzelnen Nachricht, z.B. eine Datenanfrage oder einen Auftrag. Ein Empfänger kann diese Anfrage lesen, bearbeiten und falls erforderlich mit einer eigenen Nachricht antworten (Reply). Um trotz der asynchronen Kommunikation eine Zuordnung einer Antwort zur ursprünglichen Anfrage zu erreichen, wird mit der Anfrage eine Korrelationskennung (Korrelations-ID) übertragen und die resultierende Antwort ebenfalls mit dieser gekennzeichnet. Je nach Anwendungsfall können durch mehrere konkurrierende Empfänger eine Lastverteilung oder durch mehrere simultane Empfänger eine Abfrage mit mehreren Antworten erreicht werden. Durch die lose Kopplung im Peer-to-Peer-Modell müssen weder der bzw. die tatsächlichen Empfänger noch deren Anzahl bekannt sein. Allerdings muss darauf geachtet werden, dass im Falle nicht vorhandener Empfänger gegebenenfalls nie eine Antwort erhalten wird.

Das Publish/Subscribe-Modell ist auf die Verteilung von Nachrichtenströmen ausgelegt. Ein oder mehrere Publisher senden Nachrichten an einen definierten Kanal und erzeugen auf diese Weise einen Nachrichtenstrom [9]. Hierbei kann es sich sowohl um zyklische, kontinuierliche (z.B. Positionsdaten) wie auch um sporadische, nicht kontinuierliche Daten (z.B. Ereignisse) handeln. Dem gegenüber können Empfänger den Erhalt dieser Nachrichten abonnieren. Dabei ist es auf Grund der losen Kopplung für den Sender bzw. den Empfänger irrelevant ob aktuell überhaupt ein Empfänger bzw. ein Sender vorhanden ist.

Zur Kommunikation zwischen den Diensten des intelligenten Maschinenfensters kommen je nach Art der übertragenen Daten sowohl das Request/Reply- wie auch das Publish/Subscribe-Modell zum Einsatz.

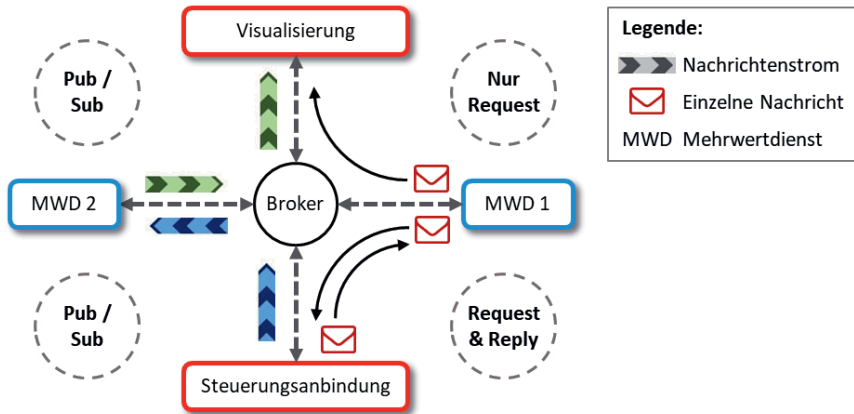


Abbildung 4-11: Verschiedene Messaging-Arten (Quelle: ISW)

Wie im Rahmen der Anforderungen bereits erläutert, muss der Zugriff von Mehrwertdiensten auf Basisdienste und deren Daten geregelt werden. Dazu werden die zur Ausführung eines Dienstes erforderlichen Zugriffsrechte in den Metadaten des Dienstes beschreiben. Bei der Installation eines neuen Mehrwertdienstes wird der Anwender über diese Zugriffsrechte informiert und muss diesen ausdrücklich zustimmen. Stimmt der Anwender den erforderlichen Rechten nicht zu, ist eine Nutzung des Dienstes nicht möglich. Wird ein Mehrwertdienst gestartet, autorisiert der Mehrwertdienst Manager den erforderlichen Zugriff entsprechend dieser Zugriffsrechte. Im Beispiel in Abbildung 4-12 erlaubt die Komponente zur Dienstverwaltung, d.h. der Mehrwertdienst Manager, dem gestarteten Mehrwertdienst nur den Zugriff auf die Basisdienste zur Steuerungsanbindung und zur Visualisierung jedoch nicht auf die Basisdienste zum Parsen von NC-Programmen und zur Maschinensimulation.

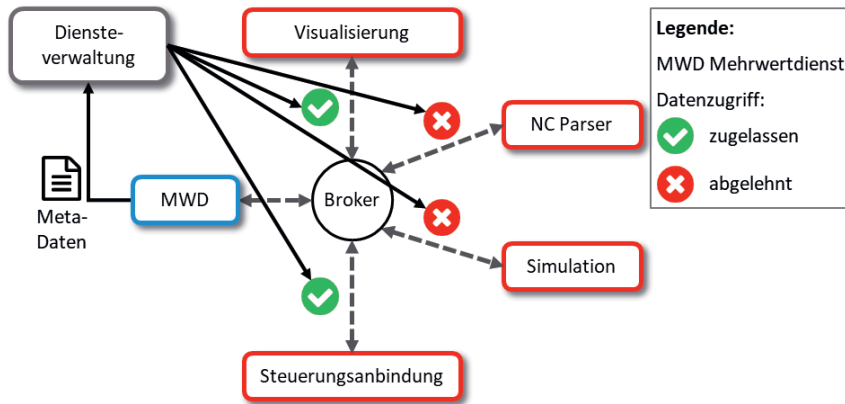


Abbildung 4-12: Verwaltung von Zugriffsrechten (Quelle: ISW)

4.4 Literaturverzeichnis

- [1] Bundesagentur für Arbeit: *Maschinen- und Anlagenführer/in*. 2017
- [2] Bundesagentur für Arbeit: *Zerspanungsmechaniker/in*. 2017
- [3] Bundesagentur für Arbeit: *CNC-Fachkraft bzw. NC-Anwendungsfachmann/-frau*. 2016
- [4] Bauernhansl, Thomas ; Krüger, Jörg ; Reinhart, Gunther ; Schuh, Günther: *WGP-Standpunkt Industrie 4.0*. Berlin, 2016
- [5] Wahbe, Robert ; Lucco, Steven ; Anderson, Thomas E. ; Graham, Susan L.: *Efficient Software-Based Fault Isolation*. In: *Proceedings of the Symposium on Operating System Principles* (1993)
- [6] Goldberg, Ian ; Wagner, David ; Thomas, Randi ; Brewer, Eric: *A Secure Environment for Untrusted Helper Applications : Confining the Wily Hacker*. In: *Proceedings of the Sixth USENIX UNIX Security Symposium*, 1996
- [7] Docker Inc.: *What is a container : A standardized unit of software*. URL <https://www.docker.com/what-container> – Überprüfungsdatum 2018-01-02
- [8] Schill, Alexander ; Springer, Thomas: *Verteilte Systeme : Grundlagen und Basistechnologien*. 2. Aufl. Berlin, Heidelberg : Springer Vieweg, 2012
- [9] Curry, Edward: *Message-Oriented Middleware*. In: Mahmoud, Qusay H. (Hrsg.): *Middleware for Communications*. Chichester, England : John Wiley & Sons, 2004, S. 1–28

5 Datenmodell und Steuerungsanbindung über OPC UA

S. Müller, J. Brühl

5.1 Einleitung

Viele NC-Steuerungen unterschiedlicher Hersteller sind heute auf dem Markt erhältlich. Diese verfügen über eine Vielzahl proprietärer Schnittstellen, die den Zugriff auf die Steuerungsdaten ermöglichen. Bis jetzt hat sich bei der externen Kommunikation über ein Netzwerk kein einheitlicher Standard etabliert, über den ein Austausch von Steuerungsdaten vollzogen werden kann. Insbesondere für die Anforderung an eine vernetzte Fabrik, wie sie in der Forschungsinitiative *Industrie 4.0* gefordert wird, ist eine einheitliche Steuerungsanbindung über eine standardisierte Kommunikationstechnologie notwendig. Die Kommunikationstechnologie OPC UA stellt einen vielversprechenden Standard dar, welcher die heutigen und zukünftigen Anforderungen erfüllen könnte. Im Zusammenhang mit *Industrie 4.0* wird OPC UA als wesentlicher Bestandteil der Informations- und Kommunikationsschicht angesehen [1]. Ein wesentlicher Vorteil von OPC UA gegenüber vorherigen Standards ist dessen Metamodell (OPC UA Adressraum), welches die Beschreibung von beliebigen Informationsmodellen ermöglicht. Dieses kann auch die Grundlage für herstellerübergreifende Standards für die Steuerungsanbindung einer CNC-Steuerung sein. Im Rahmen des Forschungsprojekts *iWindow* war es das Ziel, eine herstellerübergreifende Steuerungsanbindung prototypisch zu evaluieren.

5.2 Ausgangslage und Problemstellung

In der Vergangenheit gab es mehrere Initiativen mit dem Ziel, eine einheitliche Schnittstelle und Anbindung für CNC-Steuerungen zu definieren [2]. Ein Ergebnis einer europäischen Initiative war OSACA [3]. Dieses beschreibt eine offene Steuerungsarchitektur mit einer standardisierten Schnittstelle. Der Umfang der OSACA Spezifikation war nicht nur eine standardisierte Schnittstelle für den externen Zugriff auf die Steuerungsdaten, sondern die Beschreibung einer Referenzarchitektur für die gesamte modulare NC-Steuerung. Obwohl deren Anwendbarkeit auf mehreren Steuerungen verschiedener Hersteller erprobt wurde [4], konnte sich der OSACA Standard im industriellen Umfeld nicht durchsetzen. Vielmehr haben sich Industriestandards wie OPC [5] etabliert, die eine standardisierte Kommunikationsschnittstelle für die Steuerungen zur Verfügung stellen. Jedoch zeigt die Umsetzung dieser Kommunikationsschnittstellen wesentliche Unterschiede in Semantik und Syntax. Dies führt zum Beispiel dazu, dass die Steuerungsparameter gleichartiger NC-Funktionen nicht herstellerunabhängig interpretiert und abgefragt werden können. Neben der herstellerspezifischen Benennung der Steuerungsparameter sind deren einzelnen Parameter proprietär zu interpretieren. Dies macht eine Abbildung herstellerspezifischer Schnittstellen aufeinander im Prinzip unmöglich. Dieses Problem kann nur mit

standardisierten Daten- und Informationsmodellen gelöst werden, welche die Syntax der Steuerungsparameter festlegt und deren Semantik eindeutig beschreibt.

Heutige CNC-Steuerungen verarbeiten eine Vielzahl von Daten und steuern eine Vielzahl von Funktionen neben der klassischen Steuerung des Verarbeitungsprozesses von NC-Programmen in bis zu 60 Kanälen. Unter anderem unterstützen diese SPS-Programme, verarbeiten Felddaten, bieten ein Dateisystem und ein elektronisches Typenschild oder geben Diagnosemeldungen aus. Es ist zu erwarten, dass der Umfang an Daten und Funktionen auch in Zukunft weiter zunehmen wird. Auf der anderen Seite steigt die Anzahl der Interessenten und Konsumenten dieser Daten in letzter Zeit deutlich. Vor allem mit der vernetzten Fabrik ist ein deutlicher Anstieg an Konsumenten zu erwarten. Diese umfassen neben den klassischen HMI auch weitere Konsumenten wie zum Beispiel Mehrwertdienste, Leitsysteme (MES/BDE-Systeme oder ERP-Systeme) oder neue Clouddienste. Nicht nur die Anzahl der Konsumenten steigt, sondern auch deren Anforderungen. Dies sind zum Beispiel Echtzeitdaten für eine Abtragsimulation, Überwachung und Verarbeitung von Prozessen mit Hilfe von Steuerungsdaten oder das umfangreiche Sammeln von Steuerungsdaten über Analysen (Big Data).

Anhand der Ausgangslage lässt sich eine Reihe an Problemstellungen identifizieren. Proprietäre Schnittstellen weisen gegenüber standardisierten Schnittstellen höhere Entwicklungskosten auf. Diese äußern sich nicht nur bei der Entwicklung der eigentlichen Schnittstellen, sondern insbesondere auch bei der Entwicklung von Anwendungen unter Nutzung dieser Schnittstellen. Gründe hierfür sind zum Beispiel, dass keine Open-Source- oder Drittanbieter-Lösungen vorhanden sind, die kostengünstig eingesetzt werden können. Zudem ist mit erhöhten Supportaufwänden auf Seiten des Steuerungsherstellers zu rechnen, da diese Schnittstelle exklusiv von ihm angeboten wird. Im Allgemeinen haben proprietäre Schnittstellen eine geringere Kundenakzeptanz gegenüber herstellernunabhängigen Lösungen. Auf Seiten der Werkzeugmaschinenhersteller ergeben sich aus der Nutzung einer proprietären Schnittstelle der Steuerung eine technologische Abhängigkeit gegenüber einem bestimmten Steuerungshersteller. Auch wird der Einsatz von Steuerungen mehrerer unterschiedlicher Hersteller erheblich erschwert. Für den Werkzeugmaschinenbetreiber ist unter Umständen die Kombination von mehreren Werkzeugmaschinen verschiedener Hersteller schwieriger. Dies gilt insbesondere bei dem Wunsch nach einer vernetzten Fabrik, bei der die horizontale als auch vertikale Integration der Werkzeugmaschine und deren Steuerung im Gesamtfeld der Fabrik gewünscht wird. Auch wird die Nutzung von Hard- und Software eines Drittanbieters stark eingeschränkt, da Drittanbieter vor allem Lösungen für weitverbreitete Schnittstellen anbieten werden. Der Werkzeugmaschinenbetreiber wird sich in dieser Situation auf wenige Hersteller konzentrieren müssen, die auf den gleichen Steuerungshersteller setzen.

Zusammengefasst lässt sich eine Vielzahl von Nachteilen erkennen, die sich durch die Nutzung einer proprietären Schnittstelle ergeben. Es gibt für alle Seiten wesentliche Vorteile bei der Verwendung einer standardisierten herstellerübergreifenden Schnittstelle zum Zugriff auf Steuerungsparameter und -funktionen. Dieser Vorteil wird sich insbesondere auch mit dem Trend der Vernetzung in der Fabrik und der immer mehr aufkommenden Idee von Plug-and-produce, angelehnt am Plug-and-play aus der Computer- und Unterhaltungselektronikbranche, verstärken.

Das Forschungsprojekt *iWindow* bietet hierbei ideale Bedingungen für die Erprobung dieser Schnittstellen. Mehrere Werkzeugmaschinenhersteller sind am Projekt beteiligt und eine Vielzahl an Anwendungen und Mehrwertdiensten sind vorgesehen, die von den verschiedenen Projektpartner umgesetzt werden.

5.3 Datenmodell

5.3.1 Grundlagen OPC UA Adressraum

OPC UA bietet die Möglichkeit Daten und Funktionen syntaktisch und semantisch zu beschreiben. Grundlage hierfür ist der OPC UA Adressraum, welcher das Metamodell von OPC UA darstellt. Im OPC UA Adressraum können beliebige Informationsmodelle in Form von verschiedenen Knotentypen und Referenztypen beschrieben werden [6]. Hierzu stehen die in Abbildung 5-1 gezeigten Elemente zur Verfügung. Die Beschreibung des Informationsmodells erfolgt objektorientiert. Es kann zwischen Instanz- und Typmodell unterschieden werden. Diese sind vergleichbar mit Objekt- und Klassenmodell aus UML. Ein Objekt besteht aus Variablen und Methoden und stellt eine Instanz eines Objekttypen dar. Die einzelnen Knoten sind über Referenzen miteinander verbunden und beschreiben die Beziehung der Knoten zueinander. In einem Informationsmodell werden Instanz- und Typmodell detailliert beschrieben und legen die semantische Bedeutung fest. Im OPC UA Adressraum kann man die unterschiedlichen Informationsmodelle an den unterschiedlichen Namensräumen erkennen.

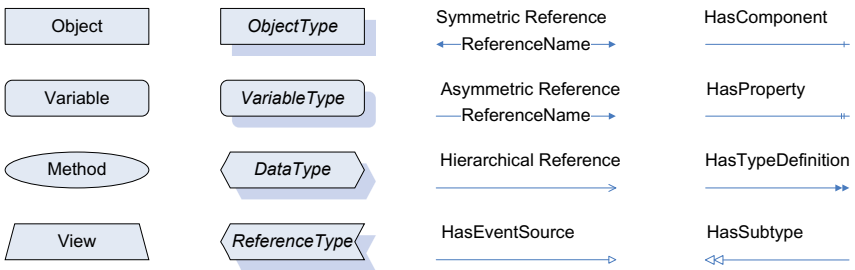


Abbildung 5-1: Grafische Darstellung der Elemente des OPC UA Adressraums (Quelle: OPC UA Spezifikation – Part 3)

5.3.2 Informationsmodelle für CNC-Steuerungen

Heutige CNC-Steuerungen bieten eine Vielzahl unterschiedlicher Daten und Funktionen, deren Beschreibung in unterschiedlichen Informationsmodellen unterteilt werden muss. Hierbei muss auch der unterschiedliche Funktionsumfang der Steuerungen beachtet werden. Es werden zum Beispiel unterschiedliche Feldbusse von den verschiedenen Steuerungen unterstützt oder moderne CNC-Steuerungen beinhalten neben den NC-spezifischen Funktionen auch SPS-Funktionalitäten als Anpasssteuerung. Im Rahmen des iWindow-Projekts wurde das CNC-System MTX eingesetzt und in diesem Zusammenhang betrachtet. In Abbildung 5-2 ist auf der linken Seite ein Ausschnitt aus dem Adressraum der MTX dargestellt und auf der rechten Seite der Abbildung ist beschrieben, für welche Daten die gezeigten UA Elemente die Einstiegspunkte darstellen. Einen großen Teil des gezeigten Adressraums nehmen die CNC-spezifischen Daten ein. Das sind Daten und Funktionen, welche innerhalb des CNC-Steuerungskerns verarbeitet bzw. bereitgestellt werden. Darunter fallen zum einen die kanalabhängigen Daten und Befehle mit Korrektur-, Vorschub- und Positionswerten als auch Programm-, Geschwindigkeits- und Spindelraten. Zum anderen sind darin aber auch Achsdaten, eine Bitschnittstelle zwischen SPS und CNC, CNC-spezifische Befehle und persistente Nutzerdaten zu finden. Das hierbei genutzte Informationsmodell ist von Bosch Rexroth definiert und herstellersistezifisch. Jedoch existieren im Adressraum der CNC-Steuerungen auch noch weitere nicht CNC-spezifische Daten und Funktionen, für die es aber bereits standardisierte OPC UA Informationsmodelle gibt. Für das Typenschild einer Werkzeugmaschine ist die Device Companion Specification [7], für die Diagnose-Informationen eine Erweiterung des OPC UA A&C Standards [8], für das Dateisystem eine Erweiterung des OPC UA Informationsmodells aus Teil 5 der OPC UA Spezifikation[9], für SPS-Daten die PLC Open Companion Specification [10] und für Sercos Feldbus-Daten die Sercos OPC UA Companion Specification [11] definiert. Obwohl die Bosch Rexroth Steuerung für einzelne Domänen entsprechende standardisierte Informationsmodelle nutzt und befolgt, können auf Grund der Vererbungsmechanismen in OPC UA noch herstellersistezifische Erweiterungen darin untergebracht werden. Bei den genannten Erweiterungen handelt es sich um herstellersistezifische Funktionalitäten der Bosch Rexroth Steuerung, welche über die in den Standards beschriebene Funktionalitäten hinausgeht. Ein Client, der nur entsprechende standardisierte Informationen kennt, erhält trotz der Erweiterungen alle Informationen, die er erwartet. Die zusätzlichen herstellersistezifischen Daten können hingegen nur von Clients interpretiert werden, welche die entsprechenden herstellersistezifischen Erweiterungen kennen. Somit bietet OPC UA die idealen Voraussetzungen für eine Kommunikationsschnittstelle, bei der standardisierte und herstellersistezifische Daten koexistent sind. Für Steuerungshersteller bedeutet das, sie können eine herstellersistezifische Kommunikationsschnittstelle anbieten, jedoch diese jederzeit mit herstellersistezifischen Daten und Funktionen erweitern, ohne die bisherigen herstellersistezifischen Daten und Funktionen einzuschränken.

Für die CNC-spezifischen Daten stand zu Beginn des Projektes noch kein standardisiertes OPC UA Informationsmodell in einer sogenannten OPC UA Companion Spezifikation zur Verfügung. Ziel war es deshalb, sich während des Projektzeitraums an der OPC UA VDW Companion Specification zu beteiligen und diese prototypisch zu evaluieren. Die Evaluierung ist in Kapitel 5.4.1 beschrieben.

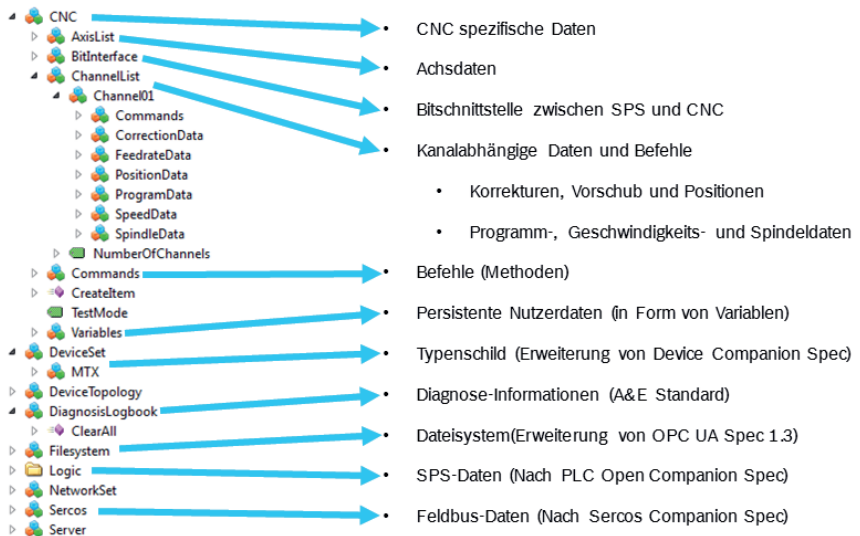


Abbildung 5-2: OPC UA Adressraum des CNC-Systems MTX – Einstiegspunkte für die Informations- und Datenmodelle der Steuerung (Quelle: Bosch Rexroth AG)

5.3.3 OPC UA VDW Companion Specification

In einer gemeinsamen Arbeitsgruppe von der OPC Foundation und dem VDW (Verein Deutscher Werkzeugmaschinenfabriken) wurde ein OPC UA Informationsmodell für CNC-Systeme erarbeitet und spezifiziert. Die Companion Specification [12] wurde im Juni 2017 veröffentlicht.

Der obere Abschnitt von Abbildung 5-3 zeigt das definierte OPC UA Informationsmodell für CNC-Systeme mit seinen *ObjectTypes* als Repräsentanten der einzelnen Komponenten eines CNC-Systems. Als oberster Einstiegspunkt ist dabei der *CncInterfaceType* gedacht, welcher die kompletten Komponenten unter sich vereint. Als Komponenten sind Kanäle (*CncChannelListType*, *CncChannelType*), Spindel (*CncSpindleListType*, *CncSpindleType*) und Achsen (*CncAxisListType*, *CncAxisType*) angedacht. Des Weiteren werden Alarme (*CncAlarmType*) und Meldungen (*CncMessageType*) für Diagnoseinformationen unterstützt. Im unteren Abschnitt von Abbildung 5-3 ist ein Ausschnitt eines beispielhaften Adressraums für ein 5-Achs-Bearbeitungszentrum wie der Projektmaschine DMU 50 ecoMill der Firma DMG MORI dargestellt. Diese besteht aus 5 Achsen mit zusätzlicher Spindel, welche die Bearbeitung eines NC-Programms in einem Kanal ermöglicht.

Als grundlegende Architektur für das CNC-System wurde das in Abbildung 5-4 gezeigte CNC-System angenommen, welches als Schnittstelle für eine Nicht-Echtzeitkommunikation einen OPC UA Server mit dem entsprechenden spezifizierten Informationsmodell bietet. Als Alternative kann auch der OPC UA Server auf einem PC installiert werden, der über eine herstellerspezifische Schnittstelle an die CNC-Steuerung angebunden ist. Die MTX verfügt über eine entsprechende Architektur und bietet bereits einen OPC UA Server als Bestandteil der Steuerungsfunktion. Deshalb eignet sich diese ideal für die prototypische Umsetzung des Informationsmodells. Dies wird in 5.4.1 beschrieben.

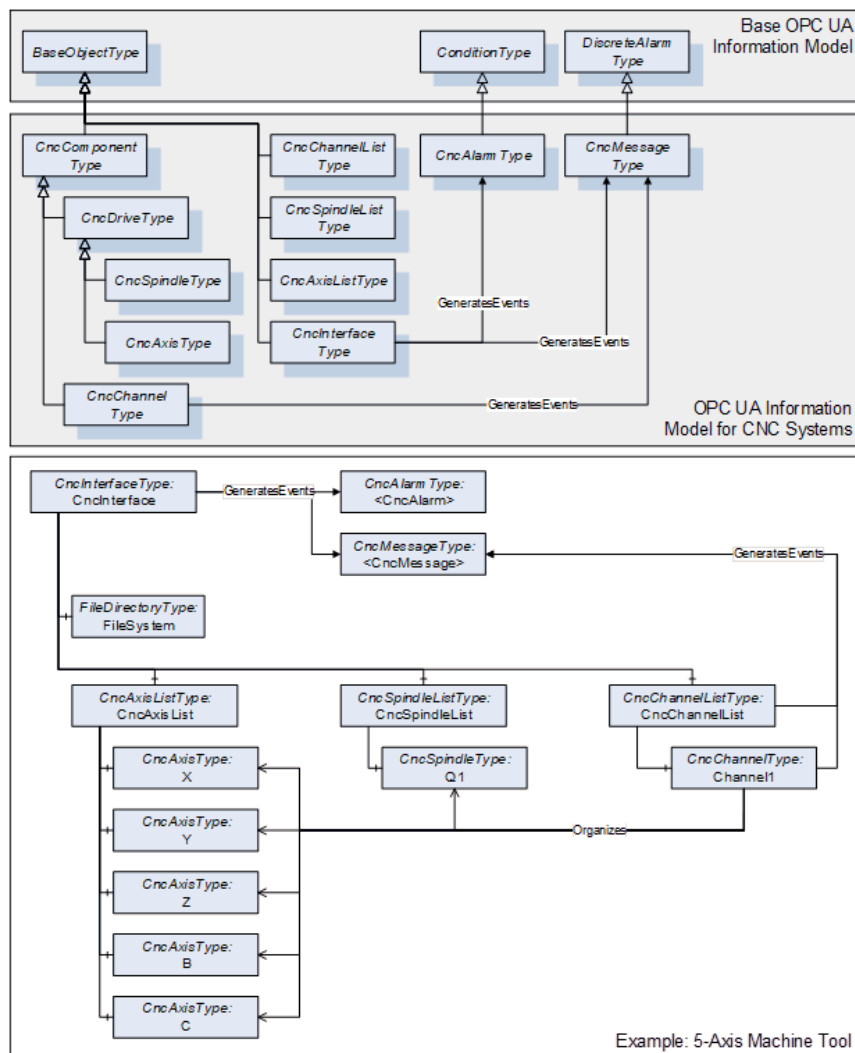


Abbildung 5-3: Oben: Übersicht über die definierten ObjectTypes im OPC UA Informationsmodell für CNC-Systeme; Unten: Beispiel einer 5-Achs-Werkzeugmaschine [12]

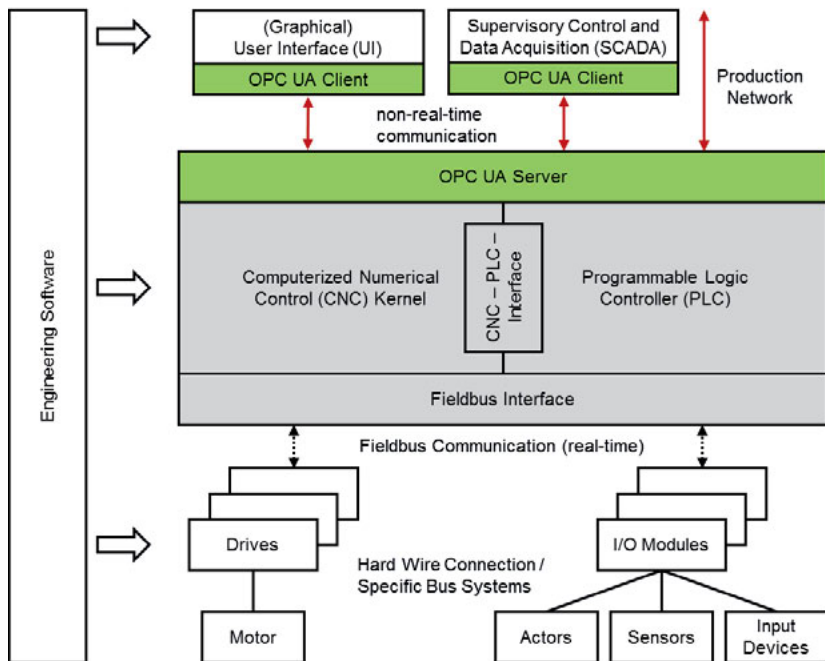


Abbildung 5-4: Grundlegende Systemarchitektur für die Realisierung des OPC UA Informations-modells für CNC-Systeme [12]

5.4 Steuerungsanbindung über OPC UA

Das im Rahmen des Projekts iWindow eingesetzte CNC-System MTX von Bosch Rexroth verfügt bereits über einen OPC UA Server. Dieser soll die Steuerungsanbindung für die verschiedenen in iWindow erprobten Applikationen ermöglichen. Ein Ziel ist es, eine herstellerübergreifende standardisierte Kommunikationsschnittstelle für die Steuerungsanbindung zur Verfügung zu stellen. Im Bereich der CNC-Steuerungsdaten stellt das in 5.3.3 beschriebene OPC UA Informationsmodell für CNC-Systeme einen vielversprechenden Ansatz dar, dessen Evaluierung nachfolgend beschrieben ist.

Eine besondere Anforderung im Rahmen des iWindow-Projektes war die Bereitstellung von Prozessdaten aus dem Echtzeitbereich der Steuerung (nachfolgend Echtzeitdaten genannt). Das CNC-System MTX bietet hierzu in seinem OPC UA Server ein spezielles Verfahren, welches es erlaubt, den OPC UA Standarddienst Subscription zu nutzen. Dies wird in 5.4.2 beschrieben.

5.4.1 OPC UA VDW Companion Specification – Evaluierung mit Prototypen

Die OPC UA VDW Companion Specification „OPC UA Information Mode for CNC Systems“ wurde prototypisch auf dem OPC UA Server des Rexroth CNC-Systems MTX umgesetzt. Zur

Evaluierung wurde als Clientapplikation die im Projekt verwendete Standard-HMI IndraWorks Operation verwendet.

Als besonders vorteilhaft bei der Umsetzung erwies sich die Berücksichtigung der heutigen Systemarchitekturen im Bereich der Werkzeugmaschinen im Informationsmodell. Somit waren alle grundlegenden Elemente im Informationsmodell vorhanden, die eine entsprechende Umsetzung innerhalb des OPC UA Servers der MTX auf den obersten Ebenen sehr einfach machten. Instanzen zu den im Informationsmodell beschriebenen ObjectTypes für die Komponenten Kanäle, Achsen und Spindel konnten ganz einfach abgebildet werden. Jedoch auf den unteren Ebenen erwies sich die Umsetzung als schwierig. Dort werden die Eigenschaften und Zustände der einzelnen Komponenten (ObjectTypes) durch Variablen (Instanzen von VariablesTypes) beschrieben. Insbesondere die Anbindung an die internen Schnittstellen der Steuerung ist hierbei aufwendig. Gründe hierfür sind zum einen, dass die im Informationsmodell beschriebenen Daten und Funktionen in dieser Weise nicht in der Steuerung direkt vorliegen und somit Erweiterungen in den internen Schnittstellen des Steuerungskerns notwendig sind. Der andere Grund ist eine sehr vereinfachte oder unvollständige Beschreibung einzelner Datenelemente oder Funktionen im Informationsmodell. Insbesondere fehlen noch genauere Beschreibungen der einzelnen Prozesse und die dazugehörigen Zustände und lassen einen Interpretationsspielraum. Insgesamt hat dies zur Folge, dass das von verschiedenen Steuerungsherstellern umgesetzte Informationsmodell zwar syntaktisch korrekt ist, jedoch semantisch einzelne UA Datenelemente unterschiedlich interpretiert werden. Somit entstehen wieder inkompatible herstellerspezifische Lösungen.

Der Umfang der aktuellen Version des OPC UA Informationsmodells für CNC-Systeme reicht aus, um einfache Benutzeroberflächen für Werkzeugmaschinen zu realisieren, bei denen die wichtigsten Kennwerte (Achsen-, Spindel-, Kanal-, Programm- und Vorschubdaten) der Werkzeugmaschine angezeigt werden können. Jedoch kann dadurch noch keine umfangreiche Bedienung der Werkzeugmaschine erreicht werden, wie dies zurzeit zum Beispiel mit IndraWorks Operation möglich ist. Neben den Defiziten bei den Positions- und Geschwindigkeitsdaten fehlen beim jetzigen Informationsmodell insbesondere Korrektur- und Werkzeugdaten. Für die Bedienung der Werkzeugmaschine müssten auch entsprechende Befehle und Methoden definiert werden. Ebenfalls fehlt eine Beschreibung der Schnittstelle zwischen CNC und SPS. Eine allgemeine Konfigurationsschnittstelle und die Möglichkeit der Definition von persistenten Nutzerdaten würden das Informationsmodell abrunden.

5.4.2 Echtzeitdaten über OPC UA

Mit der Fähigkeit, Daten aus dem Echtzeitbereich des Steuerungskerns über die OPC UA Kommunikationsschnittstelle bereitzustellen, bietet die MTX einen wesentlichen Mehrwert für zahlreiche Anwendungen innerhalb des iWindow-Projekts. Diese Funktion wird unter

anderem bei den in den Kapiteln 2, 6.3, 8 und 9 beschriebenen Demonstratoren und Technologien verwendet.

Die Daten aus dem Echtzeitbereich werden über eine OPC UA Variable vom in Abbildung 5-5 gezeigten *ConsistentStructureDataType* im Adressraum des OPC UA Servers bereitgestellt. Der *ConsistentStructureDataType* beschreibt eine OPC UA Struktur, welche neben dem Kanal, Achspositionen, Programmnamen und Programmposition auch Werkzeugkorrekturdaten, Kanalstatus und individuelle Systemdaten beinhaltet. Eine Besonderheit der Struktur ist die Nutzung von optionalen Feldern. Diese ermöglichen es, dass in der Struktur nur die Daten angezeigt werden, welche sich zu dem gegebenen Zeitpunkt auch geändert haben. Dies bedeutet insbesondere eine Verringerung der Last für Datenübertragung und Auswertung innerhalb der Client-Applikationen. Die Echtzeitdaten erhält der OPC UA Client durch Nutzung der OPC UA Dienste Subscriptions und MonitoredItems, indem er die entsprechende Variable in einer OPC UA Subscription überwacht. Abbildung 5-6 zeigt die Darstellung der Strukturvariable mit Beispielwerten in einem generischen OPC UA Client.

In Abbildung 5-7 wird die Datenbereitstellung für die Echtzeitdaten innerhalb der MTX gezeigt. Die Wertänderungen der gewünschten Daten werden im Echtzeitbereich nach der Interpolationsberechnung erfasst und in einem Ringbuffer zwischengespeichert. Dieser stellt sicher, dass der OPC UA Server lückenlos alle Wertänderungen erhält und an den OPC UA Client übertragen kann. Zudem können durch den Ringbuffer kurze Unterbrechungen oder Verzögerungen in der Nicht-Echtzeitkommunikation des OPC UA Server ohne Datenverlust abgefangen werden. Der OPC UA Client erhält für jeden Interpolationszeitpunkt, bei der eine Datenänderung erfolgte, einen Wert der zuvor beschriebenen Struktur. Durch Nutzung der Struktur kann die zeitliche Konsistenz sichergestellt werden, da garantiert ist, dass alle Datenänderungen für einen Zeitpunkt in der Struktur enthalten sind. Dies erweist sich insbesondere bei der Darstellung und Analyse von Prozessen als sehr vorteilhaft.



Abbildung 5-5: OPC UA Datentyp für die Übertragung der Echtzeitdaten in zeitlich konsistenter Form (Quelle: Bosch Rexroth AG)

Abbildung 5-6: Darstellung der konsistenten Echtzeitdaten im OPC UA Client UaExpert von Unified Automation (Quelle: Bosch Rexroth AG)

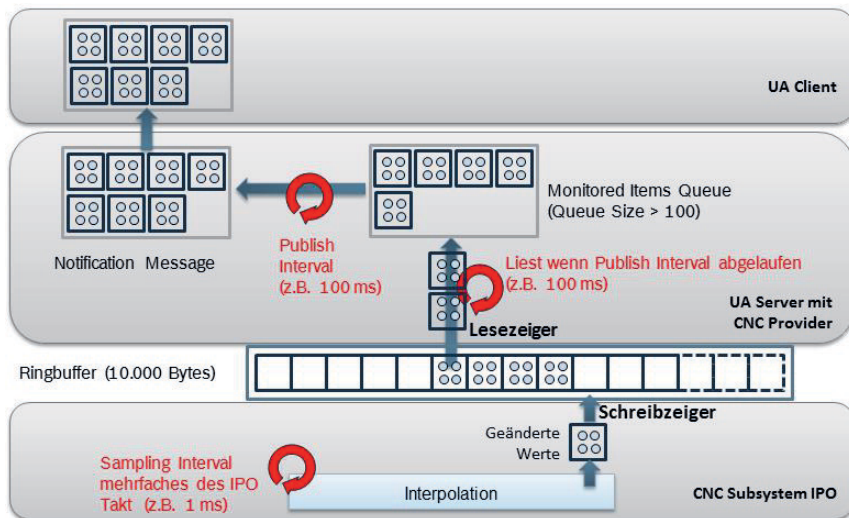


Abbildung 5-7: Datenbereitstellung der Echtzeitdaten – Weg vom Echtzeitbereich der Steuerung bis zum OPC UA Client (Quelle: Bosch Rexroth AG)

5.5 Ergebnisse und Ausblick

Die Anbindung eines CNC-Systems einer Werkzeugmaschine über OPC UA wurde mit dem Rexroth CNC-System MTX erreicht. Erfolgreich war insbesondere die Umsetzung der Bereitstellung von Echtzeitdaten über OPC UA. In Rahmen des iWindow-Projekts konnte diese in mehreren Anwendungen und Mehrwertdiensten erfolgreich angewendet werden. Für mehr Flexibilität bietet sich als Erweiterung eine Konfigurationsmöglichkeit der Echtzeitdaten an. Dadurch lassen sich die gewünschten Daten noch mehr auf die benötigte Anwendung und Dienste anpassen. Dies geht insbesondere mit dem Wunsch einher, eine große Auswahl an Datenelementen für die Konfiguration zur Verfügung zu haben. Dies sollten in Zukunft auch Daten aus dem Echtzeitbereich der SPS sein.

Bei der Umsetzung einer allgemeingültigen OPC UA Schnittstelle wurden auch zahlreiche Fortschritte erreicht, auf die in zukünftigen Arbeiten aufgebaut werden kann. Es wurde erkannt, dass insbesondere bei den CNC-spezifischen Daten und Funktionen noch keine vollständige Beschreibung in Form eines OPC UA Informationsmodells vorhanden ist. Die Evaluierung der OPC UA VDW Companion Specification *OPC UA Informationsmodell for CNC Systems* in einer prototypischen Umsetzung auf der MTX war vielversprechend, jedoch nicht vollständig befriedigend. Es wurde erkannt, dass das bestehende Informationsmodell grundlegend geeignet ist, aber an vielen Stellen noch eine ausführlichere Beschreibung benötigt. Nur dadurch kann eine eindeutige Semantik des Informationsmodells und der dazugehörigen OPC UA Schnittstelle erreicht werden. Des Weiteren wurde noch der Bedarf

an Erweiterungen am bestehenden Informationsmodell erkannt und die dazugehörigen Elemente identifiziert. Diese Ergebnisse können in zukünftige Spezifikationsarbeiten zu diesem Informationsmodell einfließen.

5.6 Literaturverzeichnis

- [1] Rauen, H., Mosch, C., Niggemann, O. & Jasperneite, J.: *Industrie 4.0 Kommunikation mit OPC UA. Leitfaden zur Einführung in den Mittelstand*, 2017. Hg. v. VDMA und Fraunhofer-Anwendungszentrum Industrial Automation. Frankfurt am Main (978-3-8163-0709-9).
- [2] Brecher, C., Verl, A., Lechler, A. & Servos, M.: *Open control systems: state of the art*, 2010. *Production Engineering* 4 (2-3), S. 247-254. Internet: <http://dx.doi.org/10.1007/s11740-010-0218-5>.
- [3] Lutz, P. & Sperling, W.: *OSACA - The vendor neutral control architecture, Facilitating Deployment of Information and Communications Technologies for Competitive Manufacturing*, 1997, 247-256
- [4] Brühl, J.: *Methoden zur Realisierung einheitlicher Schnittstellen für offene Steuerungssysteme*, Dissertation, RWTH Aachen, 2003
- [5] Schleipen, M.; Münnemann, A. & Sauer, O.: *Interoperabilität von Manufacturing Execution Systems (MES) - Durchgängige Kommunikation in unterschiedlichen Dimensionen der Informationstechnik in produzierenden Unternehmen*, at - Automatisierungstechnik, Oldenbourg Wissenschaftsverlag, 07/2011, 59, 413-424
- [6] N. N.: *OPC Unified Architecture Specification, Part 3: Address Space Model*, Release 1.04. OPC Foundation, Scottsdale, Arizona (USA), 2017.
- [7] N. N.: *OPC Unified Architecture for Devices Companion Specification*, Release 1.01. OPC Foundation, Scottsdale, Arizona (USA), 2015. Internet: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-devices-di/>
- [8] N. N.: *OPC Unified Architecture Specification, Part 9: Alarms & Conditions*, Release 1.04. OPC Foundation, Scottsdale, Arizona (USA), 2017. Internet: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-9-alarms-and-conditions/>
- [9] N. N.: *OPC Unified Architecture Specification, Part 5: Information Model*, Release 1.04. OPC Foundation, Scottsdale, Arizona (USA), 2017. Internet: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-5-information-model/>

-
- [10] N. N.: *PLCopen and OPC Foundation: OPC UA Information Model for IEC 61131-3*, Release 1.00, 2010. Internet: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-plcopen-information-model/>
 - [11] N. N.: *OPC Unified Architecture for Sercos Devices Companion Specification*, Release 1.2, 2017. Internet: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-sercos-devices/>
 - [12] N. N.: *VDW and OPC Foundation: OPC UA Information Model for CNC Systems*, Release 1.0, 2017. Internet: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-cnc-systems/>

6 Visualisierung des Maschineninnenraums mit VR & AR

P. Sommer, A. Verl

6.1 Darstellungsarten

Zur Darstellung des Maschineninnenraums und zusätzlicher Mehrwertinformationen eignen sich verschiedene Darstellungsarten. Je nach Anwendungsfall können mehr oder weniger reale Komponenten durch virtuelle ersetzt werden. Somit kann beispielsweise der ganze Maschineninnenraum virtuell dargestellt oder ausschließlich virtuelle Mehrwertinformationen in den realen Innenraum integriert werden. Die Mengen der virtuellen bzw. realen Inhalte lassen sich hierbei wie auf einem Schieberegler entsprechend den aktuellen Anforderungen lückenlos anpassen. Diese Gegebenheit und die zugehörigen Begriffe werden durch das Reality-Virtuality Kontinuum nach Milgram [1] beschrieben, siehe Abbildung 6-1.

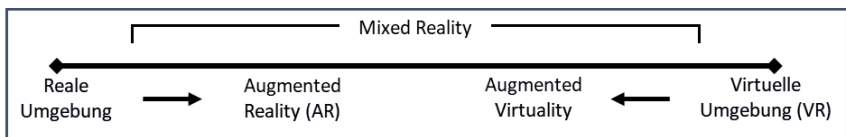


Abbildung 6-1: Reality-Virtuality Kontinuum nach Milgram [1]

Die virtuelle Realität (engl. Virtual Reality; VR), beschreibt das interaktive Erleben einer computergenerierten Umgebung. Von großer Bedeutung ist dabei die Immersion, die den Grad des Eintauchens in die virtuelle Welt und der mentalen Präsenz beschreibt. Hierzu bedarf es auch einer Möglichkeit den eigenen Beobachtungspunkt in der virtuellen Umgebung interaktiv verändern zu können. Dies kann beispielsweise durch sensorische Erfassung tatsächlicher Bewegungen im Raum geschehen. [2]

Die erweiterte Realität (engl. Augmented Reality, AR) kombiniert die reale Umgebung mit virtuellen Inhalten. Meist geschieht dies durch visuelle Erweiterung, es können jedoch auch andere Sinne wie Hören, Fühlen, Riechen und Schmecken angesprochen werden. Im Folgenden wird nur auf die visuelle Erweiterung der Realität eingegangen. AR erfordert ebenfalls Interaktivität in Form der Anpassung des Beobachtungspunkt. Dies geht jedoch immer mit der Veränderung des Beobachtungspunktes in der realen Umgebung einher. Damit virtuelle Objekte sich zu jeder Zeit korrekt in die reale Umgebung integrieren, müssen die Position und die Perspektive des Beobachters bekannt sein. Diese perspektivische Überlagerung nennt man Registrierung (engl. Registration). [3]

Je nach Einsatzzweck und daraus resultierenden Anforderungen eignen sich verschiedene Darstellungsarten des Reality-Virtuality Kontinuums, sowie verschiedene Technologien und Medien zur Visualisierung bzw. Beobachtung des Maschineninnenraums. Grundsätzlich ergibt

sich jedoch durch Verwendung eines umfangreichen Modells des Maschineninnenraums die Möglichkeit auf alle diese Darstellungsarten, Technologien und Medien (siehe Abbildung 6-2) zurückzugreifen.

Sollen beispielsweise alle Maschinenfenster entfernt und durch herkömmliche Displays ersetzt werden, ist eine direkte Beobachtung des Innenraumes nicht mehr möglich. Hier bieten sich nun entweder eine rein virtuelle Darstellung mit Hilfe des Modells (VR) oder eine Erfassung mit Innenraumkameras an (AR). Eine rein virtuelle Darstellung ermöglicht eine Beobachtung aus jeder beliebigen Perspektive, wohingegen Kameras die Beobachtung auf die Blickwinkel der Kameras einschränken. In beiden Fällen dient das Modell zur korrekten Integration von zusätzlichen Mehrwertinformationen. Werden Mehrwertinformationen in eine rein virtuelle Darstellung integriert, ist die Überlagerung und korrekte Verdeckung zwischen Komponenten der Maschine und Mehrwertinformationen relativ einfach zu lösen. Komplexer gestaltet sich diese Integration bei Verwendung von Kameras. Hierbei müssen einerseits die Perspektive der Kamera, andererseits die korrekte Verdeckung zwischen realen Objekten im Kamerabild und virtuellen Mehrwertinformationen berücksichtigt werden. Die dafür eingesetzten Methoden werden in Kapitel 6.4 ausführlicher erläutert.

Wird Wert darauf gelegt, dass der Innenraum direkt durch ein Fenster sichtbar ist, können optische Elemente wie z.B. transparente Displays eingesetzt werden. Ein solches transparentes Display kann direkt vor bereits vorhandenen Maschinenfenstern angebracht werden. Im Falle einer überlagernden Darstellung (AR) werden ausschließlich zusätzliche Mehrwertinformationen auf diesem Display überlagernd dargestellt. Um eine korrekte Überlagerung erzeugen zu können, ist es notwendig die Perspektive des Betrachters nachzubilden und die Mehrwertinformationen entsprechend darzustellen. Problem die hierbei auftreten können ergeben sich durch Fokussierung und Parallaxe. Neben der Nutzung von Augmented Reality zur überlagernden Darstellung, ermöglicht ein transparentes Display auch eine rein virtuelle Darstellung (VR) des gesamten Innenraums. Ein transparentes Display bietet somit die Möglichkeit, vergleichbar mit einem Schieberegler, lückenlos ausgehend von einer reinen Beobachtung realer Objekte mehr und mehr virtuelle Inhalte hinzuzufügen.

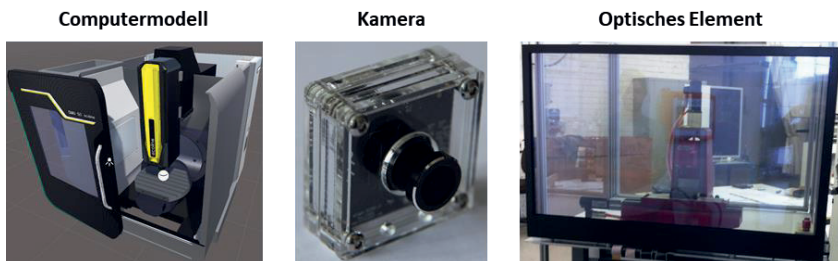


Abbildung 6-2: Verschiedene Medien zur Betrachtung des Maschineninnenraums
(Quelle: ISW)

Zu Fertigung transparenter Displays kommen grundsätzlich zwei Technologien in Frage. Flüssigkristallanzeigen (engl. liquid crystal display, LCD) bestehen aus in einem gleichmäßigen Raster angeordneten Flüssigkristallen. Diese Kristalle können unabhängig voneinander ihre Polarisationsrichtung ändern. Auf diese Weise ändert sich die Durchlässigkeit für polarisiertes Licht, das mit einer Hintergrundbeleuchtung (engl. Backlight) und Polarisationsfiltern erzeugt wird. Das resultierende Bild wird durch Ansteuerung der Transparenz der einzelnen Flüssigkristalle erzeugt, wohingegen das ausgehende Licht vom dauerhaft aktiven Backlight ausgeht. Die zur Bildgenerierung angesteuerten Pixel eines auf organischen Leuchtdioden (engl. organic light emitting diode, OLED) basierende Displays hingegen sind selbstleuchtend und benötigen daher kein zusätzliches Backlight. Auf Grund der notwendigen Hintergrundbeleuchtung eignet sich die LCD Technologie nur eingeschränkt für transparente Displays. Üblicherweise werden transparente LCDs in Verbindung mit geschlossenen und stark beleuchteten Boxen verwendet, in denen einzelne kleinere Gegenstände präsentiert werden. Bereits bei größeren Boxen und Gegenständen kann es zu Beeinträchtigungen durch Verschattung kommen. Im Umfeld einer Werkzeugmaschine ist damit zu rechnen, dass selbst bei starker Ausleuchtung des Innenraums, die Maschinenspindel, Werkstücke, Spannvorrichtungen, Späne und Kühlschmiermittel zu weiteren Beeinträchtigungen durch Verschattung führen. Selbstleuchtende OLED Displays umgehen dieses Problem, da das Licht erst außerhalb des Maschineninnenraums erzeugt wird. Ein weiterer Vorteil von OLED Displays zeigt sich im inaktiven Zustand. Während LCDs im inaktiven Zustand lichtundurchlässig und damit undurchsichtig sind, sind OLED Displays im ausgeschalteten Zustand vollständig transparent. Transparenz im inaktiven Zustand ist dann wichtig, wenn auch bei ausgeschalteter Maschine oder trotz durch eine Fehlfunktion verursachter plötzlicher Spannungsausfall am Display der Innenraum einsehbar sein soll.

6.2 Virtuelles Modell des Maschineninnenraums

Wie bereits erläutert dient unabhängig der Darstellungsart ein virtuelles Modell des Maschineninnenraums als Grundlage für die Visualisierung und zur Integration von Mehrwertinformationen. Ein solches Modell muss möglichst genau dem tatsächlichen Innenraum der Maschine entsprechen. Im Falle einer rein virtuellen Darstellung (VR) stellt es die einzige Möglichkeit zur Beobachtung des Innenraumes dar und muss diesem daher zu jeder Zeit sehr exakt entsprechen. Ist der reale Innenraum entweder indirekt mittels Kameras oder direkt durch ein transparentes Display sichtbar, dient das Modell zur Integration von Mehrwertinformationen mittels Augmented Reality. Hierbei werden alle Mehrwertinformationen dreidimensional im Modell verankert und ermöglichen somit die korrekte perspektivische Darstellung entsprechend der Orientierung und Position der Kameras (vgl. Kapitel 6.4) bzw. des menschlichen Betrachters vor einem Display (vgl. Kapitel 6.5). Im Falle einer überlagernden Darstellung (AR) entstehen zwangsläufig zwei Betrachtungsebenen, eine Ebene aus realen Objekten (direkt oder indirekt sichtbar) und eine

Ebene aus virtuellen Objekten. Zwischen diesen Betrachtungsebenen kommt es zu gegenseitigen Verdeckungen, entsprechend der dreidimensionalen Positionen der realen und virtuellen Objekte. Um solche Verdeckungen auflösen und korrekt darstellen zu können (vgl. Kapitel 6.4), ist es notwendig den gesamten Innenraum, reale und virtuelle Objekte, in einem zu jeder Zeit exakten Modell vorzuhalten.

Als Basis für ein möglichst exaktes Modell des Maschineninnenraums eignen sich CAD-Modelle. In der produzierenden Industrie werden heute nahezu durchgängig CAD-Modelle als Standard in der Konstruktion eingesetzt. Eine weitere Verwendung dieser Daten über die Konstruktion hinaus findet bisher jedoch häufig nicht statt. Um CAD-Daten zu Visualisierungszwecken nutzen zu können gilt es einige Dinge zu beachten. Da das Modell möglichst exakt der tatsächlichen Maschine entsprechen soll, müssen die CAD-Daten einerseits mit allen Details erstellt und andererseits nachträgliche Anpassungen und Änderungen eingepflegt werden. Hierbei spielen je nach verwendeter Darstellungsart entweder nur geometrische oder auch optische Aspekte (z.B. Texturen oder Spiegelungen) eine Rolle. Weiterhin erfordert die Nachbildung des realen Innenraumes auch eine Synchronisierung mit Maschinendaten (vgl. Kapitel 6.3). Diese zu synchronisierenden Daten beinhalten beispielsweise Achspositionen, aktuelles Werkzeug, Materialabtrag, usw. Um eine solche Synchronisierung zu ermöglichen, muss das Maschinenmodell in einzeln transformierbare (Translation bzw. Rotation) Komponenten bzw. Baugruppen unterteilt sein. Des Weiteren sollten diese Baugruppen in einer Hierarchie (Eltern-Kind-Beziehungen) angeordnet sein, welche der kinematischen Kette der Maschine entspricht. Die Gestaltung der Koordinatensysteme einzelner Baugruppen ist ebenfalls zu berücksichtigen, um Translations- und Rotationsachsen unkompliziert definieren zu können. [5]

Zu Visualisierung des Innenraummodells sind, wie in Abbildung 6-3 dargestellt, grundsätzlich verschiedene Ansätze möglich:

- Durch Nutzung gängiger Programmierschnittstellen (APIs) wie beispielsweise DirectX oder OpenGL, können eigene Applikationen mit grafischer 3D-Visualisierung entwickelt werden. Die sehr grundlegenden Funktionen dieser Schnittstellen führen jedoch üblicherweise zu einem relativ hohen Entwicklungsaufwand. Diese Variante empfiehlt sich daher nur für die Weiterentwicklung bereits vorhandener Applikationen bzw. wenn die nachfolgenden Ansätze auf Grund anderer Anforderungen nicht in Frage kommen.
- Applikationen, die bereits Funktionalität zur grafischen 3D-Visualisierung besitzen, können gegebenenfalls durch Plugins oder ähnliches erweitert werden. Hierzu gehören auch CAD-Konstruktionsprogramme, die sich um eigene Funktionalität erweitern lassen. Diese Programme verfügen meist über qualitativ hochwertige Visualisierungsmöglichkeiten und umfangreiche Szeneneditoren zur Gestaltung der virtuellen 3D-Szene. Die Erweiterungsmöglichkeiten durch Plugins sind jedoch, falls

überhaupt vorhanden, meist stark eingeschränkt und somit nur für manche Anwendungsfälle geeignet.

- Toolkits und Frameworks zur Gestaltung interaktiver 3D-Visualisierungen bieten ebenfalls umfangreiche und qualitativ hochwertige Visualisierungsmöglichkeiten und verfügen über umfassende Szeneneditoren. Toolkits wie z.B. OpenSceneGraph [4] bauen auf den bereits genannten Visualisierungs-APIs auf und reduzieren den Entwicklungsaufwand durch eine einfachere knotenbasierte Beschreibung virtueller Umgebungen. Funktional noch etwas umfangreicher sind Frameworks wie beispielsweise sogenannte Game Engines zur Entwicklung von Computerspielen und anderer interaktiver und visueller Applikationen (z.B. Architektur, Produktdesign, Simulation). Üblicherweise verfügen Game Engines über umfangreiche Kernfunktionalitäten wie beispielsweise Renderingsystem, Soundsystem, Physiksystem und können ohne größere Modifikationen wiederverwendbar eingesetzt werden. [6] Aktuelle Beispiele für Game Engines sind Unity und die Unreal Engine.

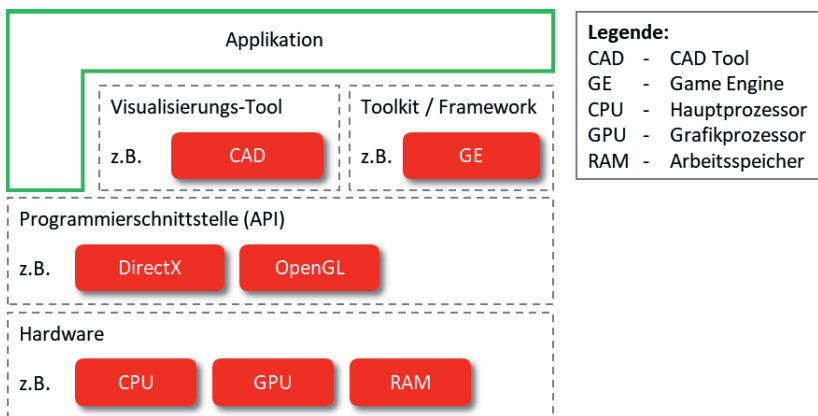


Abbildung 6-3: Applikationsentwicklung zur grafischen Visualisierung (Quelle: ISW)

6.3 Synchronisierung mit aktuellen Maschinendaten

Wie in den vorigen Kapiteln bereits erläutert erfordert die Nachbildung des realen Innenraumes auch eine Synchronisierung mit Maschinendaten. Dies beinhalten beispielsweise Achspositionen, aktuelles Werkzeug, Materialabtrag, usw. Eine Anbindung an die hierfür benötigten Daten erfolgt mittels Implementierung der in Kapitel 4.3 erläuterten Kommunikation zwischen Diensten. Die Visualisierung tritt hierbei als Basisdienst im Mehrwertdienstframework auf und koppelt sich an die Daten des Dienstes zur Steuerungsanbindung.

Wenn das Maschinenmodell, wie in Kapitel 6.2 erläutert, in einzeln transformierbare (Translation bzw. Rotation) Komponenten bzw. Baugruppen unterteilt ist, lassen sich diese

entsprechend der aktuellen Achspositionen bewegen. Durch korrekt definierte Hierarchien (Eltern-Kind-Beziehungen) wird hierbei die kinematische Kette der Maschine berücksichtigt. [5]

Durch Kopplung mit den Daten über das aktive Werkzeug lässt sich der Werkzeugwechsel ebenfalls nachbilden. Die Geometrie der Werkzeuge kann entweder vorab beispielsweise mit Hilfe von CAD-Daten hinterlegt werden (siehe Abbildung 6-4) oder falls vorhanden zur Laufzeit aus Werkzeugparametern der Steuerung generiert werden. Aus CAD-Daten generierte Werkzeugmodelle sind genauer und detaillierter, wohingegen eine automatische Generierung aus Parametern den Einrichtungsaufwand minimiert. Mit Blick auf den durch Industrie 4.0 geprägten Begriff des digitalen Zwillings, sollten solche Werkzeugmodelle sowie die zugehörigen Werkzeugparameter jederzeit mit den physischen Werkzeugen mitgeführt werden.

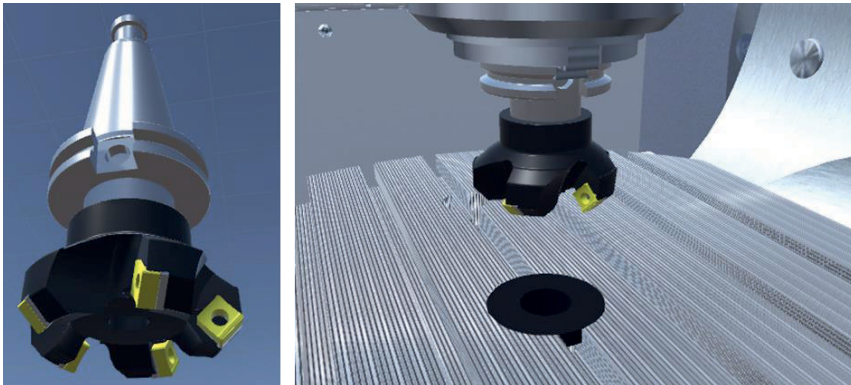


Abbildung 6-4: Werkzeug im virtuellen Maschinenmodell (Quelle: ISW)

Durch aktuelle Achspositionen und Werkzeugparameter lassen sich der Materialabtrag und damit verbunden die aktuelle Geometrie des Werkstücks ebenfalls im Maschinenmodell nachbilden (vgl. Kapitel 8.2). Dies spielt, wie die bereits genannten Methoden zur Synchronisierung der Maschinenachsen und des aktiven Werkzeugs, ebenfalls eine Rolle wenn eine rein virtuelle Darstellung die einzige Möglichkeit zur Beobachtung des Innenraumes darstellt, sowie zur Verdeckungsrechnung bei der Integration von Mehrwertinformationen (siehe Kapitel 6.4). Des Weiteren kann auf diese Weise auch ein virtuelles Werkstück in die reale Umgebung integriert werden und somit eine geplante Bearbeitung vorab validiert werden, ohne die Gefahr von beispielsweise Kollisionen, Werkzeugbrüchen und anderen Beschädigungen einzugehen (siehe Kapitel 9).

Abbildung 6-5 zeigt zwei Momentaufnahmen eines synchronisierten virtuellen Maschinenmodells mit bewegten Maschinenachsen, aktivem Werkzeug und Materialabtrag.

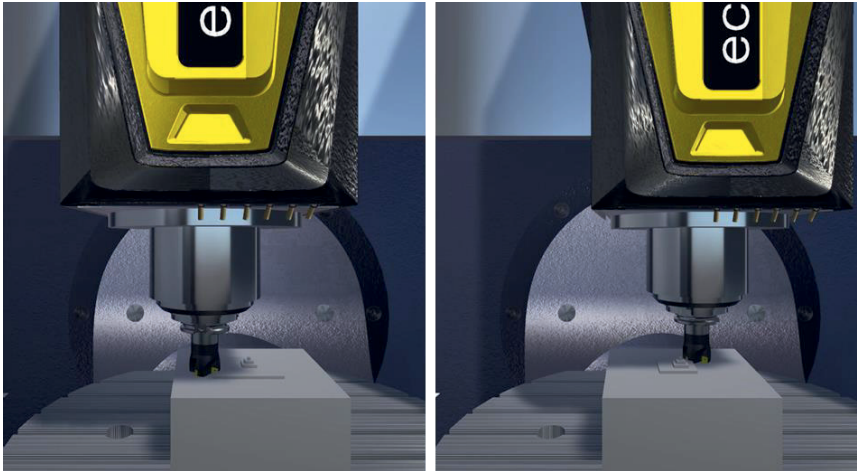


Abbildung 6-5: Synchronisiertes Maschinenmodell und Materialabtrag (Quelle: ISW)

6.4 Kamerabasierte AR

Wie eingangs bereits erläutert ergibt sich durch die Nutzung von Kameras die Möglichkeit den Maschineninnenraum indirekt zu beobachten. Dadurch können beispielsweise neue Betrachtungspositionen (insbesondere bei großen Maschinen) geschaffen oder alle Maschinenfenster entfernt und durch herkömmliche Displays ersetzt werden. Zur Integration von Mehrwert-informationen in ein Kamerabild müssen einerseits die Perspektive der Kamera, andererseits die korrekte Verdeckung zwischen realen Objekten im Kamerabild und virtuellen Mehrwertinformationen berücksichtigt werden.

6.4.1 Beschreibung der Kameraperspektive

Um die Perspektive einer Kamera und darauf aufbauend die perspektivische Darstellung anderer Objekte aus Sicht der Kamera beschreiben zu können, werden verschiedene Koordinatensysteme verwendet, dargestellt in Abbildung 6-6.

Allen Objekten im Raum lassen sich eine Position und eine Orientierung in einem dreidimensionalen Weltkoordinatensystem zuordnen. Im Falle von Objekten in einer Werkzeugmaschine handelt es sich hierbei um das Maschinenkoordinatensystem mit dem Maschinennullpunkt als Ursprung [7]. Auch der Kamera, mit der der Maschineninnenraum beobachtet wird, lassen sich eine Position und eine Orientierung in diesem Koordinatensystem zuordnen.

Jede Kamera definiert ein lokales dreidimensionales Kamerakoordinatensystem. Üblicherweise liegt hierbei die Z-Achse dieses Koordinatensystems in oder entgegen der Blickrichtung der Kamera. [8] Objekte im Weltkoordinatensystem lassen sich durch

Koordinatentransformation auch im Kamerakoordinatensystem darstellen. Bei dieser Transformation, die aus Rotation und Translation besteht, handelt es sich um eine affine Abbildung [9]. Da diese Transformation nicht von den Eigenschaften der Kamera, sondern nur von deren Position und Orientierung im Raum abhängig ist, werden die für diese Transformation benötigten Parameter als äußere (extrinsische) Kameraparameter bezeichnet [8].

Ausgehend vom Kamerakoordinatensystem werden sichtbare Objekte mittels Projektion auf zweidimensionale Bildkoordinaten abgebildet. Hierbei handelt es sich entsprechend dem Lochkameramodell um eine zentralprojektive Abbildung. Diese Abbildung ist ausschließlich von den optischen Eigenschaften der Kamera und nicht von deren Position und Orientierung im Raum abhängig, daher werden die benötigten Parameter als innere (intrinsische) Kameraparameter bezeichnet. [8]

Die genannten intrinsischen und extrinsischen Kameraparameter lassen sich durch Methoden der Kamerakalibrierung, näher erläutert in Kapitel 7, vermessen.

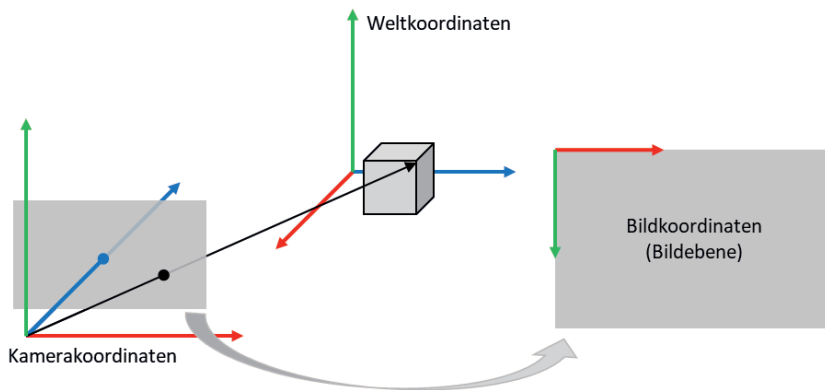


Abbildung 6-6: Transformation von Welt- über Kamera- in Bildkoordinaten (Quelle: ISW)

Der von der zentralprojektiven Abbildung betroffene Raum wird bei realen Kameras durch den Bildsensor und im Computerkameramodell durch die Bildebene bzw. den Bildschirm begrenzt. Da sowohl Bildsensoren wie auch Displays üblicherweise eine rechteckige Form aufweisen, entsteht ein pyramidenförmiger Abbildungsraum dessen Spitze an der Kameraposition liegt. Die vier diesen Raum seitlich begrenzenden Pyramidenflächen werden entsprechend der zugehörigen Kanten des Bildschirms als linke, rechte, obere und untere Fläche (engl. plane) bezeichnet. Die minimale und die maximale Entfernung für die Kamera sichtbarer Objekte werden durch die sogenannten Clipping Planes definiert. Die Near Clipping Plane begrenzt den Abbildungsraum hinsichtlich der minimalen, die Far Clipping Plane hinsichtlich der maximalen Entfernung von der Kameraposition. Begrenzt durch die vier seitlichen Pyramidenflächen und

die zwei Clipping Planes entsteht, wie in Abbildung 6-7 dargestellt, ein Abbildungsraum in Form eines Pyramidenstumpfes (engl. frustum). [10] Die Position der Kamera, die die Pyramidenspitze des Abbildungsraumes darstellt, wird im Zusammenhang mit der zentralprojektiven Abbildung auch Augenpunkt genannt, da sich hier das Auge des Betrachters befindet. [11]

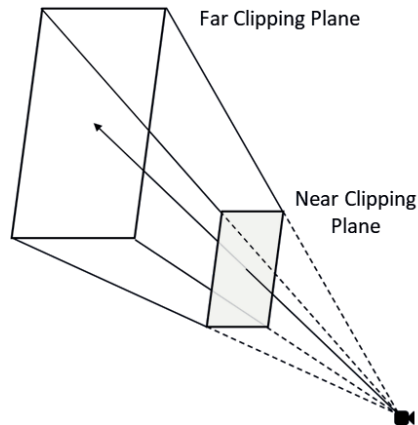


Abbildung 6-7: Abbildungsraum einer Kamera in Form eines Pyramidenstumpfes (Quelle: ISW)

6.4.2 Überlagerung des Kamerabilds mit virtuellen Inhalten

Ausgehend von der Vermessung und Beschreibung der Kameraparameter lässt sich die Perspektive der physischen Kamera im Computermodell nachbilden. Dazu werden die Position und Orientierung des Computerkameramodells entsprechend der extrinsischen und die optischen Eigenschaften des Modells entsprechend der intrinsischen Parameter angepasst. Durch diese Konfiguration erzeugt die Kamera im Computermodell dieselbe Ansicht, d.h. denselben Abbildungsraum, wie die physische Kamera im Maschineninnenraum.

Zur Überlagerung beider Bilder muss definiert werden, welche Teile des resultierenden Bildes aus dem Computermodell und welche von der physischen Kamera übernommen werden sollen. Eine häufig verwendete Methode hierfür ist das Chroma-Key-Verfahren. Bei diesem Verfahren wird der Hintergrund des computergenerierten Bildes mit einer einheitlichen Farbe, die im übrigen Bild nicht verwendet wird, eingefärbt. Zur Überlagerung der Bilder werden anschließend alle dieser Farbe entsprechenden Pixel durch den Inhalt des Kamerabildes ersetzt. [12] Abbildung 6-8 zeigt anhand eines Beispiels die Überlagerung eines realen Werkstücks im Kamerabild mit virtuellen Spanneisen aus einem Computermodell. Zu beachten ist, dass bei diesem Verfahren die virtuellen Elemente grundsätzlich vor dem Kamerabild und damit im Vordergrund der darin enthaltenen Objekte dargestellt werden. Virtuelle Objekte können reale Objekte im Kamerabild verdecken, jedoch nicht umgekehrt.

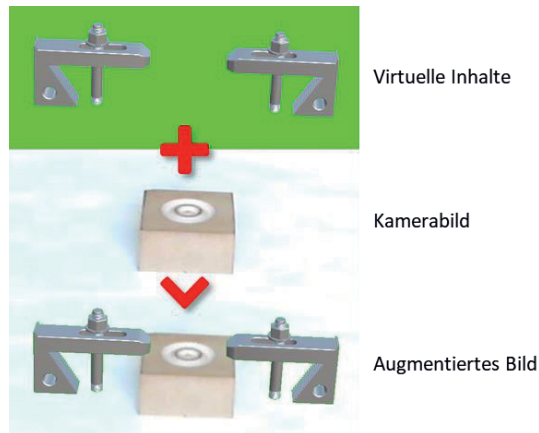


Abbildung 6-8: Überlagerung mit dem Chroma-Key-Verfahren (Quelle: ISW)

6.4.3 Verdeckungsberechnung

Durch fehlende oder einseitige Verdeckung zwischen realen und virtuellen Objekten kommt es zu einer Beeinträchtigung des räumlichen Vorstellungsvermögens. Objekte die sich eigentlich im Hintergrund befinden, werden im Vordergrund wahrgenommen. Um korrekte Verdeckungen zwischen realen und virtuellen Inhalten darstellen zu können, bedarf es Informationen darüber, welche Objekte sich im Vordergrund befinden. Sind diese Informationen, beispielsweise aus einer Tiefenkamera vorhanden, kann bereits im Überlagerungsschritt mittels der Depth Keying Methode eine korrekte Verdeckung erzeugt werden. Bei dieser Methode wird anhand der entsprechenden Tiefeninformationen für jedes Bildpixel entschieden aus welchem Ausgangsbild, Kamerabild oder Computermodell, das resultierende Pixel entnommen wird [12]. Verfügt die genutzte Kamera nicht über Funktionalität zur Tiefenmessung, kann auch ein Computermodell der realen Umgebung zur Verdeckungsberechnung genutzt werden [13]. Voraussetzung zur Verwendung eines Computermodells des Maschineninnenraums zur Verdeckungsberechnung ist, dass das Modell zu jeder Zeit exakt dem tatsächlichen Innenraum entspricht. Dazu zählen die geometrische Gestaltung auf Basis von CAD-Daten, wie auch die Synchronisierung mit aktuellen Maschinendaten.

Eine Verdeckungsberechnung durch ein der realen Umgebung entsprechendes Computermodell lässt sich auf Basis des Chroma-Key-Verfahrens realisieren. Dazu werden Objekte im Computermodell, die real vorhandenen Objekten im Maschineninnenraum entsprechen, mit der Hintergrundfarbe eingefärbt. Dies führt dazu, dass im Hintergrund liegende Objekte im Computermodell durch diese digitalen Abbilder verdeckt werden. Anschließend wird an der entsprechenden Stelle die Hintergrundfarbe durch das Kamerabild ersetzt und damit das verdeckende reale Objekt sichtbar gemacht.

Das Verfahren zur Verdeckungsberechnung wird in Abbildung 6-9 an einem Beispiel veranschaulicht. Im Beispiel wird ein reales Werkstück von einer Kamera erfasst und das resultierende Kamerabild durch Überlagerung mit Spanneisen und einer Werkzeugbahn erweitert. Die Werkzeugbahn verläuft dabei hinter dem Werkstück, ebenso befindet sich eines der Spanneisen hinter dem Werkstück. Diese virtuellen Objekte sollten folglich durch das reale Werkstück verdeckt werden. Voraussetzung für die Berechnung und Darstellung der Verdeckung sind die Beschreibung und Nachbildung der Kameraperspektive (vgl. Kapitel 6.4.1) sowie die Überlagerung von computergeneriertem Bild und Kamerabild mittels Chroma-Key-Verfahren (vgl. Kapitel 6.4.2). Im linken Bild ohne Verdeckungs-berechnung führt die falsche Darstellung zu einer Beeinträchtigung des räumlichen Vorstellungsvermögens. Das mittlere Bild zeigt das verdeckende virtuelle Werkstück in einer einheitlichen Farbe. Dieses virtuelle Werkstück entspricht dem realen Werkstück in Geometrie, Position und Orientierung. Die Verdeckung der Werkzeugbahn und des hinteren Spanneisens durch das virtuelle Werkstück ist bereits sichtbar. Im rechten Bild ist abschließend die finale Darstellung mit korrekter Verdeckungs-berechnung dargestellt. Das virtuelle Abbild des Werkstücks wurde mittels Chroma-Key-Verfahren durch das reale Werkstück im Kamerabild ersetzt. Das reale Werkstück verdeckt wie gewünscht Werkzeugbahn und Spanneisen.

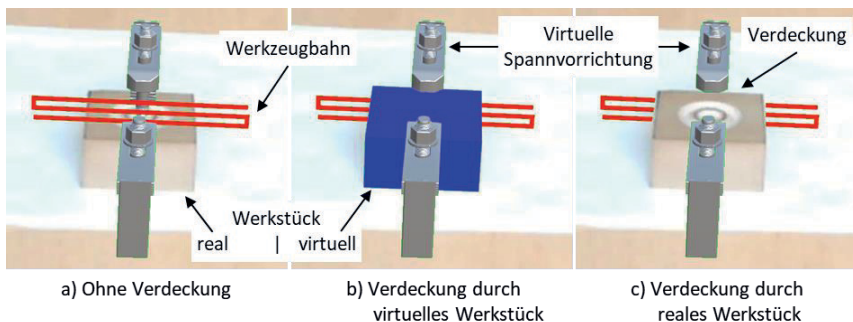


Abbildung 6-9: Beispiel Verdeckungsberechnung (Quelle: ISW)

6.5 Berücksichtigung der Betrachterperspektive

Die Perspektive des Betrachters spielt sowohl bei der rein virtuellen Darstellung wie auch bei der Überlagerung am transparenten Display eine Rolle. Das menschliche Auge wird durch das physikalische Prinzip einer Kamera mit Linse und Blende beschrieben [14]. Daher kann die perspektivische Darstellung virtueller Inhalte durch ein Computerkameramodell nachgebildet werden.

Durch Berücksichtigung der Betrachterperspektive können mit herkömmlichen Displays und dem Computermodell des Maschineninnenraums beliebig viele und große Fenster imitiert werden. Auf diese Weise können Werkzeugmaschinen zum Schutz der Bediener vollständig

mit einem Gehäuse umschlossen werden, treuere und wartungsintensive Schutzscheiben entfallen. Die verwendeten Displays können an beliebiger Stelle am Maschinengehäuse angebracht werden, auch dort wo sonst Maschinenkomponenten die Sicht einschränken würden. [15]

Zur Überlagerung des Maschineninnenraums mit virtuellen Inhalten an transparenten Displays ist die Berücksichtigung der Perspektive des Betrachters unabdingbar. Nur auf diese Weise lassen sich die virtuellen Inhalte an der richtigen Stelle und in korrekter Verzerrung auf dem Display darstellen. [15] Leichte Abweichungen bei der Nachbildung der Betrachterperspektive führen in Augmented Reality Szenarien zu elementaren Auswirkungen und beeinträchtigen des räumlichen Vorstellungsvermögens, wohingegen die Auswirkungen in Virtual Reality Szenarien weniger gravierend sind [10]. Überlagerung und Verdeckungsrechnung erfolgen an transparenten Displays auf gleiche Weise wie bei der kamerabasierten Überlagerung. Einzig die Schritte zum Ersetzen der, in diesem Fall transparenten, Hintergrundfarbe durch das Kamerabild entfallen.

Unabhängig der gewählten Darstellungsart lässt sich die Perspektive des Betrachters durch dasselbe Vorgehen nachbilden. Dieses Vorgehen, dargestellt in Abbildung 6-10, gliedert sich in drei Schritte. Da der sichtbare Bereich des Maschineninnenraums durch das Display begrenzt wird, muss zuerst dieses Display beschrieben und nachgebildet werden. Des Weiteren ist die perspektivische Darstellung abhängig von der Position des Betrachters, die daher erfasst werden muss. Anschließend können auf Basis dieser Informationen die zur Darstellung benötigten perspektivischen Parameter berechnet werden.

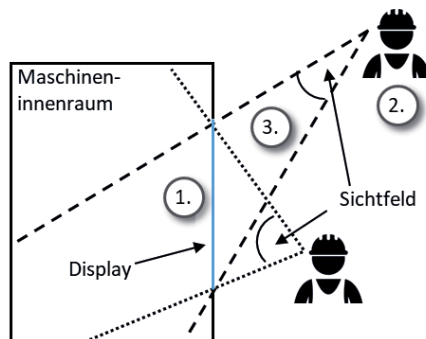


Abbildung 6-10: Schritte zur Berücksichtigung der Betrachterperspektive (Quelle: ISW)

6.5.1 Nachbildung des realen Displays durch ein virtuelles Display

Der sichtbare Bereich des Maschineninnenraums wird durch ein Fenster und damit auch durch ein zur Imitation oder Überlagerung dieses Fensters verwendetes Display begrenzt. Zur Nachbildung der Betrachterperspektive ist es daher notwendig die Maße, die Position und die Orientierung dieser Displays zu kennen. Dazu wird das reale Display, unabhängig ob es sich

dabei um ein herkömmliches oder ein transparentes Display handelt, im Computermodell nachgebildet. Da im Computerkameramodell üblicherweise die Z-Achse des Kamerakoordinatensystems in oder entgegen der Blickrichtung der Kamera liegt [8], wurde dies ebenfalls für das lokale Koordinatensystem zur Beschreibung des Displays festgelegt. Die Bildebene des Displays liegt folglich in der lokalen X/Y-Ebene und orthogonal zu Z-Achse. Zur Beschreibung des Displays hinsichtlich Position, Orientierung und Maße im globalen Koordinatensystem sind nur zwei diagonal gegenüberliegende Ecken notwendig. Alternativ können jedoch auch die Position einer Ecke, die Orientierung im Raum sowie Höhe und Breite angegeben werden. Abbildung 6-11 zeigt die Definition des lokalen Koordinatensystems eines Displays sowie die Nachbildung eines realen Displays vor dem Maschineninnenraum im Computermodell.

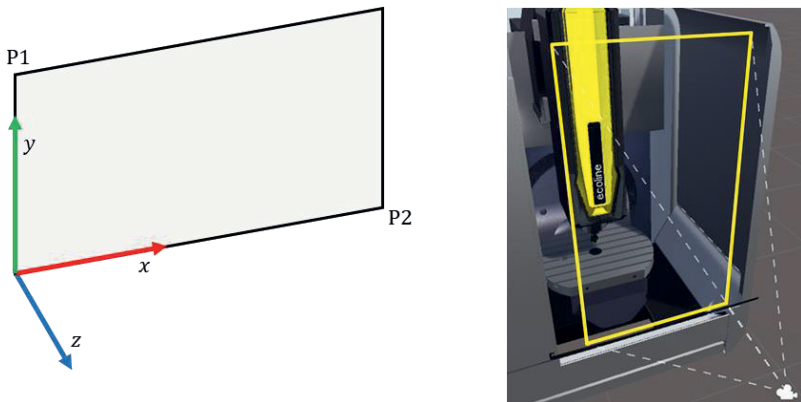


Abbildung 6-11: Nachbildung des Displays als virtuelles Objekt (Quelle: ISW)

6.5.2 Erfassung der Position des Betrachters mittels Head- bzw. Eye-Tracking

Wie in Kapitel 6.4.1 bereits erläutert entsteht durch das Kameramodell ein pyramidenförmiger Abbildungsraum dessen Spitze an der Kameraposition liegt. Bei Verwendung transparenter Displays und einer von der Position des Betrachters abhängigen perspektivischen Darstellung trifft dieses Modell, für jedes Auge separat, ebenfalls zu. Das transparente Display begrenzt den dahinterliegenden sichtbaren Raum und führt daher zu einer rechteckigen Grundfläche. Je ein Auge des Betrachters bildet die Spitze des pyramidenförmigen Abbildungsraums (Augenpunkt) und entspricht daher der Position der Kamera im Computermodell. Um diese Position im Computermodell entsprechend nachbilden zu können, müssen die Positionen der Augen des Betrachters im globalen Koordinatensystem bekannt sein. Diese Positionen können mittels Head- bzw. Eye-Tracking sensorisch erfasst werden. Je nach verwendetem System werden dabei die Augenpositionen entweder direkt (Eye-Tracking) oder durch Erfassung der Kopfposition des Betrachters (Head-Tracking) ermittelt. Verschiedene Sensorik für Head- bzw. Eye-Tracking ist kommerziell verfügbar und zeichnet sich durch unterschiedliche

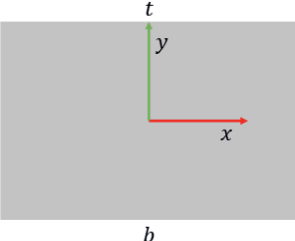
Eigenschaften aus. Die Anbindung der gewählten Sensorik an die Visualisierung erfolgt über einen austauschbaren Basisdienst im Mehrwertdienstframework (vgl. Kapitel 4). Für jedes Sensorsystem wird ein spezifischer Basisdienst verwendet, die jedoch alle die aktuelle Augenposition im gleichen Ausgangsformat zur Verfügung stellen.

Im Rahmen des Projekts wurden zwei Lösungen betrachtet, eine Marker-basierte und eine Marker-lose. Das Marker-basierte System SMARTTRACK der Firma ART zeichnet sich durch eine hohe maximale Messfrequenz von 60 Hz, eine hohe Genauigkeit und einen großen Arbeitsraum aus. Nachteilig sind die notwendigen Marker die z.B. mittels einer Brille am Betrachter angebracht werden müssen, sowie die daraus resultierende Transformation zwischen Markerposition und Augenposition. Die Vor- und Nachteile des Marker-losen Systems Kinect der Firma Microsoft stellen sich genau umgekehrt dar. Das System ist nicht nur in der Lage Körper und Köpfe von Personen, sondern auch deren Gesichtsmerkmale wie Augen, Mund und Nase vollständig ohne Marker zu erfassen. Ungünstigerweise erfolgt dies jedoch auf Kosten einer geringeren maximalen Messfrequenz von 30 Hz und eines etwas kleineren Arbeitsraumes. Auch hinsichtlich der referenzierten Koordinatensysteme unterscheiden sich die beiden Lösungen. Während die Microsoft Kinect Positionen grundsätzlich in Bezug zu einem eigenen, auf den Sensor zentrierten Koordinatensystem ausgibt, ermöglicht das ART SMARTTRACK die Einrichtung eines Koordinatensystems mit beliebigen Ursprung. In jedem Fall muss jedoch eine Koordinatentransformation in das Maschinenkoordinatensystem, in dem auch das Display sowie die realen und virtuellen Objekte im Maschineninnenraum vorgehalten werden, hergestellt werden.

6.5.3 Berechnung der zur Darstellung benötigten perspektivischen Parameter

Ausgehend von den Informationen über das Display, das den dahinterliegenden sichtbaren Raum begrenzt, und der Position des Auges des Betrachters, das die Spitze des pyramidenförmigen Abbildungsraums bildet, kann die notwendige zentralprojektive Abbildung berechnet werden. In der Computergrafik wird die zentralprojektive Abbildung mittels einer perspektivischen Projektionsmatrix beschrieben. In gängigen grafischen Programmierschnittstellen wie DirectX und OpenGL, aber beispielsweise auch in Game Engines, lässt sich die perspektivische Darstellung über die Projektionsmatrix eines Kameramodells beeinflussen [6]. Die perspektivische Projektionsmatrix wird, wie in Abbildung 6-12 dargestellt, durch Gleichung (6.1) berechnet. In der projektiven Geometrie werden üblicherweise homogene Koordinaten eingesetzt, da sich auf diese Weise nicht nur Rotationen sondern auch Translationen durch Matrixoperationen abbilden lassen [10]. Schneidet die optische Achse der Kamera, d.h. deren Z-Achse, die Clipping Planes mittig, spricht man von einer On-Axis Projektion. Fallen die Schnittpunkte der optischen Achse mit den Ebenen der Clipping Planes jedoch nicht auf das Zentrum dieser, handelt es sich um eine Off-Axis-Projektion. [16] Die Projektionsmatrix wird für On- und Off-Axis Projektionen auf dieselbe Weise berechnet. In die Formel zur Berechnung der Projektionsmatrix gehen die

Abstände der Kanten der Near Clipping Plane relativ zur optischen Achse der Kamera, d.h. relativ zum Schnittpunkt der optischen Achse mit der Ebene der Near Clipping Plane, ein.



$$P = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (6.1)$$

Legende:

n : Near Clipping Plane; f : Far Clipping Plane

Abstand der l : linken / r : rechten / t : oberen / b : unteren Kante der Near Clipping Plane relativ zur optischen Achse.

Abbildung 6-12: Berechnung der perspektivischen Projektionsmatrix (Quelle: ISW)

Da bei der perspektivischen Darstellung von dahinterliegenden Inhalten auf einem herkömmlichen oder transparenten Display keine Objekte vor dem Display dargestellt werden sollen, kann die Near Clipping Plane in die Displayebene gelegt werden. Des Weiteren wird die optische Achse der Kamera parallel zur Z-Achse des Displays und damit orthogonal zur Displayebene ausgerichtet. Für den Fall, dass sich der Betrachter exakt zentral vor dem Display befindet, entsteht eine On-Axis Projektion in der die optische Achse der Kamera die Displayebene genau im Zentrum schneidet. Alle anderen Betrachterpositionen führen zu einer Off-Axis Projektion, wie im Beispiel in Abbildung 6-13 skizziert.

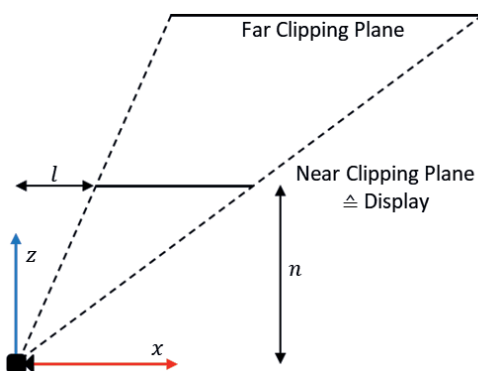


Abbildung 6-13: Perspektivische Off-Axis-Projektion (Quelle: ISW)

Für eine stereoskopische Darstellung virtueller Inhalte muss jedes Auge durch ein eigenes Kameramodell nachgebildet werden. Hierzu sind die zur perspektivischen Visualisierung

notwendigen Schritte 2 (Eye-Tracking) und 3 (Berechnung der Projektionsmatrix) für jedes Auge durchzuführen. Die Nachbildung des realen Displays durch ein virtuelles Display in Schritt 1 muss nur einmal erfolgen und kann zur Berechnung beider Projektionsmatrizen herangezogen werden. Die Erfassung beider Augen wird üblicherweise durch einen Sensor in einem Messzyklus erfolgen. Dennoch werden als Ausgangsdaten Positionskoordinaten für jedes Auge benötigt.

6.6 Literaturverzeichnis

- [1] Milgram, Paul ; Takemura, Haruo ; Utsumi, Akira ; Kishino, Fumio: *Augmented Reality: A class of displays on the reality-virtuality continuum*. In: *Proceedings of SPIE* (1994)
- [2] Sherman, William R. ; Craig, Alan B.: *Understanding virtual reality : Interface, application, and design*. San Francisco : Morgan Kaufmann, 2003 (The Morgan Kaufmann series in computer graphics and geometric modeling)
- [3] Azuma, Ronald T.: *A Survey of Augmented Reality*. In: *Presence: Teleoperators and Virtual Environments* (1997)
- [4] *OpenSceneGraph*. URL <http://www.openscenegraph.org/> – Überprüfungsdatum 2018-01-15
- [5] Sommer, Philipp ; Botsch, Daniel: *Leitfaden zur Nutzung von CAD-Modellen im Maschinen- und Anlagenbau für innovative Virtual Reality Konzepte*. 05/2017
- [6] Gregory, Jason: *Game Engine Architecture*. Wellesley, Massachusetts : A K Peters, Ltd., 2009
- [7] Hirsch, Andreas ; Hellmich, Arvid: *Aufbau und Auslegung funktionsbestimmender Werkzeugmaschinen-Baugruppen*. In: Neugebauer, Reimund (Hrsg.): *Werkzeugmaschinen : Aufbau, Funktion und Anwendung von spanenden und abtragenden Werkzeugmaschinen*. Berlin, Heidelberg : Springer, 2012 (VDI-Buch), S. 289–461
- [8] Süße, Herbert ; Rodner, Erik: *Bildverarbeitung und Objekterkennung : Computer Vision in Industrie und Medizin*. Wiesbaden : Springer Vieweg, 2014
- [9] Bosch, Siegfried: *Lineare Algebra*. 5., überarb. u. erweiterte Aufl. 2014. Berlin, Heidelberg : Springer Spektrum, 2014 (Springer-Lehrbuch)
- [10] Lengyel, Eric: *Mathematics for 3D Game Programming and Computer Graphics*. 3rd ed. Boston, MA : Cengage Learning, 2012
- [11] Fücke, Rudolf ; Kirch, Konrad ; Nickel, Heinz: *Darstellende Geometrie für Ingenieure*. 15. Aufl. Leipzig, Köln : Fachbuchverlag, 1993 (Lehrbücher der Mathematik)

-
- [12] Azuma, Ronald T.: *Augmented Reality : Approaches and Technical Challenges*. In: Barfield, Woodrow; Caudell, Thomas (Hrsg.): *Fundamentals of wearable computers and augmented reality*. Mahwah, NJ : Lawrence Erlbaum Associates, 2001, S. 27–63
 - [13] Behringer, Reinhold: *Augmented Reality*. In: Kent, Allen; Williams, James G. (Hrsg.): *Encyclopedia of Computer Science and Technology : Volume 45 - Supplement 30*. New York, US : Marcel Dekker Inc, 2002, S. 45-57
 - [14] Ehrmann, Andrea ; Błachowicz, Tomasz: *Das menschliche Auge*. In: *Physik in unserer Zeit* 46 (2015), S. 136–139
 - [15] Sommer, Philipp; Verl, Alexander: *View-dependent Virtual and Augmented Reality for Machine Tools*. In: *2016 European Modelling Symposium (EMS)* : IEEE, 2016, S. 149–154
 - [16] Bimber, Oliver ; Raskar, Ramesh: *Spatial augmented reality : Merging real and virtual worlds*. Wellesley Mass. : A K Peters, Ltd., 2005

7 Kalibrierung von Kameras in Werkzeugmaschinen

W. Schotte, P. Sommer

7.1 Einleitung und Zielsetzung

Allgemeine Kernaufgabe der Bildaufnahme ist die Projektion von Objekten in den Bildraum der verwendeten Kamera. Das heißt, Objekte bzw. Punkte im Weltkoordinatensystem werden über eine Optik auf einen Kamerasensor abgebildet, durch diesen ausgelesen und damit in das Koordinatensystem der Kamera projiziert. Die optischen Systeme als auch der mechanische Aufbau realer Kameras führen zu Abbildungsfehlern, die beispielsweise durch eine geometrische Verzerrung des Objektivs oder aber den nicht exakt zentrisch und plan sitzenden Kamerasensor hervorgerufen werden. So entstehen radiale und tangential Verzerrungen wie Kissen- oder Tonnenverzerrung sowie Längsverzerrung. Weitere Fehler können durch zum Beispiel partielle Linsenfehler günstiger Kunststofflinsen oder ungenügende Vergütung eben dieser führen, so dass lediglich ein Ausschnitt im Bild fehlerhaft abgebildet wird.

Der Vorgang der Kamerakalibrierung lässt sich in zwei Kalibrieraufgaben unterteilen. Die intrinsische Kamerakalibrierung hat zur Aufgabe, eben die genannten Abbildungsfehler von Weltkoordinatensystem auf Bildraum der Kamera nach Bildaufnahme zu bestimmen. Dieser Schritt wird ergänzt durch die extrinsische Kamerakalibrierung, die zur Aufgabe hat, die vermeintliche Kameraposition im Weltkoordinatensystem zu ermitteln und damit die Projektionsmatrix der Kamera zu vervollständigen. Man kann sich vorstellen, dass dies insbesondere bei Mehrkamerasystemen von Bedeutung ist, wenn bestimmte Punkte im Weltkoordinatensystem durch mehrere Kameras abgebildet werden und Position und Lage dieses Punktes zurückgerechnet werden soll.

Ziel der Kalibrierung von Kameras in Werkzeugmaschinen ist es, ein Verfahren zu entwickeln, dass eine möglichst automatisierte intrinsische und extrinsische Kamerakalibrierung in der besonderen Umgebung der Arbeitsräume von Werkzeugmaschinen realisiert. Da sich die Position als auch die Einstellungen der Kamera durch äußere Einwirkung wie Vibration oder Nutzereingriff verändern kann, ist die Einbettung des vorgeschlagenen Kalibrierverfahrens in die Maschinenbedienung zudem von besonderer Bedeutung.

7.2 Grundlagen der Kamerakalibrierung

Typische Digitalkameras bilden durch die Optik der Kamera 3-dimensionale Weltpunkte $P = (X, Y, Z)$ auf den 2-dimensionalen Kamerasensor und damit den Bildpunkt (u, v) ab (Abbildung 7-1). Wie bereits angesprochen, ist die Aufgabe der Kamerakalibrierung die Bestimmung der Projektionsmatrizen M_k für jede im Kamerasystem eingesetzte Kamera k . Die bestimmte Projektionsmatrix M gestattet somit die Berechnung des Punktes (u, v) bei bekannten P sowie die Berechnung eines Sichtstrahls ausgehend vom Kameraursprung F_c durch den gegebenen Punkt (u, v) und den unbekannten Punkt P .

Kameramodell

Je nach verwendeter Optik werden unterschiedliche Kameramodelle eingesetzt, um die angesprochene Projektionsmatrix auf Grundlagen der besonderen Charakteristiken in der Bildaufnahme aufzustellen. Insbesondere Kamerasysteme mit stark verzerrenden Optiken, wie zum Beispiel extreme Weitwinkel- oder Tilt-und-Shift-Objektive bedürfen spezieller Modelle, da diese Verzerrungen gesondert behandelt werden müssen.

Die im Projekt genutzten Kamerasysteme [1, 2] verwenden bezogen auf die Sensorgröße eine Standard- bzw. Normalbrennweite, so dass das vereinfachte Lochkameramodell eine ausreichende Näherung zur Berechnung der Kameraprojektion im ersten Schritt darstellt (siehe Abbildung 7-1).

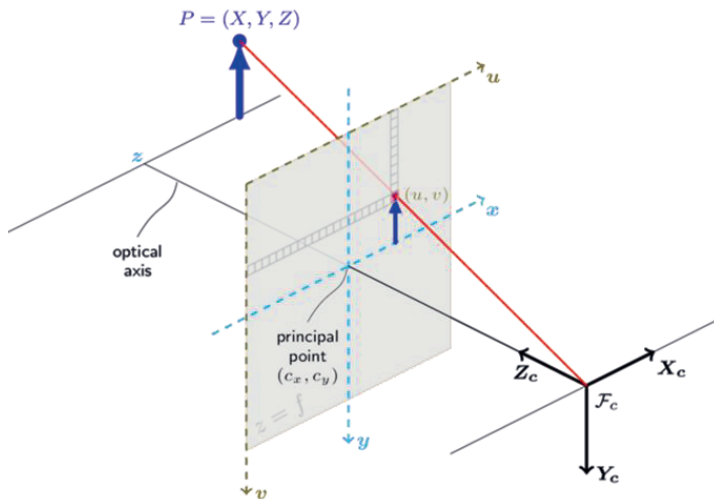


Abbildung 7-1: Lochkameramodell in OpenCV (Quelle: opencv.org)

Die Projektionsmatrix M für das Lochkameramodell lässt sich als perspektivische Projektion wie folgt darstellen.

$$M = A[R|t] \quad \text{mit} \quad A, R \in \mathbb{R}^{3 \times 3}, t \in \mathbb{R}$$

Hierbei beschreibt A die Kameramatrix mit den intrinsischen Parameter. Die Matrix A ist szenenunabhängig und muss initial für jede Kamera einmal bestimmt werden, solange Fokus und Zoom der Kamera fix eingestellt sind.

$$A = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (7.1)$$

f_x und f_y ist die Brennweite bezüglich der Sensordimensionen in Pixeleinheit. Der Punkt (c_x, c_y) ist die Bildmitte auf dem Bildsensor und ist ebenfalls in Pixeleinheiten angegeben. $[R|t]$ die kombinierte Rotations- und Translationsmatrix und enthält die extrinsischen Parametern.

$$[R|t] = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix}$$

$[R|t]$ beschreibt die Position t und Orientierung r der Kamera in der Szene. Ist die Kamera raumfest, so ist $[R|t]$ ebenfalls initial einmalig zu bestimmen. Bei bewegter Kamera beschreibt $[R|t]$ die Bewegung der Kamera in der Szene bzw. bei raumfester Kamera die bewegte Szene oder Objekte relativ zur Kamera.

Unter Verwendung von M und dem Skalierungsfaktor s bildet die Abbildungsfunktion des vereinfachten Lochkameramodells den Punkt $P=[X,Y,Z,1]$ in homogenen Koordinaten auf den Punkt $(u,v,1)$ ab.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = s \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (7.2)$$

Der Skalierungsfaktor s ist insbesondere dann von Bedeutung, wenn beispielsweise bei der 3D-Rekonstruktion absolute Größen bzw. Abstände rekonstruiert werden sollen oder multi-Kamerasystem mit verschiedenen bzw. verschieden eingestellten Kameras zum Einsatz kommen. Der Faktor s ist in diesem Fall ebenfalls für jede einzelne Kamera individuell zu bestimmen.

Mit der Projektion aus (7.2) ist es nun möglich Punkte im Weltkoordinatensystem in das Kamerakoordinatensystem umzurechnen. Nach Transformation von P mittels $[R|t]$ durch

$$[R|t]P = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = p \quad (7.3)$$

und Umkehrung der homogenen Koordinaten $x' = x/z$ und $y' = y/z$ ergibt sich die Abbildung wie folgt.

$$\begin{aligned} u &= f_x \cdot x' + c_x \\ v &= f_y \cdot y' + c_y \end{aligned} \quad (7.4)$$

Die Abbildung in Gleichung (7.4) bildet also die Punkte der Szene nach Transformation mittels $[R|t]$ aus dem Weltkoordinatensystem in den Bildraum der Kamera ab unter Berücksichtigung der Lage des Sensors auf der optischen Achse (c_x, c_y) und der Brennweite (f_x, f_y) . Verzerrungen

werden im Modell soweit noch nicht berücksichtigt. Das Standardkameramodell in OpenCV [3] erweitert das Lochkameramodell um weitere 8 Parameter, die Koeffizienten k_1, \dots, k_6 , die die radiale Verzerrung beschreiben, sowie die Koeffizienten p_1 und p_2 , die die tangentiale Verzerrung darstellen.

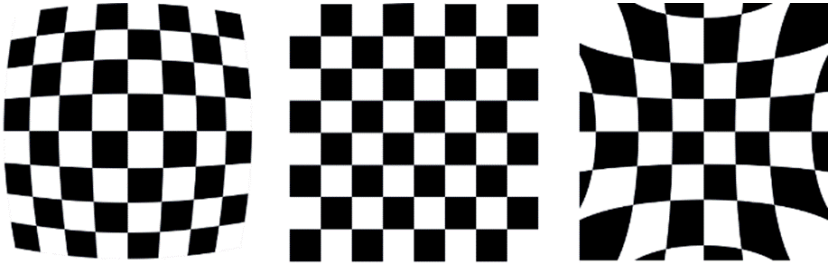


Abbildung 7-2: Tonnenverzerrung $k_1 > 0$ (links), ohne Verzerrung (Mitte), Kissenverzerrung $k_1 < 0$ (rechts) (Quelle: opencv.org)

Die Erweiterung des OpenCV Kameramodells [3] basieren wiederum auf der Projektion (7.2), der Abbildung von P nach p mittels $[R/t]$ (7.3) sowie der Umkehrung der homogenen Koordinaten $x' = x/z$ und $y' = y/z$. Im Anschluss werden die Koeffizienten k_1, \dots, k_6 sowie p_1 und p_2 wie folgt gebraucht.

$$\begin{aligned} x'' &= x' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \\ y'' &= y' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \end{aligned} \quad \text{mit } r^2 = x'^2 + y'^2$$

Die Abbildung nach (u, v) in den Bildraum der Kamera ergibt sich dann wie folgt.

$$\begin{aligned} u &= f_x \cdot x'' + c_x \\ v &= f_y \cdot y'' + c_y \end{aligned} \quad (7.5)$$

Bemerkenswert ist, dass die Koeffizienten als konstant angenommen werden können, während die Auflösung des Bildes variiert werden kann oder Ausschnitte festgelegt werden können, lediglich die optischen Achse (c_x, c_y) und die Brennweite (f_x, f_y) müssen gegebenenfalls neu berechnet bzw. angegeben werden. Werden die Parameter der Optik geändert, wird zum Beispiel neu fokussiert oder die Brennweite eines Zoomobjektivs verändert, sind auch die Koeffizienten neu zu bestimmen, da die Verzerrungen sich nicht linear über die Einstellbereiche ändern und unterschiedliche Verzerrungen in Erscheinung treten können.

Intrinsische Kalibrierung

Durch die intrinsische Kalibrierung wird versucht, die Matrix A (7.1) als auch die Koeffizienten k_1, \dots, k_6 sowie p_1 und p_2 zu bestimmen. Gängige Verfahren bestehen im Kern darin, zunächst unterstützt durch bekannte Muster oder Geometrien im Bild möglichst viele Punktepaaire $((X, Y, Z), (u, v))$ zu bestimmen (siehe Abbildung 7-3), um im zweiten Schritt das

Gleichungssystem (7.5) für alle Punktpaare aufzustellen.

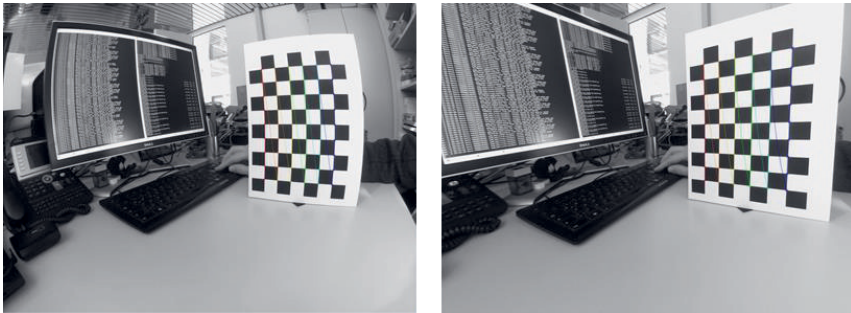


Abbildung 7-3: Verzernte Weitwinkelaufnahme (links), gleiches Bild entzerrt (rechts) (Quelle: HLRS)

Im Anschluss wird versucht, das Gleichungssystem zu lösen bzw. numerisch zu approximieren, da eine exakte Lösung des Gesamtsystems beispielsweise aufgrund der diskreten Bildauflösung oft nicht möglich ist. Die radiale und tangentielle Verzerrung veranschaulicht Abbildung 7-4.

Die Bildmitte des Sensors ist durch das Kreuz markiert, während die Zentren der radialen bzw. tangentialen Verzerrungen durch den Kreis markiert sind. Sehr schön zu sehen ist, dass hier die optische Achse des Objektivs nicht mit der des Kamerasensors übereinstimmt und diese die tatsächliche Bildmitte des Sensors nicht durchkreuzt. Wie bereits erwähnt, werden durch das Verfahren radiale und tangentielle Verzerrungen bestimmt, punktuelle bzw. partielle Verzerrungen werden aber nicht betrachtet.

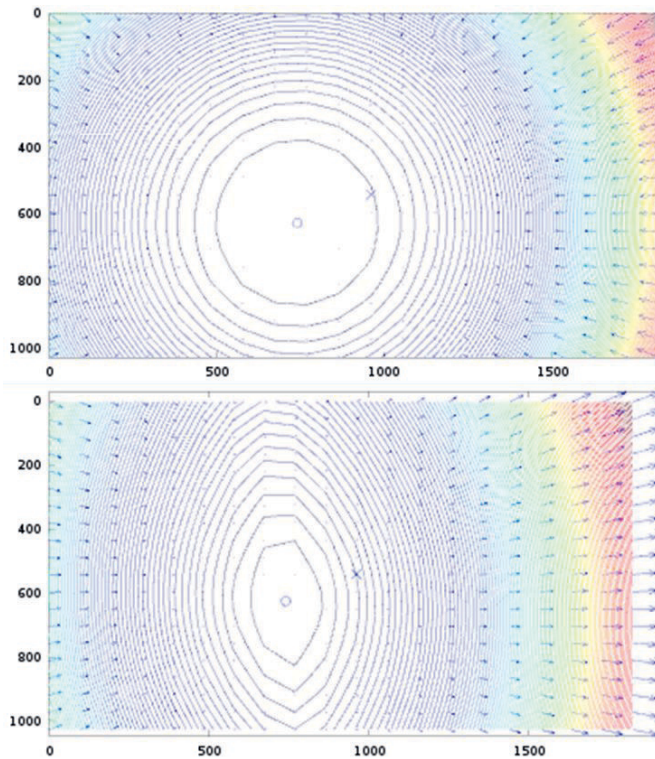


Abbildung 7-4: Visualisierung der radialen (oben) und tangentialen (unten) Verzerrung
(Quelle: HLRS, Camera Calibration Toolbox for Matlab by Caltech)

Extrinsische Kalibrierung

Die extrinsische Kalibrierung hat zur Aufgabe, die Matrix $[R/t]$ aus (7.3) zu bestimmen. Im Prinzip ist das die Bestimmung der Transformation der Lage und Orientierung der Kamera relativ zum Koordinatensystem, das der Marker aufspannt bzw. zum Ursprung des Weltkoordinatensystems, wenn die Position und Orientierung des Markers bekannt ist. Je nach Kalibrierverfahren, kann die Bestimmung der extrinsischen Kameraparameter gleichzeitig zur Bestimmung der intrinsischen Kameraparameter durch Nutzung derselben Punktepaaire erfolgen. So erlaubt zum Beispiel die Kalibrierung per Schachbrettmuster (Abbildung 7-3) in OpenCV auch die Bestimmung der Kameraposition relativ zu den Schachbrettmustern (Abbildung 7-5 links).

Bleibt das Objektiv in der gleichen Konfiguration, verändern sich also die intrinsischen Parameter nicht und ändert sich lediglich die Kameraposition, kann die Bestimmung der extrinsischen Parameter auch in einem separaten Schritt erfolgen. Hierfür können auch

anders geartete Marker eingesetzt werden, wie sie beispielsweise beim Objekttracking in der Erweiterten Realität genutzt werden (Abbildung 7-5 m.).

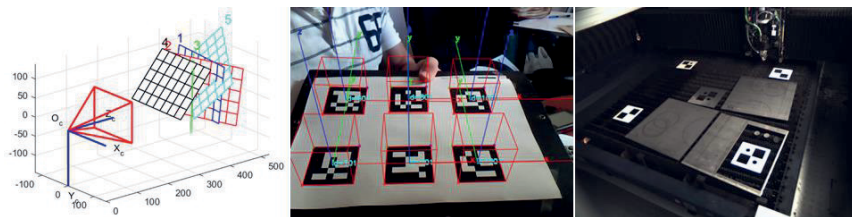


Abbildung 7-5: Extrinsische Kameraparameter mit Schachbrett (links), AR-Marker (m.), Punktmarker (rechts) (Quellen v.l.n.r.: Brown University; OpenCV aruco; TRUMPF)

Wie angesprochen, ist bei gegebenen intrinsischen Parametern lediglich die Bestimmung von hinreichend vielen Punktpaaren Weltkoordinatensystem zu Kamerakoordinatensystem zur Lösung des Gleichungssystems (7.3) notwendig. Wenn alle Punkte im entsprechend Koordinatensystem präzise bestimmt werden können, sind im besten Fall sogar nur vier Punktpaare notwendig (vgl. Kapitel 7.4.4). Aufgrund der diskreten Bildräume, Unschärfe im Bild oder komplexer Verzerrung werden in der Regel mehr Punkte genutzt, um die exakte Lösung besser zu approximieren. Marker, die helfen, diese Punkte zu bestimmen können entsprechend der Anwendung bzw. des genutzten Featuredetektors gestaltet werden (Abbildung 7-5 rechts).

3D-Kamera Kalibrierung

3D-Kameras bilden im Allgemeinen die Szene ebenfalls durch eine entsprechende Optik auf einen 2-dimensionalen Bildsensor ab. Zusätzlich werden Tiefeninformationen beispielsweise mit Hilfe eines Time-of-Flight-Sensors (ToF-Sensor) oder eines strukturierten Musters, welches in die Szene projiziert wird, gemessen bzw. errechnet. Das heißt, dass die intrinsische Kalibrierung des 2-dimensionalen Kamera bzw. ToF-Sensors regulär erfolgen kann, das Kameramodell muss aber um die Tiefendaten des entsprechenden Tiefensensors erweitert werden. Dies erfordert die zusätzliche Kalibrierung der Tiefendaten.

Die Notwendigkeit der zusätzlichen Kalibrierung liegt in der Vielzahl an physikalischen und systemtechnischen Effekten begründet, die die Messgenauigkeit der Tiefenwerte beeinträchtigen [4]. Das grundlegende Prinzip einer ToF-Kamera besteht in der Aussendung eines modulierten Infrarot-Lichtimpulses und der Laufzeitmessung des Lichtes von Aussendung bis zur Aufnahme des Lichtimpulses durch den Kamerasensor. Die Laufzeitmessung selbst geschieht durch reine Zeitmessung oder durch Überlagerung der Modulation und Bestimmung der Phasenverschiebung von ausgehendem und aufgenommenem Licht (Abbildung 7-6).

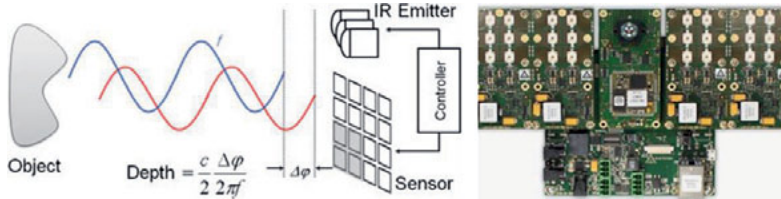


Abbildung 7-6: Prinzip der Distanzmessung mit ToF (links), ToF-Kamera mit Sensor und LED-Modulen (rechts) (Quellen v.l.n.r.: Hyunjung Shim, Seungkyu Lee; BlueTechnix)

Messfehler sind durch das Aufnahmesystem als auch durch die Effekte verursacht, wie zum Beispiel durch unterschiedliche Reflektivität oder ungewollte Überlagerungen. So zeigt sich beispielsweise, dass unterschiedliche Farben und Materialien das von der Kamera ausgesendete Licht unterschiedlich stark sowie mit leichter Phasenverschiebung reflektieren oder aber sich die Wellenpakete durch Überlagerung verschieden tief reflektierter Wellenpakete aufweiten (Abbildung 7-7 links).

In Studien zu Messfehlern der ToF-Sensoren hat sich auch gezeigt, dass die Messfehler über die Distanz des Messobjekts zur aufnehmenden Kamera nicht linear sind [5]. Zudem kommen weitere systembedingt Messfehler, wie sie beispielsweise durch Reflexionen der Lichtimpulse in der Optik der Kamera entstehen. Hier kommt es wiederum zu Überlagerungseffekten, die die Bestimmung der Phasenverschiebung erschweren bzw. mit einem Fehler behaften.

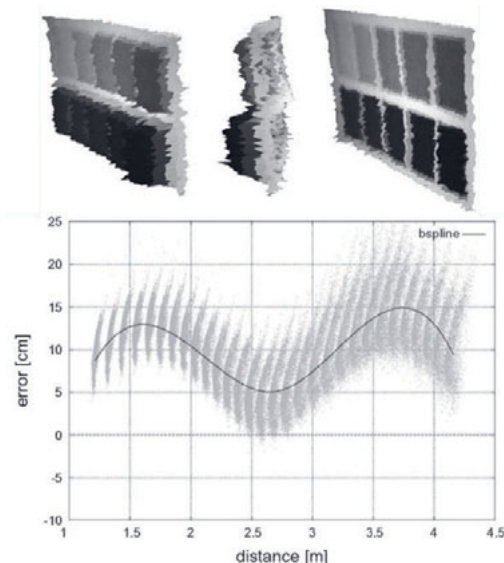


Abbildung 7-7: Messfehler der ToF-Sensoren (Quelle: [5])

Zur Kalibrierung der über die Tiefe nicht linearen Fehler wurde gemeinsam mit TRUMPF ein 3D-Kalibrierblech in der Größe 50x50 cm gefertigt (vgl. Kapitel 7.4.3), das aus zwei Blechlagen besteht. Das der Kamera zugewandte Blech ist mit rechteckigen 10x10 cm großen Löchern versehen, die sich an einem Schachbrettmuster orientieren. Das hintere Blech ist ohne Löcher gefertigt (Abbildung 7-8).

Beide Bleche sind über Gewindestangen miteinander verschraubt, so dass sich ein definierter Abstand der beiden Bleche bis ca. 20 cm einstellen lässt. Dieses 3D-Kalibrierblech kann händisch durch den Erfassungsbereich der Kamera geschwenkt werden, so dass die bekannte Distanz Δd der beiden Bleche durch die Kamera gemessen werden kann. Eine eventuelle Verkipfung des Bleches bei Handführung ist bei kalibrierten IR-Sensor feststellbar und kann herausgerechnet werden. Diese Methode ermöglicht eine Kalibrierung der Kamera ohne Messstand und Messung absoluter Entfernungen zum Messobjekt.

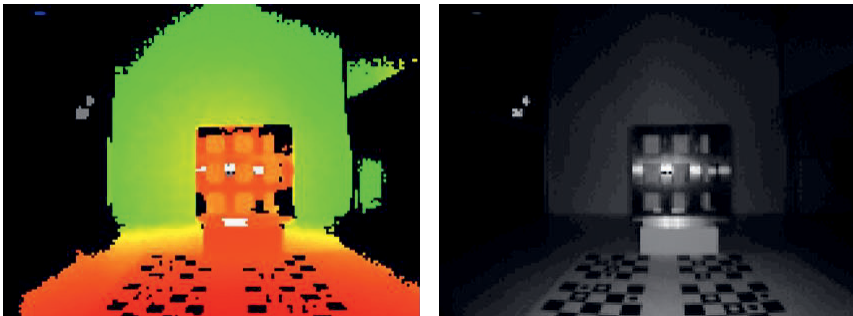


Abbildung 7-8: 3D-Kalibrierblech, Tiefenwerte (links) und IR-Bild (rechts) (Quelle: HLRS)

7.3 Anforderungen

Im Projekt vorgesehen ist die Anbringung der Kameras innerhalb der Werkzeugmaschinen, so dass diese den Bauraum und die darin positionierten Werkstücke optimal erfassen können. Gleichzeitig müssen die Kameras so angebracht werden, dass sie beispielsweise vor abgespanntem Material, Funkenflug oder Kühl- und Schmiermittel geschützt werden. Dieser Abschnitt betrachtet die wesentlichen Voraussetzungen für die Kamerakalibrierung in Werkzeugmaschinen insbesondere der, die im Projekt als Demonstrator vorgesehen sind bzw. durch die Projektpartner zur Verfügung gestellt wurden.

TRUMPF TruLaser 5030

Die Laserschneidmaschine TruLaser 5030 des Projekt-partners TRUMPF repräsentiert den 2D Anwendungsfall (vgl. Kapitel 2.3), in dem der Bauraum auf eine gegebene Bearbeitungsebene festgelegt ist. Der Ursprung des Weltkoordinatensystems liegt hier in der Bearbeitungsebene auf Höhe der Palette, die das Blech trägt. Zur Bestimmung der Blechoberfläche ist die Materialstärke des gewählten Blechs zu berücksichtigen. Diese Parameter sind durch den Anwender vorgegeben und können über die Maschinensteuerung abgefragt werden.



Abbildung 7-10: TRUMPF TruLaser 5030
(Quelle: TRUMPF)

Bosch Rexroth: SW CNC-Fräsmaschine

Seitens Bosch Rexroth kommt eine 5-Achsen CNC Fräsmaschine der Firma Schwäbische Werkzeugmaschinen GmbH (SW) zum Einsatz (Abbildung 7-9). In diese Maschine ist eine HiPerCam I IP-Kamera der Firma Eltec Elektronik AG [2] verbaut, deren Bilder an die Steuerungssoftware übertragen werden.



Abbildung 7-9: SW CNC-Maschine
(Quelle: Bosch Rexroth)

DMG MORI DMU50 eco

Die 5-Achs CNC-Fräsmaschine DMU50 ecoMill von DMG MORI repräsentiert den im Projekt geplanten finalen Demonstrator. In diese werden die im Projekt realisierten Innovationen verbaut und verifiziert. Unter anderem ist ein Kamerasystem zur 3D-Rekonstruktion (Kapitel 10), das transparente Display als Maschinenfenster mit Trackingsystem und Benutzeroberfläche (Kapitel 6), die den Mehrwertdienst offeriert (Kapitel 4), ein opakes Display sowie eine in die Maschinensteuerung eingebettete Visualisierung verbaut, die unter anderem die Abtragsimulation demonstriert.



Abbildung 7-11: DMG MORI DMU50 ecoMill (Quelle: DMG MORI)

7.3.1 Arbeitsräume der Werkzeugmaschinen

Wie bereits die Außenansicht der im Projekt genutzten Werkzeugmaschinen zeigt (Abbildung 7-10, Abbildung 7-9, Abbildung 7-11), gibt es hier deutliche Unterschiede bezogen auf die Größe der Arbeitsräume als auch hinsichtlich der Sichtbarkeiten und der Montagemöglichkeiten für die Kameras.

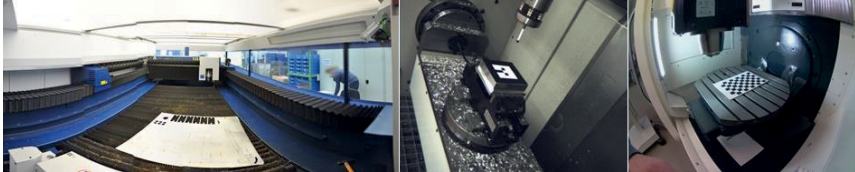


Abbildung 7-12: Arbeitsräume der Werkzeugmaschinen a) TRUMPF b) SW c) DMG MORI (v.l.n.r) (Quelle: HLRS, Bosch Rexroth, HLRS)

Der Arbeitsraum der Laserschneidmaschine TruLaser 5030 hat in der Ebene eine Größe von 2 m x 4,5 m (Abbildung 7-12 links). Vorgesehen sind an dieser Maschine mindestens zwei Eltec HiPerCam I IP-Kameras, welche im Dachbereich der Maschine angebracht werden, um die Wahrscheinlichkeit von Funkenflug auf das Schutzglas der Kameras zu minimieren. Die Kameras haben die Aufgabe, den Arbeitsraum der Maschine vollständig zu erfassen, um dem Benutzer an der Maschinenkonsole einen Gesamtüberblick über den Arbeitsraum sowie die eingebrachten Bleche zu ermöglichen. Es werden unterschiedliche Ausschnitte des Arbeitsraums mit geringer Überlappung im Randbereich der Bildausschnitte von oben erfasst. Nennenswert ist, dass die hier genutzte Maschine im Vergleich zu anderen TRUMPF Laserschneidmaschinen einen relativ kleinen Bauraum aufweist. TRUMPF bietet Laserschneidmaschinen an, die eine Bearbeitung von Blechen in einem Bauraum von größer 10 m Länge ermöglichen. Die Übertragung der Ergebnisse aus dem Projekt soll auch auf diese Maschinen, d.h. auf große Bauräume, möglich sein. Für diese Anwendung sind deutlich mehr Kameras erforderlich, um den vollständigen Bauraum zu erfassen. Die Sicht der Kameras auf die Palette ist im Ruhebetrieb der Maschine frei, da der Laserkopf üblicherweise in eine Parkposition am Ende der Maschine gefahren wird. In die TruLaser 5030 sind eine Beleuchtungsanlage sowie eine Deckenabdeckung verbaut, die eine gute Ausleuchtung des Arbeitsraums realisieren.

Die bei Bosch Rexroth eingesetzte SW Maschine verfügt über einen Bauraum von 600x400x400 mm (BxHxT) (Abbildung 7-12 m.). In der Maschine ist eine Eltec HiPerCam I Kamera im oberen Bereich des Innenraums fest montiert, um das Schutzglas der Kamera vor abgespanntem Material sowie zerstäubten Schmier- und Kühlmitteln zu schützen. Der Werkzeugkopf kann in eine Parkposition gefahren werden, in der das Werkzeug zwar im Kamerabild noch zu sehen ist, der Bauraum aber mit einem Normalobjektiv noch gut übersehen werden kann. Die Beleuchtung ist durch zwei Leuchtstofflampen einseitig entlang

der oberen Türöffnung realisiert und ermöglicht eine zufriedenstellende Erkennung von Schachbrettern oder Markern.

Der Bauraum der Universalfräsmaschine DMU50 ecoMill beträgt 500x400x450 mm (BxHxT) (Abbildung 7-12 rechts). In der Maschine sind für die 3D-Rekonstruktion 10 Kameras vorgesehen. Jeweils vier Stück werden in den 4 Ecken des Bauraums ungefähr auf Höhe des Werkstücks und ca. 40 cm darüber angebracht. Zwei weitere Kameras sollen das Werkstück aus Benutzersicht und aus Sicht des Werkzeugkopfs aufnehmen. Für den Demonstrator ist kein besonderer Schutz der Kameras geplant. Die Beleuchtung in der DMU50 ecoMill ist durch eine Leuchtstofflampe realisiert, die sich in der vorderen Ecke in Richtung des Werkzeugwechslers befindet. In der DMU50 ecoMill werden die Kameramodule oCam 5CRO-U [1] des koreanischen Herstellers hardkernel genutzt. Diese verfügen über ein Standard M12 Objektivgewinde, so dass je nach Maschinentyp unterschiedliche Objektive gewählt werden können. Um das Werkstück im Arbeitsraum der DMU50 ecoMill optimal erfassen zu können, wurden verschiedene Objektivbrennweiten getestet (Abbildung 7-13). Das 3,6 mm Standardobjektiv stellt einen guten Kompromiss bzgl. geringer Verzerrung und Bildausschnitt dar.

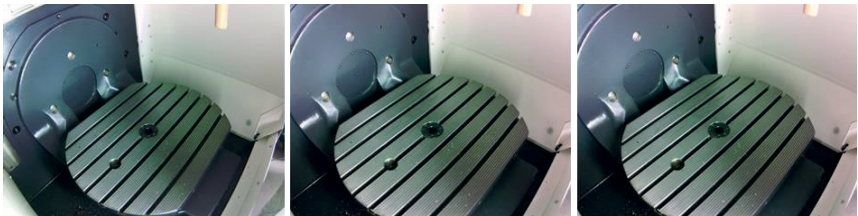


Abbildung 7-13: Auswahl einer geeigneten Brennweite (2,65 mm (links) 3 mm (m.) 3,6 mm (rechts) (Quelle: HLRS)

7.3.2 Bildaufnahme

In der SW Maschine bei Bosch Rexroth sowie in der Laserschneidmaschine bei TRUMPF ist der Kameratyp durch die Vorauswahl des Maschinenherstellers bereits durchgeführt. Hier kommen die bereits erwähnten Eltec HiPerCam I Kameras zum Einsatz [2]. Die HiPerCam ist eine IP-Kamera mit einer Auflösung von 5MP und einem rolling-Shutter. Die Kamera wird über Power-over-Ethernet (PoE) mit Strom versorgt. Zur Bildaufnahme werden durch die Kamera über die Anbindung durch Gigabit-Ethernet verschiedene Möglichkeiten angeboten. So ist unter anderem die Bildaufnahme durch ein einfaches Herunterkopieren des Bildes über Fileservices möglich oder aber, es kann ein mpeg Bildstrom konfiguriert werden, so dass die Kamera kontinuierlich Bilddaten an einen beliebigen Rechner im Netzwerksegment liefert. Die Konfiguration erfolgt über ein Webinterface oder durch http-POST-requests an die Kamera. Die Kamera verfügt über Schutzklasse IP54 und ein austauschbares Frontglas. Die Kamera ist in Ihrem aktuellen Gehäuse inklusive des Anschlusses auf der Rückseite der Kameras sehr lang.

In der DMU50 ecoMill sollen deutlich mehr Kameras verbaut werden. Der relativ kleine Arbeitsraum schränkt die Auswahlmöglichkeit an Kameramodulen ein, die ohne Modifikation des Maschinengehäuses oder Behinderung des Werkzeugkopfs in die Maschine verbaut werden können. Grundsätzlich bestehen hier zwei Anforderungen, es sollen mindestens 8 Kameras verbaut werden, um den Arbeitsbereich möglichst komplett abzudecken, und die Kameras sollen nicht im Arbeitsbereich des Werkzeugkopfs liegen. Diese Anforderungen führen direkt zur nächsten Anforderung, dass die Bildaufnahme mit hoher Auflösung von allen relevanten Kameras gleichzeitig nahe Echtzeitbedingungen realisiert werden muss, um Bewegungen im Arbeitsraum rekonstruieren zu können. Der Rohdatenstrom bei einer Auflösung 1920x1080 Pixel, 24bit Farbe und 15 Bildern pro Sekunde beträgt bereits ca. 750MBit/s. Ein Motion-JPEG-Stream, wie ihn beispielsweise die Eltec HiPerCam I anbietet erreicht bei noch hoher Bildqualität eine Kompression von ca. 1:5, was einen Bildstrom von ca. 150 Mbit/s bedeutet. Kompressionsartefakte können insbesondere für Featuredetektoren in der Bildverarbeitung zu einem Problem werden, während die hohen Datenraten der Rohdaten die Bandbreite des Gigabit-Ethernets bereits saturieren können.

7.3.3 Kalibrierungsprozess

Wie bereits in Kapitel 7.2 beschrieben, ist die Aufgabe der intrinsischen Kalibrierung, optische Verzerrungen beispielsweise des Objektivs zu bestimmen, und Aufgabe der extrinsischen Kalibrierung, die Position und Orientierung der Kamera relativ zu einem gegebenen Weltkoordinatensystems zu bestimmen. Eine Kalibrierung ist initial immer notwendig, wenn die Kamera in die Maschine verbaut wird. Eine nachträgliche Rekalibrierung ist genau dann notwendig, wenn sich das optische System oder die Position und Orientierung der Kamera ändert. Das optische System wird zum Beispiel durch eine Veränderung des Fokus oder der Brennweite geändert, die Lage der Kamera kann sich beispielsweise durch Vibration oder durch einen versehentlichen Stoß des Nutzers an die Kamera ändern. Das heißt, eine Kalibrierung der Kameras ist nicht nur zum Zeitpunkt des Aufstellens bzw. der Installation der Werkzeugmaschine oder des Systems notwendig, sondern kann potentiell auch zu späteren Zeitpunkten notwendig sein. Die Kalibrierung sollte somit möglichst einfach und durch den Nutzer durchführbar sein.

7.4 Evaluierung der Methoden zur Kamerakalibrierung

7.4.1 Kalibrierverfahren in OpenCV

Das Standardverfahren in OpenCV ermöglicht die intrinsischen als auch extrinsischen Kalibrierung von Kameras, die mit dem Lochkameramodell beschrieben werden können und nutzt Schachbrettmuster (Abbildung 7-14 links), um eine Mindestmenge an Punktpaaren, Objektpunkte im Weltkoordinatensystem zu Bildpunkten, auf dem Kamerasensor zu bestimmen (Kapitel 7.2). Die gesuchten Punkte sind die jeweiligen Berührungspunkte der schwarzen Quadrate untereinander. Durch den Anwender wird die Größe des Schachbretts,

das heißt die Anzahl dieser Kontaktpunkte in Höhe und Breite im Schachbrettmuster sowie die Kantenlänge eines solchen Quadrats, vorgegeben. Damit ist das Muster im Objektraum vorgegeben, durch Kantenextraktion und Subsampling lassen sich die Berührungspunkte der schwarzen Quadrate im Bildraum der Kamera gut detektieren.

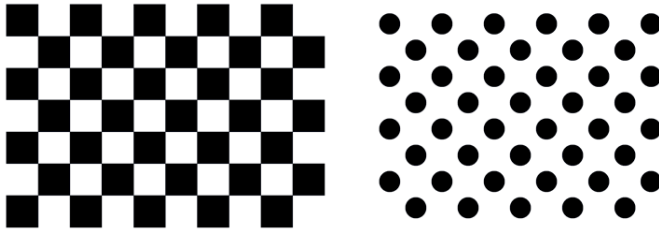


Abbildung 7-14: Beispiele für Kalibriermuster (Quelle: OpenCV)

Die Schachbretter werden in unterschiedlichen Positionen im Bild in unterschiedlichen Orientierungen aufgenommen und deren Punkte im Bildraum werden durch die Schachbretterkennung in OpenCV mittels `cv2.findChessboardCorners()` bestimmt. Die Genauigkeit wird durch die Anwendung der Methode `cv2.CornersSubPix()` nochmals verbessert. OpenCV empfiehlt die Aufnahme von mindestens 10 solcher Muster [6], gute Ergebnisse erhält man mit der Aufnahme von ca. 25-30 Bildern, indem die Muster in alle Bereiche des Kamerabildes gebracht werden, diese auch gekippt und in unterschiedlichen Entfernungen aufgenommen werden. Durch die Überbestimmung des Gleichungssystems zur Lösung von (7.2) können so innerhalb OpenCV auch numerische Optimierungen genutzt werden, die gefundenen Werte iterativ zu verbessern.

Eine weitere in OpenCV zur Verfügung gestellte Methode ist die Nutzung eines Musters bestehend aus Kreisen, die asymmetrisch auf der Fläche verteilt liegen (Abbildung 7-14 rechts). Anstatt der Berührungspunkte von Quadraten wird hier die Position des durch die Perspektive im Bild zur Ellipse verzerrten Kreises gesucht. Durch die elliptische Verformung als auch die perspektivische Verzerrung des Musters im Bild, kann dessen Lage durch die OpenCV Methoden `cv2.findCirclesGrid()` bestimmt werden. Das weitere Vorgehen entspricht dem obigen Verfahren unter Nutzung des Schachbrettmusters.

Die Qualität des Verfahrens lässt sich durch eine Reprojektion der Objektpunkte in den Bildraum mittels der bestimmten Parameter und den Vergleich dieser projizierten Punkte mit den Punkten, die der Algorithmus zur Punktbestimmung im Schachbrettmuster erkannt hat, ermitteln. Die normierte Distanz dieser Abweichung wird über alle Punkte arithmetisch gemittelt und man erhält eine gute Abschätzung des Fehlers in der Kalibrierung.

Das OpenCV Standardverfahren eignet sich für die meisten Kameraanwendungen, die das Lochkameramodell zugrunde legen. Der Kalibriervorgang ist jedoch entsprechend aufwendig und händisch durchzuführen. Am Beispiel des Bauraums der TRUMPF Laserschneidmaschine

(Abbildung 7-12 a)) muss der Anwender bei vormontierter Kamera in den Bauraum klettern bzw. sich auf die Palette setzen oder knien. Ein freies Bewegen des Kalibrieramusters ist bereits im relativ kleinen Arbeitsraum der SW Maschine sehr schwierig (Abbildung 7-12 b)). Lediglich die DMU50 ecoMill ermöglicht durch die Zugänglichkeit und Größe des Bauraums eine halbwegs brauchbare Kalibrierung (Abbildung 7-15).



Abbildung 7-15: Verzerres und entzerrtes Kamerabild mit Schachbrett in DMU50 ecoMill (Quelle: HLRS)

Die oft kleinen Arbeitsräume der Werkzeugmaschinen machen das Verfahren zur Kamerakalibrierung mit Hilfe solcher Muster unpraktisch, insbesondere dann, wenn die Kameras mehrfach intrinsisch oder extrinsisch neu kalibriert werden müssen.

7.4.2 MATLAB Camera Calibration Toolkit

MATLAB ist eine Programmierumgebung, die in Forschung und Industrie unter anderem für Entwicklung und Implementierung von Aufgaben im Bereich der Computer Vision eingesetzt wird. Das für MATLAB am California Institute of Technology entwickelte Camera Calibration Toolkit nutzt ebenfalls, wie die durch OpenCV direkt angebotene Kalibriermethodik, ein Schachbrettmuster, um Punkte im Objektraum den im Bildraum erkannten Punkte zuzuordnen.

Die grundlegende Funktionalität beider Umgebungen ist ähnlich. Das Camera Calibration Toolkit bringt von Haus aus eine Vielzahl an Optimierungseinstellung und Analysemöglichkeiten der Kalibrierung mit, so dass diese vom Anwender nicht selbst implementiert werden müssen. Beispielsweise ist die manuelle Vorselektion der Position der Eckpunkte des Schachbrettmusters möglich, was zum Beispiel in schlecht ausgeleuchteten Aufnahmen ein nützliches Werkzeug ist. Auch können die gewonnenen intrinsischen Parameter (Abbildung 7-4) als auch die extrinsischen Parameter (Abbildung 7-16 oben) visualisiert werden. Dies ermöglicht beispielsweise eine schnelle Identifizierung von schlecht oder fehlerhaft erkannten Mustern in den entsprechenden Aufnahmen, so dass die Kalibrierung verbessert werden kann. Auch der erzielte Reprojektionsfehler kann mit Hilfe des

Toolkits direkt visualisiert werden, das ermöglicht die Abschätzung der Qualität der Kalibrierungsparameter (Abbildung 7-16 unten).

Da aber auch das MATLAB Camera Calibration Toolkit Schachbrettmuster verwendet, ergeben sich bei Verwendung des Toolkits zur Kalibrierung von Kameras in Werkzeugmaschinen vergleichbare Vor- als auch Nachteile.

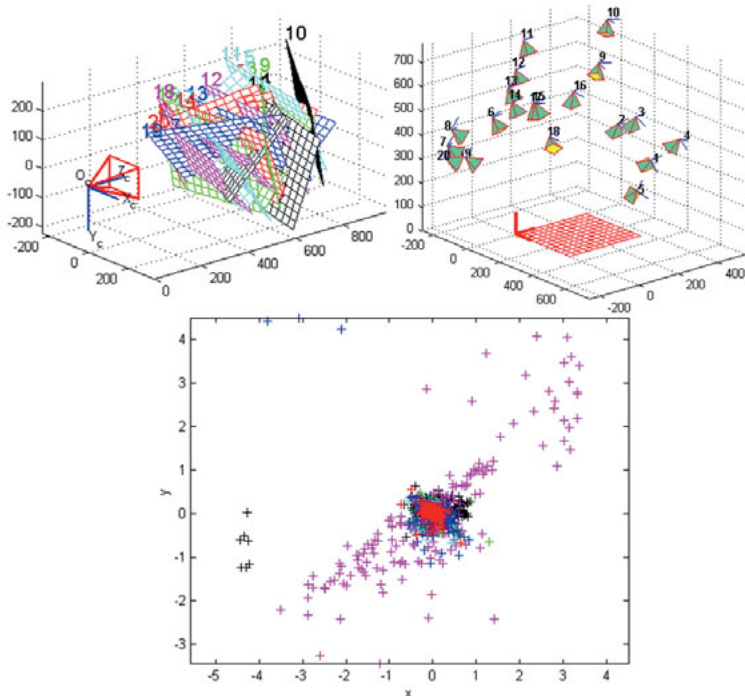


Abbildung 7-16: Auswertung eines Kalibrierungsvorgangs mit dem Matlab Camera Calibration Toolkit (Quelle: Computational Vision, Caltech)

7.4.3 Kalibrierung von ToF-Kameras

ToF-Kameras senden meist im Infrarotbereich Lichtimpulse aus und nehmen diese über einen Bildsensor auf, ähnlich der Technik in herkömmlichen 2D-Kameras. Das Infrarotbild, das diese Kameras üblicherweise zur Verfügung stellen, ist ebenso durch die verwendete Optik und Konstruktion bzw. Montage des Sensors verzerrt. Zusätzlich zu diesen Fehlern kommen die Fehler in der Tiefenmessung der entsprechenden Pixel des Sensors, die wiederum ganz unterschiedliche Fehlerursachen haben können (Abbildung 7-7, Abbildung 7-8).

Die Kalibrierung einer ToF-Kamera erfolgt in zwei Schritten, zunächst wird der 2D-Sensor kalibriert, danach folgt die Kalibrierung der Tiefenwerte. Zur Kalibrierung des 2D-Sensors

können die gleichen Methoden genutzt werden, die auch für 2D-Kameras genutzt werden (Kapitel 7.4.1 und 7.4.2). Die Korrektur der Tiefenwerte wird mit Hilfe eines 3D-Kalibrierblechs bestimmt (Abbildung 7-8). Diese ermittelt einen Messfehler in der Distanz der beiden Bleche relativ zur Kamera. Da beide Bleche fix zueinander montiert sind, ergibt sich dann auch der Fehler bezogen auf die Distanz der Bleche.

$$d_m - d_r = f(z_1) + f(z_2)$$

Hierbei ist $d_m - d_r$ die Differenz des gemessenen Abstand d_m und des realen Abstand d_r gleich der Summe $f(z_1) + f(z_2)$ der beiden gemessenen Tiefenwerte der jeweiligen Bleche. Über genügend viele Messungen kann die diskrete Funktion f bestimmt werden, die die Fehlerkurve approximiert und eine Korrektur

$$z_{corr} = z_m + f(z_m)$$

ermöglicht. Diese einfache Kalibrierung korrigiert den über den Tiefenwert nichtlinearen Messfehler annähernd, kann aber stark ausreißende Messfehler aufgrund der Diskretisierung nicht kompensieren.

Die intrinsische Kalibrierung einer ToF Kamera erscheint recht komplex, da zu der herkömmlichen Kalibrierung des 2D-Sensor auch die Bestimmung der Tiefe kalibriert werden muss, deren Messfehler nicht triviale, sehr unterschiedliche Ursachen haben kann und von Messsystem als auch Messobjekt abhängig sind. Ist die Kamera intrinsisch kalibriert, kann die extrinsische Kalibrierung über Marker analog der 2D-Kameras erfolgen. Da ToF-Kameras zur Messung der Tiefe in der Szene Lichtimpulse nutzen, erzeugen sie das zur Beleuchtung der Objekte notwendige Infrarot-Licht selbst, was den Vorteil hat, dass sie keine externe Beleuchtung benötigen.

7.4.4 Versuch der Kalibrierung auf Palette in Laserschneidmaschine (extrinsisch)

In den im Projekt gegebenen Laserschneidmaschinen lässt sich die Bedingung nutzen, dass alle Werkstücke bzw. Bleche auf einer ebenen Palette zur Bearbeitung abgelegt werden und diese somit in der gleichen Ebene liegen, um das Kalibrierverfahren für das geforderte Zusammenfügen der Kamerabilder zu einem Gesamtbild zu vereinfachen. Die Ebene der Palette ist in Maschinenkoordinaten bekannt. Zudem erscheint es interessant, dass die Laserschneid

maschine die zur Kalibrierung notwendigen Marker selbst zuschneiden kann. Dieses Verfahren wurde hinsichtlich Benutzbarkeit durch den Maschinenbediener sowie Zuverlässigkeit der erzeugten Kalibrierungen getestet.

Zur Kalibrierung sind, wie in Kapitel 7.2 erwähnt, eine ausreichende Menge Punktpaare, Objektpunkte zu Bildpunkten, zur Zuordnung von Maschinenkoordinaten zu Kamerakoordinaten notwendig. Liegen alle Objektpunkte in einer Ebene und ist die

intrinsische Kalibrierung gegeben, so genügen vier solche Punktpaare, um die perspektivische Projektion zwischen Palette und Bildsensor zu bestimmen.

Zur Durchführung der Kalibrierung wurden mehrere 2x2 Schachbrettmarker konstruiert und durch die Laserschneidmaschine zugeschnitten. Um sie optisch für den Benutzer besser unterscheidbar und sichtbar zu machen, wurden sie zusätzlich mit Löchern versehen und matt-weiß lackiert. Die fünf Marker wurden in das jeweilige Kamerabild gelegt. Die Koordinaten der Marker im Maschinenkoordinatensystem wurden mit Hilfe der Positionierung des Schneidkopfs bestimmt, an diesem war eine grüne Laserdiode angebracht, die auf das Blech einen definierten Punkt projiziert hat (Abbildung 7-17). Die Positionen der Marker in Bildkoordinaten wurden durch Anklicken der Markerzentren am Bildschirm bestimmt.



Abbildung 7-17: Marker in TRUMPF Werkzeugmaschine (Quelle: TRUMPF & HLRS)

Mit Hilfe der beiden perspektivischen Projektionen der Kameraebene in die Blechebene, konnten die beiden Kamerabilder in die Blechebene abgebildet und zusammengesetzt werden (Abbildung 7-18 links).

Um die Genauigkeit des Verfahrens und der Benutzereingaben zu untersuchen, wurden die Positionen der Marker im Kamerabild möglichst genau bestimmt. Jeweils eine der Eingaben wurde mit einem Fehler versehen, einer Varianz der Eingabe um jeweils ± 1 , ± 2 oder ± 3 Pixel. Diese Abweichungen sind Abschätzungen der Genauigkeit der Laserpunktbestimmung sowie der Findung und Markierung der Markerzentren durch den Benutzer. Markante Punkte im Zentrum des Kamerabilds wurden mit Hilfe der unterschiedlichen perspektivischen Transformationen in die Maschinenebene überführt und entsprechend eingefärbt (Abbildung 7-18 mitte, rechts). Deutlich zu erkennen ist die Verschmierung des Punktes durch die Transformation eines Punktes in die Ebene durch die fehlerbehaftete Perspektive. Die Stegweite, der Abstand zwischen zwei Lamellen, auf denen das Blech liegt, beträgt hier 10 cm, einige verschmierte Ausgangspunkte liegen deutlich in dieser Größenordnung.

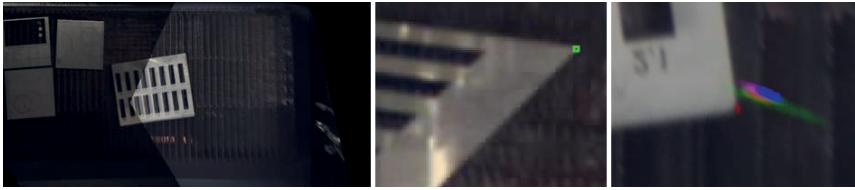


Abbildung 7-18: Stitching (links), Markierung im Kamerabild (m.) und in der Reprojektion (rechts) (Quelle: TRUMPF & HLRS)

Diese Untersuchung zeigt deutlich die Ungenauigkeiten auf, die durch fehlerhafte oder unpräzise Bestimmung der Markerposition bzw. Benutzereingabe resultieren. Die Abweichungen selbst geringer Fehler in der Markerposition gehen deutlich über die maximal tolerierte Abweichung hinaus. Fehler seitens des Nutzers bei Angabe der Markerpositionen mit Hilfe der Maus im Kamerabild lassen sich durch diverse Methoden noch verbessern, die präzise Positionierung des Laserkopfs auf den Markern unter Verwendung der Kamerabilder erscheint fraglich. Auf Nutzereingaben, die eine präzise Angabe bzw. Wahl der Markierung erfordern, soll aus diesen Gründen im Weiteren verzichtet werden.

7.4.5 Versuch der Kalibrierung in der SW Maschine (extrinsisch)

Für die Darstellung der erweiterten Realität, der Überlagerung von Aufnahmen einer realen Szene mit am Rechner generierten Bildern, werden häufig Marker benutzt, um Gegenstände oder die Szene selbst relativ zur Kamera zu verorten. Das Muster wird üblicherweise entsprechend der Anwendung ausgesucht. Typische Marker sind schwarz Quadrate auf weißem Hintergrund, die mit Hilfe eines Binärmusters innerhalb des Quadrats eine Identifikationsnummer kodieren. Eine typische Anwendung am HLRS ist die Überlagerung von realen Modellgeometrien mit Ergebnissen beispielsweise aus der Strömungssimulation (Abbildung 7-19 links). In diesem Beispiel wurde die Überlagerung zu Testzwecken mit Hilfe des transparenten Displays realisiert, das später in den Demonstrator verbaut wurde. Die Transparenz der transparenten OLED Technologie ist im Gegensatz zu den transparenten LCD Displays so gut, dass die Markererkennung selbst durch das transparente Display hindurch zuverlässig funktioniert.

Mit der Verortung des Markers in der Szene bei fix montierter Kamera, kann dieses Verfahren für die extrinsische Kalibrierung genutzt werden. Der Marker muss hierfür groß genug bzw. nahe genug an der Kamera und für diese vollständig sichtbar sein, um eine entsprechende Genauigkeit gewährleisten zu können. Die Eltec HiPerCam I Kamera, die in der SW Maschine verbaut ist, löst mit einer Auflösung von 5MP auf, was eine gute Erkennung des Markers ermöglicht (vgl. Kapitel 2.5). Die Prozedur zur extrinsischen Kalibrierung sieht vor, dass der Marker auf ein ebenes Werkstück aufgebracht wird und eingespannt wird. Mit Hilfe eines Messkopfes kann die Ebene, in der der Marker aufgebracht ist, sowie seine Position präzise vermessen werden (Abbildung 7-19 rechts).

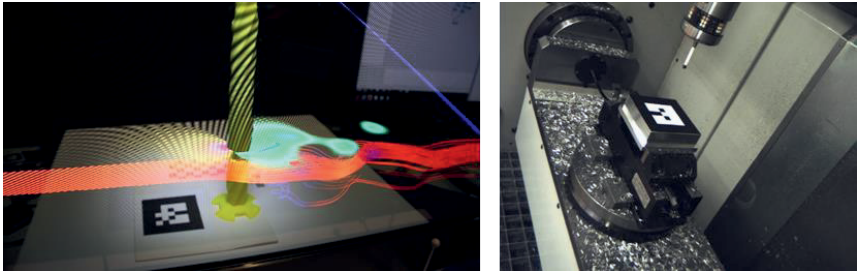


Abbildung 7-19: Markererkennung durch transparentes OLED Display (links), Markerplatzierung in SW Maschine (rechts) (Quelle: HLRS & Bosch Rexroth AG)

Nach der Bildaufnahme mit Hilfe der OpenCV Library wird die AR Marker Tracking Funktionalität der Library für Virtuelle und Erweiterte Realität ALVAR genutzt [7], um den Marker im Bauraum der SW Maschine zu detektieren und zu verorten. Eine Alternative zu ALVAR stellt die inzwischen in den Entwicklungsbaum von OpenCV integrierte [8] Library ArUco dar [9,10]. Aruco ist eine AR-Detection and Tracking Software. Sie verwendet ähnliche Marker wie ALVAR und erreicht gute Ergebnisse in der Markererkennung und Verortung.

7.5 Kalibrierung eines 1-Kamerasystems

Das manuelle Vorgehen zur Kamerakalibrierung mittels ausgedrucktem Schachbrettmuster (vgl. Kapitel 7.4.1 und 7.4.2) ist im Werkzeugmaschineninnenraum weder praktisch noch einfach durchzuführen. Die manuelle Aufnahme mehrerer Bilder des Schachbretts in verschiedenen Orientierungen und Positionen ist sehr aufwendig. Auch ist es je nach Gestaltung der Maschine oft nicht möglich, jede relevante Stelle des Innenraums zu erreichen, um dort eine Kalibrierung durchzuführen. Des Weiteren liefert das klassische Kalibrierverfahren die resultierenden extrinsischen Kameraparameter relativ zu einem der Schachbrettkoordinatensysteme. Einen Bezug zwischen diesem Koordinatensystem und dem Maschinenkoordinatensystem herzustellen ist äußerst schwierig und nur mit begrenzter Genauigkeit möglich. Ein solcher Bezug ist jedoch nötig, um eine Überlagerung des von der Kamera erfassten Maschineninnenraums mit Mehrwertinformationen, deren Positionen relativ zum Maschinenkoordinatensystem angegeben werden, zu ermöglichen (vgl. Kapitel 6.4.1).

Um die genannten Nachteile des klassische Kalibrierverfahrens zu überwinden und es tauglich für den Einsatz im Innenraum von Werkzeugmaschinen zu machen, wurde das Vorgehen auf ein automatisiertes Verfahren zur Nutzung in Werkzeugmaschinen übertragen. Anstelle von ausgedruckten Schachbrettmustern wird zur Erzeugung der für die mathematische Berechnung der Kameraparameter notwendigen Kalibrierpunkte nun ein spezielles Werkzeug in die Maschine eingespannt. Dieses Werkzeug, dargestellt in Abbildung 7-20 links, besteht

aus einem für den automatischen Werkzeugwechsel geeigneten Steilkegel und einer leuchtenden LED-Werkzeugspitze.

Diese Werkzeugspitze kann sehr zuverlässig mittels Bildverarbeitung in Kamerabildern detektiert werden. Mit diesem Werkzeug kann die Maschine, wie in Abbildung 7-20 rechts dargestellt, beliebig viele Kalibrierpunkte in Form mehrerer Schachbrettmuster anfahren. Auch andere Muster und Anordnungen der Kalibrierpunkte sind möglich. Die Positionen der Kalibrierpunkte werden in Maschinenkoordinaten beschrieben, was dazu führt, dass auch die resultierenden extrinsischen Kameraparameter in diesem Koordinatensystem berechnet werden. Die dynamische Bestimmung der Kalibrierpunkte sowie die Kommandierung der Werkzeugmaschine erfolgt durch einen maschinenunabhängigen Kalibrierdienst. Dieser Dienst kann in das Framework für Mehrwertdienste (vgl. Kapitel 4) eingebunden werden. Die automatisierte Ansteuerung der Maschine erfolgt mittels der Steuerungsanbindung über OPC UA (vgl. Kapitel 5) und dient sowohl zum Ein- und Auswechseln des Kalibrierwerkzeugs wie auch zum Anfahren der Kalibrierpunkte. Nach Erreichen der angeforderten Sollposition wird die Bildaufnahme automatisch ausgelöst und die Bildposition der Werkzeugspitze ermittelt. Aus der tatsächlichen 3D-Position der Werkzeugspitze im Maschineninnenraum und der 2D-Position im Kamerabild wird ein Punktepaar (P, p) gebildet. Die aus mehreren Aufnahmen gewonnenen Punktepaare werden abschließend an den Kalibrieralgorithmus (vgl. Kapitel 7.4.1) übergeben und somit alle Kameraparameter ermittelt. Auf diese Weise kann der Maschinenbediener die Kalibrierung per Knopfdruck durchführen und muss keine weiteren manuellen Handgriffe tätigen. Auch mehrere Kameras innerhalb des Maschineninnenraums lassen sich mit diesem Verfahren gleichzeitig kalibrieren.

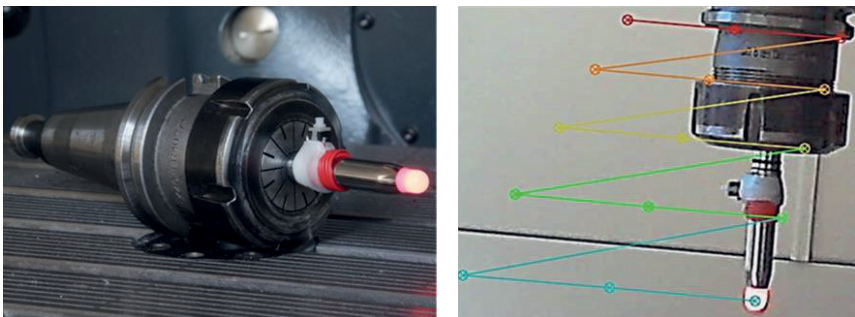


Abbildung 7-20: Werkzeug zur automatisierten Kalibrierung (links); Automatisch angefahrene Kalibrierpunkte eines Schachbrettmusters (rechts) (Quelle: ISW)

7.6 Kalibrierung eines Multi-Kamerasystems

Die bisherigen Ansätze für die Kamerakalibrierung hatten zum Ziel eine Menge Punktepaare (P, p) für jeweils eine Kamera j zu bestimmen. Diese Menge Punktepaare wurde explizit für eine

Kamera gesucht (Kapitel 7.4.1), sie kann aber auch gleichzeitig für mehrere Kameras in einem Kamerasystem bestimmt werden (Kapitel 7.5). Das gleichzeitige Bestimmen dieser Punktpaare nutzt die Sichtbarkeit der LED beispielsweise der Werkzeugspitze (Abbildung 7-20) in den verschiedenen Kameraperspektiven, sind die Punktpaarmengen gefunden, wird jede Kamera einzeln kalibriert.

Um dieses Verfahren weiter zu verbessern, kann in Analogie zur Kalibrierung von Stereokameras die Eigenschaft genutzt werden, dass der Punkt P , der in allen Kameras $j=1,\dots,n$, die diesen Punkt sehen und auf den Punkt p_j in den Kameraraum abbilden, derselbe Punkt ist. Auf diese Weise können alle Kameraprojektionen miteinander in Bezug gebracht werden, so dass ein Gesamtgleichungssystem für das Kamerasystem aufgestellt werden kann. Diesen Ansatz nutzt man, wie angesprochen, beispielsweise für die Kalibrierung von Stereokameras mit $n=2$. Svoboda [11] erweitert diesen Ansatz auf größere n in einem Multikamerakalibrieralgorithmus [12], der in der Lage ist, aufgrund der für alle Kameras zur meisten Zeit sichtbare Punkte P_j intrinsischen und extrinsischen Parameter zu bestimmen (Abbildung 7-21).

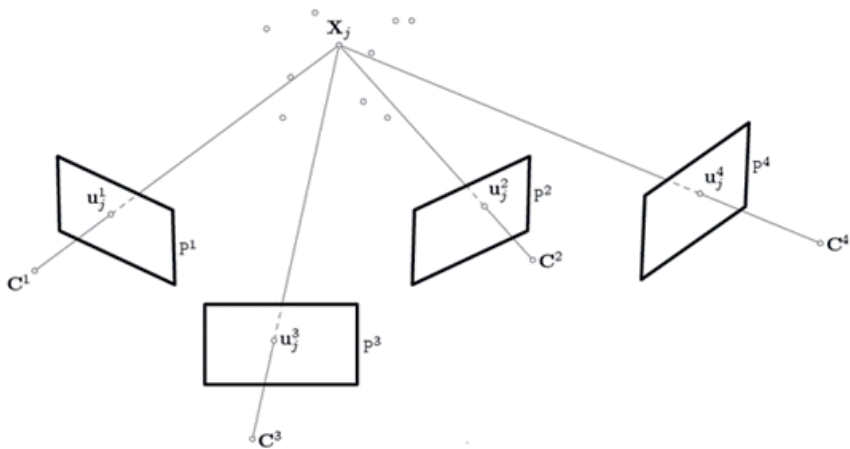


Abbildung 7-21: Schematische Darstellung eines Multi-Kamerasystems mit Ansatz nach Svoboda (Quelle: [11])

In Svobodas Ansatz [11] wird der Punkt X_j von den Kameras C^1 bis C^4 auf den Punkt u_j^i in den jeweiligen Bildraum p^1 bis p^4 der Kameras projiziert. Svoboda nutzt für die Aufnahme der Punkte X_j einen Laserpointer bzw. ein ähnlich leuchtendes Utensil, das im Kamerabild eine punktförmige Form erscheinen lässt. Für die Umsetzung dieses Verfahrens am Prototyp sind die folgenden Kalibrierutensilien aufgebaut worden (Abbildung 7-22).

Der Kalibrierwinkel dient zur Bestimmung der Skalierung s in (7.2). Dieser kommt nur am Prototyp zum Einsatz, da dies hier die einzige Möglichkeit ist, eine für alle Kameras einheitliche Skalierung vorzugeben. In den Werkzeugmaschinen kann beispielsweise der Werkzeugkopf mit LED (Abbildung 7-20 rechts) genutzt werden, um entsprechende Punkte mit definierten Abstand anzufahren. Die Skalierung kann so direkt aus den Maschinenkoordinaten übernommen werden. Der Kalibrierpointer ist eine SMD LED, die durch eine Batterie versorgt wird und handgeführt im Arbeitsraum bewegt werden kann. Die LED, welche in ein transparentes SMD Gehäuse verbaut ist, ermöglicht die Aufnahme einer nahezu punktförmigen Lichtquelle aus fast allen Richtungen. Der Kalibrierpointer nutzt einen NodeMCU [13] mit ESP8266, um die Bildaufnahme zu synchronisieren. Sobald der Knopf am Kalibrierpointer gedrückt wird, sendet der NodeMCU ein UDP Packet an das Kamerasystem, wodurch alle Kameras die Bildaufnahme auslösen. Dieses Verfahren ist für die Entwicklung am Prototypen notwendig, die spätere Aufnahme der LED in der Werkzeugmaschine erfolgt, wie auch mit dem Kalibrierwinkel, mit einem LED-Werkzeugkopf. Die Synchronisierung ist notwendig, um alle 18 Kameras des Prototypen bzw. alle 8 Kameras des Demonstrators gleichzeitig auslösen zu können.



Abbildung 7-22: Kalibrierwinkel (links), Kalibrierpointer (Mitte), LED des Kalibrierpointers (rechts) (Quelle: HLRS)

7.6.1 Vorgehen

Das Vorgehen zur Kalibrierung eines Multikamerasystems gliedert sich im Wesentlichen in die folgenden drei Schritte.

1. Finden der LED Projektionen u_j^t in allen Kamerabildern

Dieser Schritt erfolgt interaktiv mit dem Anwender bzw. mit der Maschine. Am Prototyp werden die Kameras entsprechend bei Knopfdruck am Kalibrierpointer ausgelöst. An der Werkzeugmaschine erfolgt die Aufnahme des LED-Werkzeugkopfs nach Anfahrt einer Position im Arbeitsraum der Werkzeugmaschinen. Aus den Bildsätzen für alle LED Positionen wird die Position der LED im Kamerabild detektiert (siehe Kapitel 7.6.2).

2. Finden der LEDs auf dem Kalibrierwinkel

Zur Berechnung der Skalierung s wird ein absolutes Maß im Arbeitsraum benötigt. Dies erfolgt am Prototyp durch den Kalibrierwinkel oder durch den LED Werkzeugkopf im Arbeitsraum der Werkzeugmaschine. Sind die LED Punkte im Arbeitsraum aus Schritt 1 bereits bekannt, so kann dieser Schritt auch übersprungen werden.

3. Durchführung der MultiCamCalib nach Svoboda

Mit Hilfe der Koordinaten der LED in den jeweilig zugehörigen Bildsätzen wird der MultiCamCalib Algorithmus nach Svoboda durchgeführt (siehe Kapitel 7.6.3). Dieser Schritt erfolgt unabhängig von Maschine und Nutzer.

Der Kalibriervorgang nimmt einige Zeit insbesondere zur Aufnahme der LED-Punkte in Anspruch. Je nach Anzahl der Kameras und Punktmengen benötigt die Anfahrt der Positionen, die Synchronisation mit der Maschine bzw. dem Kalibrierpointer, die Aufnahme der LEDs sowie die Bestimmung der LED Punkte in den Bildern die meiste Zeit. Bereits mit 40 LED-Positionen konnten erfolgreiche Kalibrierungen bestimmt werden. Je nach Verdeckung und Bauraumgröße werden aber 100 oder mehr LED-Positionen empfohlen, da mit zunehmender Kameraanzahl die Wahrscheinlichkeit steigt, dass eine Kamera die LED im Bild nicht finden kann.

7.6.2 Finden der LED Projektionen

Die Detektion und Bestimmung von Laserpunkten oder Laserstrichen in Kamerabildern ist ein bekanntes Problem, viele Anwendungen nutzen sehr unterschiedliche Ansätze. Svoboda empfiehlt eine Verdunklung des Raumes [11], dies erleichtert die Detektion von Licht in Kamerabildern wesentlich, eine Verdunklung kann aber nicht immer gewährleistet werden. Ziel im Projekt war, eine Bildverarbeitung zur Detektion und Bestimmung von LEDs zu implementieren, die zwar davon ausgehen kann, dass das Licht im Arbeitsraum der Werkzeugmaschine abgeschaltet werden kann, aber üblicherweise Umgebungslicht oder eine Hallenbeleuchtung immer vorhanden ist.

Das Vorgehen zur Detektion und Bestimmung der Positionen der LEDs in den Kamerabildern gliedert sich in die folgenden sechs Schritte (Abbildung 7-23).

1. `blur()`

Der erste Schritt rechnet das Ausgangsbild leicht unscharf. Das vereinfacht die spätere Konturdetektion und macht das Verfahren gleichzeitig ein wenig robuster bezüglich schlechter Fokussierung der Kamera.

2. `thresholdRGB()`

In Abhängigkeit der gewählten LED-Farbe, wird ein dynamisches Schwellwertverfahren genutzt, um die LED-Farbe sowie die durch die LED potentiell saturierten Bildbereiche hervorzuheben. Bereits in leicht gedimmter Umgebung sind LEDs oder Laserdioden bereits so hell, dass der Bildsensor den Lichtpunkt nicht mehr in Farbe auflösen kann und er weiß erscheint.

3. `findContours()`

Dieser Schritt sucht zusammenhängende Farbbereiche im Bild und liefert eine Menge von erkannten Konturen im Bild.

4. `filterContours`

Die Konturen werden hinsichtlich Größe und Form untersucht. Nur solche Konturen sind von Interesse, die Ähnlichkeit mit einer Kreisform oder der Form einer Ellipse haben und deren Größe der LED bzw. des Lichtkleks dem erwarteten Abstand der LED im Bauraum entsprechen. Die Größe des Bauraums in der Werkzeugmaschine sowie eine Positionierung der Kameras in gewissen Grenzen zu diesem werden als bekannt vorausgesetzt.

5. `analyseContourShape()`

Nachdem bereits die Konturen herausgefiltert worden sind, die sehr wahrscheinlich nicht die LED-Projektion sind, werden die übrigen Konturen in diesem Schritt analysiert. Beispielsweise sind solche Konturen sehr wahrscheinlich eine LED bzw. ein Laserpointer, die ein weißes Zentrum mit farbigem Halo besitzen (Abbildung 7-23 rechts unten).

6. `calcCentre()`

Da auch eine LED oder eine Laserdiode beispielsweise aufgrund ihrer Bauart und der diskreter Bildauflösung nicht punktförmig im Bild erscheint, erfolgt im letzten Schritt die Berechnung der Zentren der Konturen.

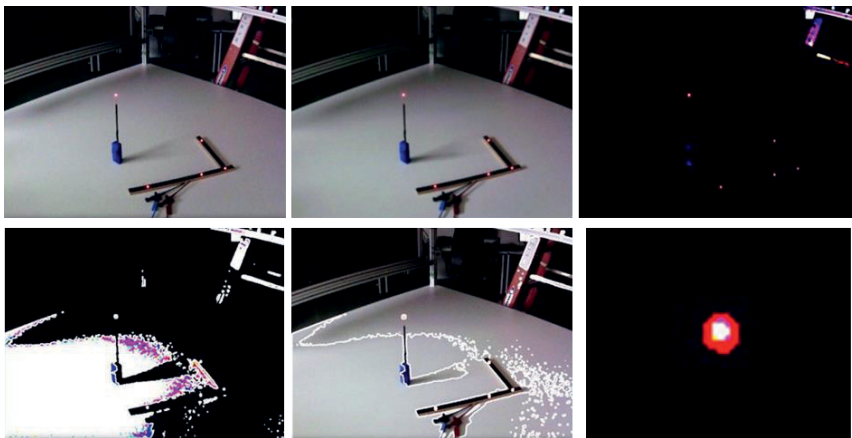


Abbildung 7-23: Ausgangsbild und Schritte 1 bis 5 zur Detektion der LED-Lichtquelle (v.l.o.n.r.u.) (Quelle: HLRS)

Ergebnis des Verfahrens ist eine Liste der Zentren der im Bild erkannten Konturen. Das Verfahren funktioniert bei leicht gedimmten Umgebungen zuverlässig (Abbildung 7-24). Direktes Sonnenlicht kann im Prinzip durch die Konturfilterung toleriert werden, macht aber die LED, wenn sich diese im Sonnenlicht befindet, für die Kamera meist aufgrund mangelnden Kontrasts nicht sichtbar. Ein zusätzliches Problem stellen Reflexionen dar. Schwache oder verzerrte Reflexionen werden herausgefiltert, spiegelnde Reflexionen werden jedoch als zusätzlicher Marker erkannt.

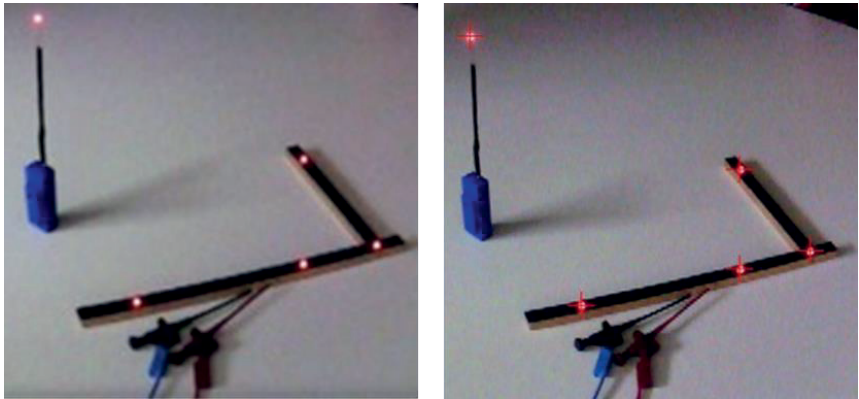


Abbildung 7-24: Ergebnis der LED Detektion und Bestimmung überlagert mit Ausgangsbild (Quelle: HLRS)

Das Verfahren ließe sich deutlich vereinfachen, wenn man LED und Kameras auf bestimmte Wellenlängen des Lichts einschränken könnte. Die Kameras sollen aber ohne Modifikation bleiben und gleichzeitig auch Normalbilder für den Anwender bzw. die 3D-Rekonstruktion liefern können.

7.6.3 Der MultiCamCalib Algorithmus

Sind in hinreichend vielen Bildsätzen, das heißt den Aufnahmen desselben LED-Punkts, die Projektionen der LED im Bild erkannt worden, kann auf diese Daten der MultiCamCalib Algorithmus angewandt werden. Der Algorithmus gliedert sich in die vier wesentlichen Schritte wie folgt.

1. Finden und eliminieren fehlerhafter Punktpaare mit RANSAC-Analyse
2. Abschätzen der Projection Depth
3. Aufstellen und Lösen des Gleichungssystems
4. Berechnung der intrinsischen und extrinsischen Parameter aller Kameras

In Zusammenhang mit der LED-Punkterkennung ist ein wesentlicher Schritt, fehlerhafte Punkte bzw. in den jeweiligen Bildpaaren nicht zusammenpassende Punkte zu finden und diese Punktpaare dann zu löschen. Schlecht zusammenpassende Punkte können dazu führen, dass das Gleichungssystem in Schritt 3 nicht lösbar wird. Solche Punkte können zum Beispiel dann entstehen, wenn die Kameras nicht synchron auslösen und sich die LED in unterschiedlichen Bildern eines Bildsatzes bereits weiterbewegt hat.

In Schritt 3 werden alle Informationen, die über das Kamerasystem bekannt sind, genutzt, um die Freiheitsgerade des Systems vorab möglichst zu reduzieren. Sind keinerlei Informationen hinsichtlich intrinsischer und extrinsischer Kalibrierung gegeben, so sind nach Svoboda 8

Kameras nötig, um das Gleichungssystem lösen zu können [14]. Sind die intrinsischen Parameter bekannt, so genügen lediglich 3 Kameras. Sind die Punkte in P in Weltkoordinaten bekannt, was in der Werkzeugmaschine gegeben ist, lässt sich das Verfahren noch weiter vereinfachen, so dass das Verfahren sogar mit nur einer Kamera arbeiten könnte. Dieser Fall geht dann in das Verfahren zur Kalibrierung eines 1-Kamerasystems (Kapitel 7.5) über.

Bei erfolgreicher Lösung des Gleichungssystems gibt der Algorithmus die intrinsischen und extrinsischen Parametersätze aus, die von OpenCV übernommen und für die weitere Bildverarbeitung genutzt werden können.

7.7 Literaturverzeichnis

- [1] Hardkernel oCam-5CRO-U
http://www.hardkernel.com/main/products/prdt_info.php?g_code=G145231889365
 (01. Feb. 2018)
- [2] Eltec HiPerCam I
<https://www.eltec.de/de/produkte/industrial-imaging/intelligente-kameras/hipercam-i.php>
 (01. Feb. 2018)
- [3] OpenCV 2.4.13.5 Documentation – Camera Calibration and 3D Reconstruction
https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html
 (01. Feb. 2018)
- [4] Peter Fürsattel et al. (2016). *A Comparative Error Analysis of Current Time-of-Flight Sensors*. In:
 IEEE Transactions on Computational Imaging, Vol. 2, No. 1, March 2016.
- [5] Rasmus Larsen, Erhardt Barth & Andreas Kolb Computer (2010) *Special issue on Time-of-Flight Camera Based Computer Vision*. In: Vision and Image Understanding 114, ff.1318-1328 Dezember 2010.
- [6] OpenCV 3.0.0-dev Documentation – Camera Calibration
https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_calib3d/py_calibration/py_calibration.html
 (01. Feb. 2018)
- [7] ALVAR – A library for virtual and augmented reality
<http://alvar.erve.vtt.fi/>
 (01. Feb. 2018)
- [8] OpenCV contrib – ArUco Marker Detection (github)
https://github.com/opencv/opencv_contrib/tree/master/modules/aruco
 (01. Feb. 2018)

-
- [9] S. Garrido-Jurado et al. (2014). *Automatic generation and detection of highly reliable fiducial markers under occlusion*. In: Pattern Recognition 47, 6, pp. 2280-2292, June 2014.
- [10] OpenCV 3.1.0-dev Documentation –Detection of ArUco Markers
https://docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detection.html
(01. Feb. 2018)
- [11] Multi-Camera Self-Calibration
<http://cmp.felk.cvut.cz/~svoboda/SelfCal/>
(01. Feb. 2018)
- [12] MultiCamSelfCal (github)
<https://github.com/strawlab/MultiCamSelfCal/>
(01. Feb. 2018)
- [13] NodeMCU
<http://www.nodemcu.com>
(01. Feb. 2018)
- [14] Svoboda, T., Martinec & D., Pajdly, T. (2005). *A Convenient Multi-Camera Self-Calibration for Virtual Environments*. In PRESENCE:Teleoperators and Virtual Environments, 14(4), August 2005.

8 Echtzeit-Maschinensimulation und der Nutzen von Mehrwertdiensten

M. Gras, D. Schaefer, E. Berckmann, M. Stautner

8.1 Ziele

Die Echtzeit-Maschinensimulation und die Verwendung von Mehrwertdiensten sind die Grundlage für den Einsatz eines intelligenten Maschinenfensters an einer Werkzeugmaschine. Der Aufgabenbereich „Echtzeit-Maschinensimulation und Mehrwertdienste“ verfolgte daher das Ziel eine flüssige Echtzeit-Maschinensimulation zu gewährleisten und für den Benutzer arbeitserleichternde Mehrwertdienste bereitzustellen.

Die Maschinensimulation wurde im Projekt auf einem vor dem Maschinenfenster installierten transparenten Display (OLED) und an einem seitlich der Werkzeugmaschine angebrachten nicht-transparenten Display gezeigt. Der Benutzer, z.B. der Maschinenbediener, hat so die Möglichkeit den Fertigungsprozess trotz Störfaktoren wie Spänen und Kühlwasser, die die Sicht während der Fertigung behindern, vollständig zu überwachen. Dazu wird eine Augmented Reality (AR) Umgebung (am OLED) und eine Virtual Reality (VR) Umgebung (am nicht-transparenten Display) geschaffen, die zudem eine Interaktion mit dem Benutzer ermöglichen. Die Umsetzung erfolgte an einer Demonstratormaschine des Typs *DMG DMU 50 ecoMill*, unter Verwendung der Trackingsysteme *Microsoft Kinect* und *ART SMARTTRACK* beim Projektpartner ISG (Abbildung 8-1, Abbildung 8-2).



Abbildung 8-1: Nicht-transparentes Display seitlich der Werkzeugmaschine mit Möglichkeit des Einsatzes virtueller Realität durch Verwendung einer Shutter-Brille (Quelle: ModuleWorks)

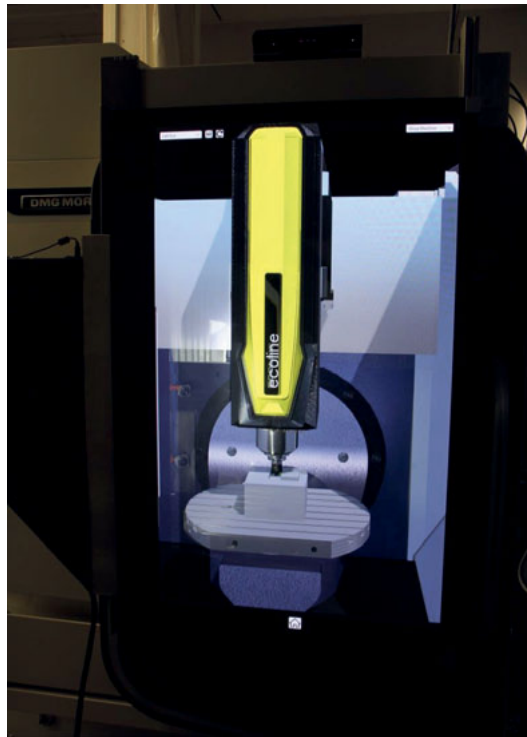


Abbildung 8-2: Transparentes (OLED) Display am Maschinenfenster mit Echtzeit-Maschinensimulation und Möglichkeit des Einsatzes von Augmented Reality durch Microsoft Kinect Tracking (Quelle: ISW)

8.2 Echtzeit-Machinensimulation

Die Umsetzung der Echtzeit-Maschinensimulation erfolgt über die Einbindung der Softwarebibliothek *ModuleWorks CNCsim*. Diese Bibliothek ermöglicht die Simulation von CNC-Fertigungsprozessen. Die dazu benötigten Eingabedaten werden vom Benutzer über die Software-Schnittstelle API bereitgestellt:

1. Daten der Werkzeugmaschine (XML / STL / API)
2. Daten der verwendeten Werkzeuge (STL / API)
3. Maschinenkommandos, z.B. für die Positionierung des Werkzeugs (NC-Code / API)

Die Modellierung der Maschine erfolgt innerhalb der Simulationsumgebung über die Beschreibung ihrer Komponenten sowie der zu Grunde liegenden Kinematik. Die Komponenten werden unter anderem in Rotations- bzw. Translationsachsen, Werkzeuge, zu

bearbeitende Werkstücke sowie andere nicht veränderliche Bauteile (z.B. das Maschinengehäuse) eingeordnet (Abbildung 8-3).

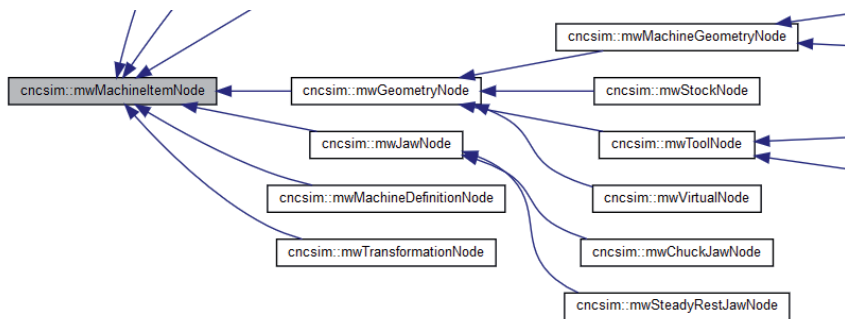


Abbildung 8-3: Ausschnitt aus der Klassenstruktur der Maschinenkomponenten in CNC_Sim (Quelle: ModuleWorks)

Die Kinematik wird als Baumstruktur modelliert und beschreibt insbesondere wie die Position der einzelnen Komponenten von den Maschinenachsen abhängen (siehe Abbildung 8-4).

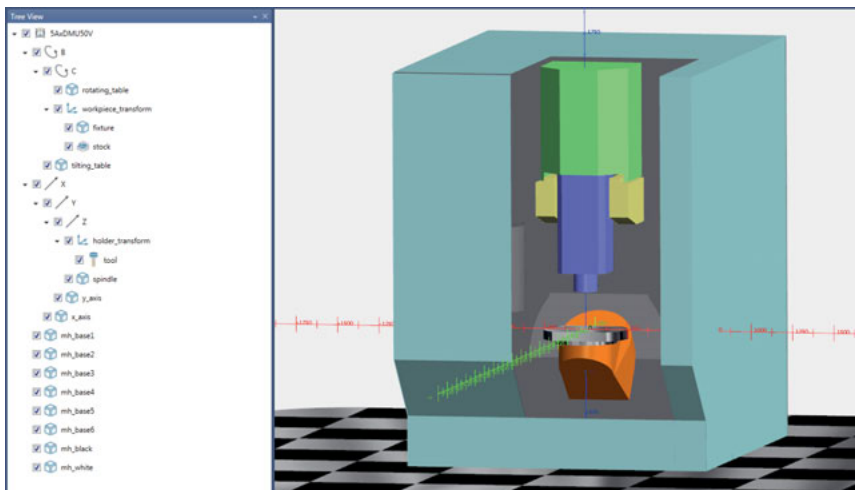


Abbildung 8-4: Baumstruktur einer Maschinenkinematik und Maschinensimulation (Quelle: ModuleWorks)

CNC_Sim bietet die Möglichkeit den kinematischen Baum über die API aufzubauen. Alternativ dazu, können auch die Kinematik des Maschinenmodells via XML- sowie die Geometrien als STL-Dateien im-/exportiert werden. Im Folgenden ist ein Code-Snippet zum Aufsetzen einer Werkzeugmaschine beschrieben.

```

// Get the machines root node:
cncsim::mwMachineDefinitionNode machineRoot = simulationEngine->GetMachine();

// Check if the machine (node) was attached to the simulator
if (!machineRoot.IsValidHandle())
{
    throw misc::mwException(0, "No machine definition set!");
}

// Create a simple machine part:
cncsim::mwMachineGeometryNode machineCoreCylinder = simulationEngine->
GetNodeFactory().CreateMachineGeometryNode();
machineCoreCylinder.SetName(misc::mwstring(_T("CoreCylinder")));

// Attach the new machine part to the machine (add it right below the root node)
simulationEngine->SetParentNode(machineCoreCylinder.GetId(), machineRoot.GetId());

// Create a cylindrical mesh for this machine part (by revolving a parallel line
round the abscissa)
cadcam::mw2dPolyLine<float> coreCylinderContour;
coreCylinderContour.AddPoint(-10, 10);
coreCylinderContour.AddPoint(+10, 10);
cadcam::mwfMeshPtr coreCylinderMesh =
    cadcam::mwMeshFactory<float>::CreateRevolved(
        measures::mwUnitsFactory::METRIC, // metric units
        coreCylinderContour, // profile to revolve
        0.f, 360.f, 5.f, // rotation section, stepping (angles in degrees)
        0.f, 0.f, 0.f, // axes offsets
        false, // inverse normals
        cadcam::mwMeshFactory<float>::HAS_TOP_BOTH, // mesh closure
        cadcam::mwMeasurableAxis<float>::AXIS_X); // revolving axis

// Add the parts mesh to the machine node
machineCoreCylinder.SetMeshes(cncsim::mwMachineGeometryNode::MeshSet(1,
coreCylinderMesh));

```

Die verwendeten Werkzeuge werden entweder als STL eingelesen oder können durch entsprechende API-Calls durch die CNCsIm Bibliothek festgelegt werden.

Die Maschinenkommandos werden der API als Eingabe zur Verfügung gestellt und können so zur Simulation des Fertigungsprozesses und zur Kollisionsüberwachung genutzt werden.

Innerhalb des Projektes *iWindow* wurde die ModuleWorks CNCsIm Bibliothek erweitert und in das Human-Machine-Interface (HMI) der CNC-Steuerungen der Projektpartner Bosch Rexroth und Trumpf eingebunden. Dazu müssen die CNC-Steuerungen die NC-Kommandos interpretieren und entsprechend die API von *CNCsIm* ansprechen. Dazu gehören neben dem Bewegen der Maschinenachsen auch andere Operation wie z. B. das Ein-/Abschalten der Werkzeugspindel. Aus der Abfolge aller Befehle ergibt sich dann direkt die Positionierung des Werkzeugs im Maschinenraum sowie der Werkzeugweg, welcher nun in 3D visualisiert werden kann. Darüber hinaus wurde die Abtragsimulation in die Echtzeit-Simulationssoftware ISG-virtuos des Projektpartners ISG integriert (siehe Kapitel 9.4).

8.3 Überlagerungsberechnung

Eine wichtige Aufgabe im Rahmen der Maschinensimulation für das Projekt *iWindow* ist die korrekte Berechnung der Überlagerung von realer Werkzeugmaschine und Maschinensimulation. Bei Verwendung eines transparenten Displays muss die Position der Werkzeugmaschine mit der auf dem transparenten Display visualisierten Maschinensimulation übereinstimmen bzw. bei Verwendung eines nicht-transparenten Displays muss der Live-Stream, der von den in der Werkzeugmaschine installierten Kameras aufgenommen wird und der in der Maschinensimulation eingebettet ist, mit der Maschinensimulation übereinstimmen.

Um dies zu gewährleisten ist eine Kalibrierung des Kamerasystems notwendig, die mit den freien Bibliotheken OpenCV [1] und ALVAR [2] realisiert wurde. Die Maschinenelemente können so in der dreidimensionalen Maschinenvisualisierung dargestellt werden. Die zur Überlagerungsberechnung verwendeten Verfahren sind in Kapitel 6 beschrieben. Abbildung 8-5 zeigt eine fehlerhafte Überlagerung der Maschinensimulation mit der realen Werkzeugmaschine am transparenten Display, bei der sich die Werkzeugmaschine weiter links befindet, die Simulation aber mittig dargestellt wird.

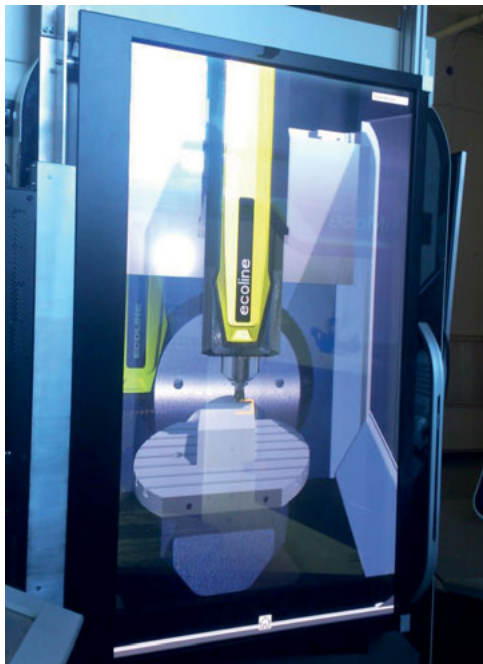


Abbildung 8-5: Echtzeit-Maschinensimulation am transparenten Display mit fehlerhafter Überlagerung der Werkzeugmaschine (Quelle: ModuleWorks)

8.4 Kollisionsvermeidung

CNCSim ermöglicht die Kollisionsvermeidung auf der CNC Steuerung. Dies ist ein wichtiges Feature, da Kollisionen zu enormen Schäden an der Werkzeugmaschine und am Werkstück führen können. Um die Kollisionsvermeidung nutzen zu können, müssen neben den aktuellen Positionsdaten der Maschine auch Informationen über zukünftige Maschinenkommandos des Fertigungsprozesses bei der Simulation berücksichtigt werden. Der Simulationskern berechnet anhand dieser Daten, ob die nun folgenden Bewegungen zu einer Kollision zwischen Maschinenkomponenten und/oder Nichteinhalten des Sicherheitsabstands führen. Wird eine kritische Bewegung erkannt stoppt die Maschine. Dabei ist es wichtig, dass ein Lookahead, d.h. ein Zeitintervall zwischen dem Erhalt der zukünftigen Maschinenkommandos und der Ausführung dieser Kommandos eingehalten wird. Dieser Lookahead muss größer sein als die längste zu erwartende Berechnungszeit für die Simulation der Kommandos und der längsten erwarteten Zeit zum Stoppen der Werkzeugmaschine. Abbildung 8-6 zeigt ein Beispiel einer Kollision zwischen Werkzeug und Maschinenkomponente. Die kollidierenden Komponenten sind rot eingefärbt.

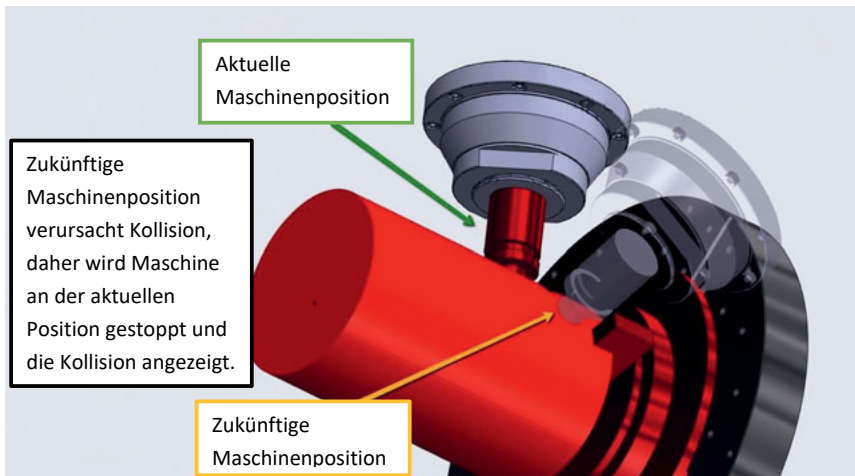


Abbildung 8-6: Beispiel einer Kollision zwischen Werkzeug und Maschinenkomponente (Quelle: ModuleWorks)

Bei der Kollisionserkennung kommen verschiedene Techniken zum Einsatz, darunter „State of the Art“ Verfahren und Eigenentwicklungen von ModuleWorks zur Berechnung der Überschneidung von geometrischen Primitiven, Dreiecksnetzen und Voxellmodellen.

8.5 Mehrwertdienste

Ein wesentlicher Fokus des Projektes *iWindow* stellen die Mehrwertdienste dar. Diese stellen durch Einblenden auf dem Display computergenerierte Informationen bereit und bilden dadurch eine Erweiterung der Informationsebene Maschineninnenraum (vgl. Kapitel 4). Dabei steht die Interaktion mit dem Benutzer im Vordergrund. Mehrwertdienste die schon in diesem Kapitel beschrieben wurden sind die Abbildung der Maschinensimulation innerhalb der AR- bzw. der VR-Umgebung. Zusätzlich wurden die Mehrwertdienste „Visualisierung der Fertigungsbahn“ und „Visualisierung der Werkstück- bzw. Werkzeugkontur“ innerhalb des Projektes *iWindow* entwickelt. Mit Hilfe dieser kann die geplante Bearbeitung verifiziert und die Inbetriebnahme der Werkzeugmaschine vereinfacht werden.

8.5.1 Bahnvisualisierung inklusive Überdeckungsrechnung

Der Mehrwert „Bahnvisualisierung“ visualisiert die Fertigungsbahn und stellt dadurch für den Benutzer eine enorme Unterstützung dar. Der Benutzer kann sowohl die schon abgefahrte Fertigungsbahn, als auch die in der Zukunft liegenden Fertigungsbahnen sehen und wenn nötig in den Fertigungsprozess eingreifen. Abbildung 8-7 zeigt die zurückgelegten Fertigungsbahnen an einem einfachen Werkstück.

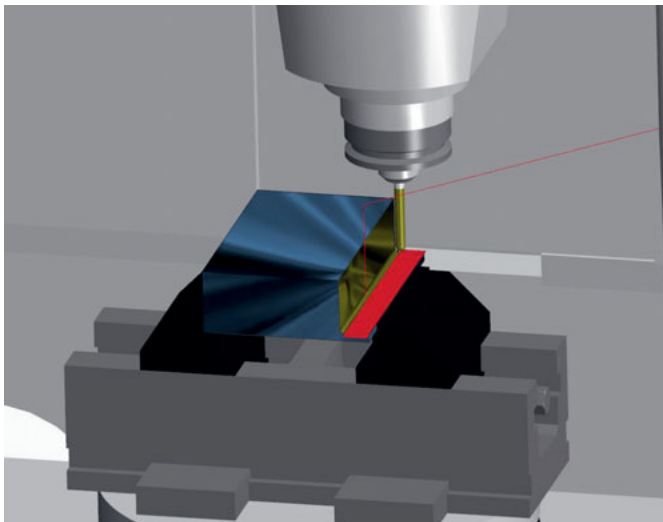


Abbildung 8-7: Visualisierung der Fertigungsbahn (Quelle: ModuleWorks)

8.5.2 Werkstück- und Werkzeugkonturvisualisierung

Die „Werkstück- und Werkzeugkonturvisualisierung“ stellt einen Mehrwert für den Nutzer dar, da er für das Einspannen des Werkstücks in die Aufspannung genutzt werden kann. Bei diesem Mehrwert kann die optimale Position des Werkstücks innerhalb der Aufspannung

durch die Simulation visuell dargestellt werden. Der Benutzer kann das Werkstück dann innerhalb des Maschinenraums bewegen und in der Aufspannung befestigen und bekommt „live“ ein Feedback ob die von ihm gewählte Position der Zielposition entspricht. Dies reduziert die für Inbetriebnahme benötigte Zeit. Abbildung 8-8 zeigt die Konturvisualisierung an einem einfachen Werkstück und der Aufspannung.

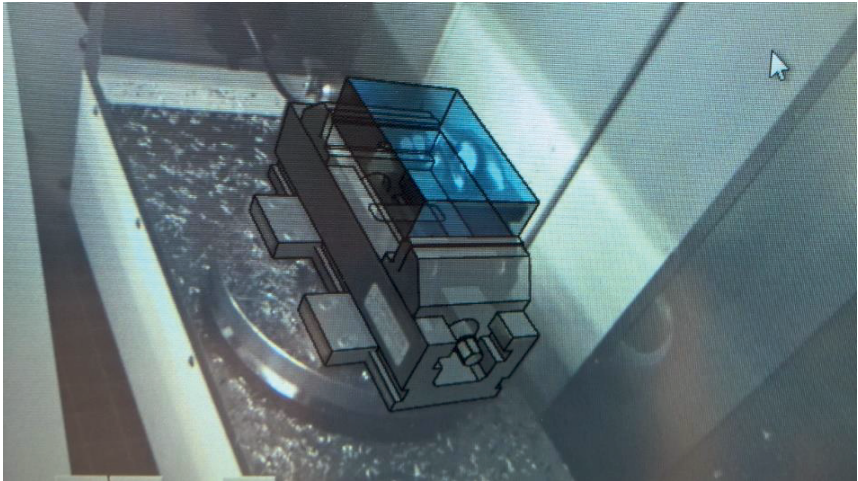


Abbildung 8-8: Visualisierung von Kanten der Aufspannung und des Werkstücks
(Quelle: ModuleWorks, Bosch Rexroth)

8.5.3 Perspektivische Visualisierung

Für die Verwendung eines AR-Systems am transparenten Display des Maschinenfensters und eines VR-Systems am nicht-transparenten Display seitlich an der Werkzeugmaschine ist die perspektivische Visualisierung notwendig. Die Bewegung des Benutzers wird durch zwei Trackingsysteme verfolgt.

Beim transparenten Display kommt eine Microsoft Kinect Kamera zum Einsatz. Beim nicht-transparenten System kommt eine Shutter-Brille zum Einsatz, die die Position des Betrachters in den Kontext der VR-Umgebung bringt.

Bei Bewegung des Benutzers werden die Positionsdaten an die Software weitergegeben und so der Blickwinkel der 3D-Visualisierung entsprechende Transformationsmatrizen angepasst (siehe Kapitel 6.5).

8.6 Ausblick

Um den Benutzer weiter zu unterstützen ist die Visualisierung weiterer kontextbezogener Prozessparameter denkbar. Diese könnten beispielsweise die noch verbleibende

Fertigungslaufzeit, den Status der Produktionsaufträge der jeweiligen Maschine oder die Stückzahl einblenden, aber auch Prozessparameter der Fertigung. Abbildung 8-9 zeigt eine Turbinenschaufel mit eingeblendeten Schnitttiefen und Histogrammen für Schnitttiefe und Bahnabstand.

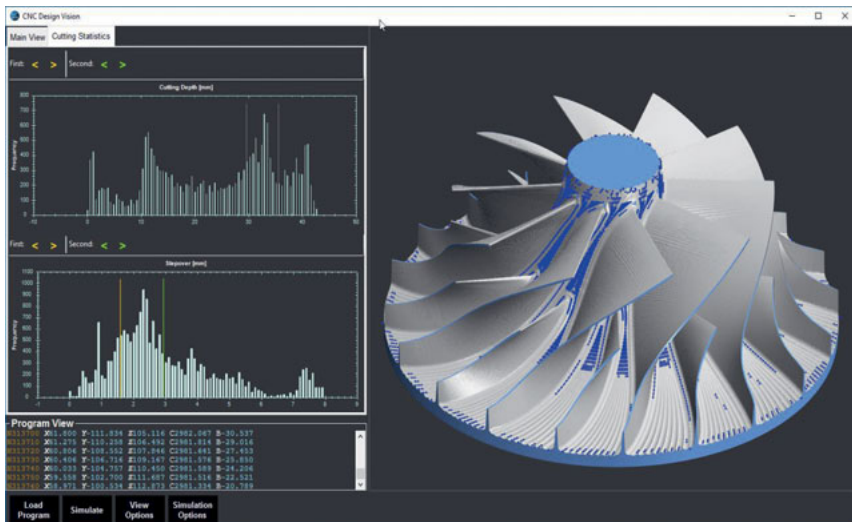


Abbildung 8-9: Einblenden der Schnitttiefe und des Bahnabstands bei der Fertigung (Quelle: ModuleWorks)

Eine Analyse der notwendigen und sinnvollen kontextbezogenen Parameter könnte durch eine Anforderungsanalyse unter Einbeziehen der Benutzer ermittelt werden.

8.7 Literaturverzeichnis

- [1] OpenCV, URL <https://opencv.org/>, Aktualisierungsdatum: 19.12.2017
- [2] ALVAR, URL <http://alvar.erve.vtt.fi/doc/>, Aktualisierungsdatum: 19.12.2017

9 Steuerungsgekoppelte Vorab-Überprüfung des Bearbeitungsprozesses

A. Neyrinck, U. Eger

Der herkömmliche Einsatzbereich von Simulationen im Maschinenbau liegt vorrangig in den Konzeptions- und Inbetriebnahmephasen sowie in der Arbeitsvorbereitung abseits der realen Maschinen. Zunehmend finden Simulationen im HMI der Steuerung in integrierten CAM-Prozessen Anwendung, um NC-Programme und Bearbeitungsergebnisse vorab zu simulieren. Das Einfahren von komplexen Bauteilen auf NC-Maschinen ist dennoch weiterhin meist ein iterativer Vorgang, bei welchem NC-Programmerstellung, Simulation und Fertigung mehrfach durchlaufen werden, um zum gewünschten Fertigungsergebnis zu gelangen. Dies resultiert in hohen Durchlaufzeiten und vermehrtem Ausschuss. Ursache für die Abweichungen zwischen simulierten und realen Fertigungsergebnissen sind unter Anderem unzureichende Abbildung der Maschinendynamik und Abweichungen zwischen CAM-Simulation und realer Bearbeitung mit CNC-Sollwerten der Steuerung. Letztere können durch in der CAM-Simulation teils unberücksichtigte Achsdynamik und unberücksichtigten Synchronisationsvorgängen zwischen CNC und SPS (M-Funktionen) der Steuerung entstehen. Ebenso werden Werkzeugwechselvorgänge in CAM-Simulationen nur unzureichend berücksichtigt.

Mit dem intelligenten Maschinenfenster wird der Werker zum einen in die Lage versetzt, Maschinenabläufe während dem realen Bearbeitungsprozess an einem virtuellen Modell zu beobachten und anhand von computergenerierten Zusatzinformationen besser zu überwachen. Zum anderen kann die Maschine in einen „DryRun“-Modus geschaltet werden, bei dem die Steuerung weitgehend von der Maschine entkoppelt und an das virtuelle Modell gekoppelt wird, um Bearbeitungsprozesse vorab simulieren zu können und somit Unzulänglichkeiten im NC-Teileprogramm vorzeitig erkennen zu können.

Die Verwendung einer realtime-deterministischen Hardware-in-the-Loop-Simulation erlaubt es, die Steuerung komplett auf das virtuelle Modell umzuschalten. Voraussetzung ist eine Kopplung über eine echtzeitfähige Kommunikation. Kommt eine Hardware-in-the-Loop-Simulation bereits während der Inbetriebnahmephase der Maschine zum Einsatz, können die Modelle für die Realisierung eines intelligenten Maschinenfensters einfach wiederverwendet werden.

Für die steuerungsgekoppelte Simulation des Bearbeitungsprozesses, wie sie am iWindow-Demonstrator (Beschreibung in Kapitel 2.8.2) gezeigt wird, benötigt es geeignete Kommunikations- und Softwaremechanismen, auf welche in den folgenden Unterkapiteln eingegangen wird. Zunächst wird in Kapitel 9.1 auf die Steuerungsanbindung eingegangen, welche im iWindow-Projekt als OPC UA-basierende Lösung eine herstellerunabhängige Kopplung zwischen Steuerung und Simulation ermöglicht. In Kapitel 9.2 werden die zum

Einsatz kommenden Visualisierungstechnologien beschrieben. Kapitel 9.3 beschreibt die für die Modellierung und Simulation von Werkstücken, Werkzeugen und Magazinen entwickelten Konzepte und Umsetzungen. Schließlich beschreibt Kapitel 9.4 die Nutzung der zuvor beschriebenen Mechanismen für die Erzeugung von Mehrwertinformationen der Informationsebene „Mehrwertdienste“.

9.1 Steuerungsanbindung mittels OPC UA

Im iWindow-Projekt wurde für das iWindow-Framework eine herstellerunabhängige Steuerungsanbindung mittels OPC UA definiert (siehe Kapitel 5). Hierbei wurden Schnittstellen und Parameter definiert, welche auch die für die Simulation benötigten Parameter umfassen:

Maschinenbewegungen - Hierzu gehören insbesondere die jeweiligen Positionen translatorischer und rotatorischer Maschinenachsen sowie die Prozesssignale zur Ansteuerung pneumatischer und hydraulischer Aktoren.

Beschreibung einzelner Werkzeuge - Die Werkzeugdefinition einzelner Werkzeuge ist über alle untersuchten Steuerungen hinweg unterschiedlich. Über Mapping-Tabellen lassen sich ca. 10 Standardwerkzeuge hinsichtlich Werkzeugtyp und –Parameter zuordnen. Alle anderen rotations-symmetrischen Werkzeugtypen fallen unter den Typ "Profilwerkzeug". Hier erfolgt die Beschreibung über eine 2D-Kontur, z. B. im DXF-Format.

Automatisierte Werkzeuggenerierung - Um Zugriff auf alle in einer Werkzeugverwaltung verwalteten Werkzeuge zu erhalten, bedient man sich einer Exportfunktion der Steuerung. Die Ergebnisse liegen hier in Steuerungshersteller-spezifischer Syntax vor. Die Ergebnisse werden über eine Importfunktion, welche die jeweilige Steuerung berücksichtigt, in die Maschinensimulation eingelesen.

Automatisierte Belegung von Werkzeugmagazinen - Abhängig von der jeweiligen Steuerung existieren hier Exportfunktionen oder ein direkter Zugriff auf einzelne SPS-Objekte, um die Zuordnung der Werkzeuge zu den Lagerplätzen zu erhalten.

SPS-gesteuerter Werkzeugwechsel - Hier werden die Steuersignale zum Ein- und Auswechseln von Werkzeugen in ein oder mehrere Werkzeugspindeln berücksichtigt.

Die Anbindung der in der Simulationsumgebung ISG-virtuos realisierten Simulation des seitlichen Demonstrators an die Steuerung der Demonstratormaschine erfolgt mit dem konfigurierbaren OPC UA-Client. Über Benutzeroberflächen werden die Kommunikationsparameter vorgegeben, der OPC UA Namensraum durchsucht und benötigte Variablen mit den Simulationssignalen verknüpft.

9.2 Visualisierung

9.2.1 3D-Stereo

Um den gesamten Ablauf einer Simulation der Verfahrensvorgänge in der Maschine zu visualisieren, ist das 3D-Rendering mit der NVidia Quadro Quad Buffered Professional Stereo-Technologie [1] umgesetzt. Diese erlaubt eine Echtzeit-Anzeige des Maschineninnenraum-Simulationsmodells auf 3D-Displays mit Shutter- oder Polfiltertechnologie. Hierfür wird für jedes Auge eine eigene Ansicht aus geringfügig abweichenden Perspektiven berechnet. Während die beiden zuletzt berechneten Bilder angezeigt werden, werden im Hintergrund die nächsten beiden Bilder gerendert. Insgesamt werden also vier Bildpuffer (links, rechts, links Hintergrund, rechts Hintergrund) genutzt.

Für eine exakte Darstellung muss der normalisierte Augenabstand des Betrachters sowie der Null-Parallaxen-Abstand eingestellt werden. Der normalisierte Augenabstand ist der reale Augenabstand (üblicherweise ca. 63 mm [2]) bezogen auf die Breite des verwendeten Displays. Die Parallaxe bezieht sich auf den Positionsunterschied in der Jeweiligen Ansicht (rechtes Auge, linkes Auge) eines auf dem Display dargestellten Punktes [3]. Die Parallaxe ist Null wenn die korrespondierenden Punkte beider Ansichten an der gleichen Stelle des Displays angezeigt werden. Wird für den Null-Parallaxen-Abstand der tatsächliche Abstand des Betrachters zum Display eingegeben, erscheinen Objekte in diesem Abstand in der Display-Ebene, wenn auch die virtuelle Kamera korrekt (zwischen beiden Augen des Betrachters, z.B. durch Head-Tracking) platziert ist. Die Einstellung eines größeren Wertes bewirkt, dass Objekte näher an den Betrachter rücken und somit vor dem Display angezeigt werden; die Einstellung eines kleineren Wertes bewirkt die Anzeige der Objekte hinter der Display-Ebene.

9.2.2 Perspektivische Darstellung als Fenster

Auf dem intelligenten Maschinenfenster kann dieselbe Darstellung wie bei einem realen Fenster erreicht werden. Wo an Maschinen eigentlich kein Fenster möglich ist oder wenn eine Maschine vollständig und ohne Fenster verkleidet werden soll, kann hierdurch ein Fenster imitiert werden. Da ein herkömmliches Display nur eine Perspektive darstellen kann, wird lediglich für einen einzelnen Betrachter die korrekte Perspektive angezeigt. Abhängig von der durch das Head-Tracking-System erfassten Kopfposition muss das auf dem Display dargestellte Bild in der richtigen Perspektive berechnet werden. Die theoretischen Grundlagen hierfür sind in Kapitel 6.5.3 beschrieben.

Die nötigen Algorithmen wurden in das Simulationssystem ISG-virtuos in Form eines speziellen Matrix-Kamera-Typs integriert, bei welchem die über das Head-Tracking erfasste Betrachterposition, die Position des Trackingsystems sowie die Position des Displays bei der perspektivischen Visualisierung berücksichtigt werden. Abbildung 9-1 zeigt die resultierende Visualisierung als Seitenfenster des iWindow-Demonstrators DMU 50. Für die Aufnahmen wurde 3D-Stereo ausgeschaltet.

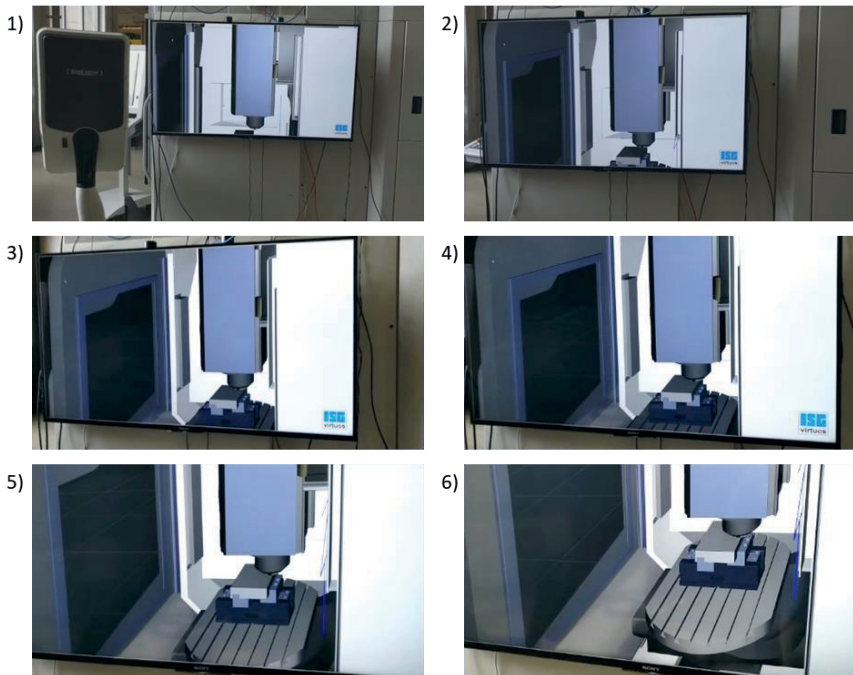


Abbildung 9-1: Perspektivische Fenster-Visualisierung bei Annäherung des Betrachters an das intelligente Maschinenfenster (Quelle: ISG)

Als Head-Tracking-System kommt am Demonstrator das Advanced Realtime Tracking SMARTTRACK-System [4] zum Einsatz. Für die Anbindung dieses und weiterer Systeme anderer Hersteller wurde in ISG-Virtuos eine TCP/IP-Schnittstelle integriert, welche die erfassten Positionen und Orientierungen entgegen nimmt. Wahlweise können die empfangenen Daten vor der Weiterverarbeitung mit einem Mittelwert-Filter geglättet werden, um Rauschverhalten der vom Headtracking-System erfassten Betrachterposen zu stabilisieren.

Abbildung 9-2 zeigt die Konfigurationsoberfläche für das Head-Tracking und die perspektivische Fenster-Darstellung.

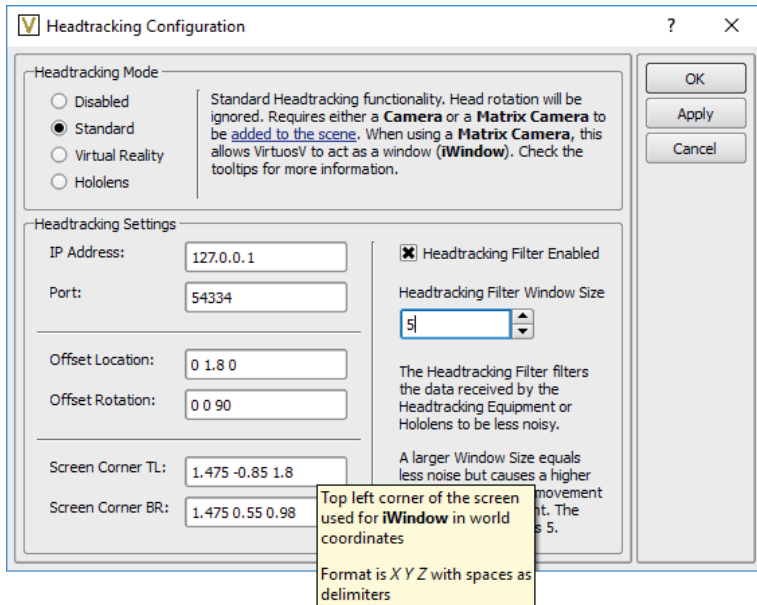


Abbildung 9-2: Konfigurationsoberfläche für das Head-Tracking und die perspektivische Darstellung (Quelle: ISG)

9.2.3 Oberflächeneigenschaften

Um die Akzeptanz des intelligenten Maschinenfensters bei Anwendern zu erreichen, bedarf es einer hochauflösenden, fotorealistischen und flüssigen Darstellung der Simulationsvisualisierung in Echtzeit. Die Übernahme von 3D-CAD-Daten aus der Mechanikkonstruktion ist hierfür der erste Anfang. Die CAD-Daten enthalten idealerweise bereits realistische Farbinformationen (oft sind die Farben aus der Mechanikkonstruktion jedoch für die bessere Unterscheidbarkeit von Baugruppen- und Konstruktionsmerkmalen gedacht und nicht für eine realistische Anzeige). Neben der Darstellung von farbigen CAD-Daten existiert eine Vielzahl an teils echtzeitfähigen Bildsyntheseverfahren, um die Visualisierung realitätsgetruer zu gestalten. Lichtquellen und Schattenwurf, Mapping-Techniken wie Texturüberlagerung, Bump Mapping und Environment Mapping sind die praxisrelevantesten der echtzeitfähigen Techniken. Im iWindow-Projekt wurde ISG-virtuos um Funktionen zur Texturüberlagerung und für Environment Mapping erweitert.

Bei der Texturüberlagerung werden zweidimensionale Bilder (Textures) über die dreidimensionalen Objektoberflächen gelegt. Um die (Kamera-) Bilder auf beliebig geformten Objekten korrekt aufzubringen, werden die CAD-Daten mit sogenannten Texturkoordinaten angereichert. Dies geschieht im CAD-System oder in hierfür geeigneten Softwarewerkzeugen wie Autodesk Maya® oder Blender. Abbildung 9-3 zeigt die Fronttür der

Demonstratormaschine ohne Texturüberlagerung; Abbildung 9-4 zeigt sie mit einem Foto als Textur.

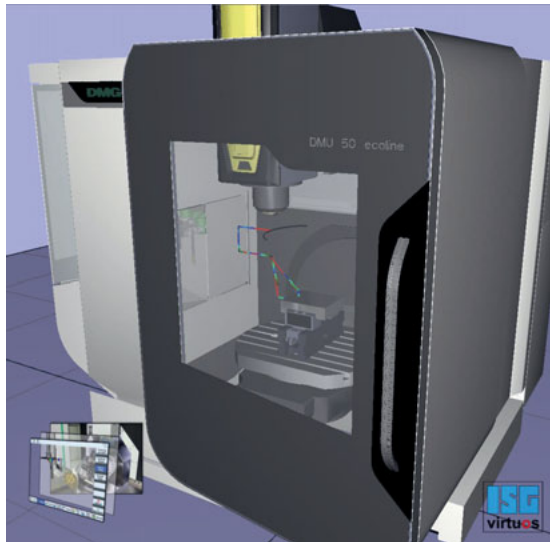


Abbildung 9-3: 3D-Visualisierung der Demonstratormaschine ohne Texturüberlagerung der Front-Tür (Quelle: ISG)

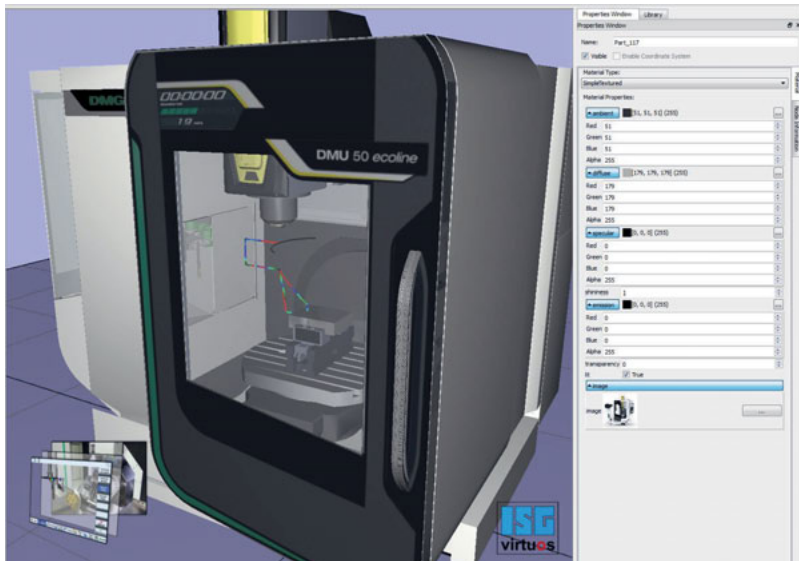


Abbildung 9-4: 3D-Visualisierung der Demonstratormaschine mit Texturüberlagerung der Front-Tür (Quelle: ISG)

Environment Mapping kommt zum Einsatz, wenn spiegelnde oder reflektierende Oberflächen visualisiert werden, beispielsweise bei metallischen Maschinenteilen oder Werkstücken. Hierfür wird ein Bild der Umgebung des Objekts als Textur auf die Innenfläche einer Hohlkugel (sphärisches Environment Mapping) oder eines Würfels (kubisches Environment Mapping) gelegt. Bei der Berechnung der darzustellenden Farbe jedes Pixels des Objekts wird dann der Reflexionsvektor bestimmt und die Farbe der Textur angewendet, auf die der Reflexionsvektor zeigt. Abbildung 9-5 zeigt das Resultat auf dem Werkstück der Demonstratormaschine.

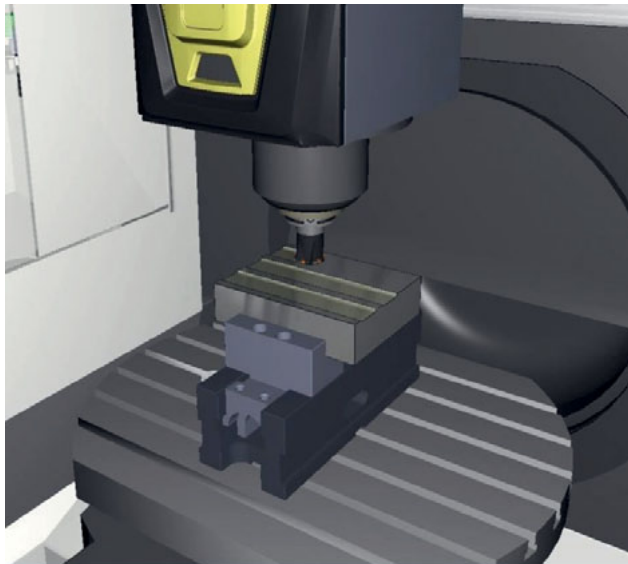


Abbildung 9-5: 3D-Visualisierung der Demonstratormaschine mit spiegelnder Metalloberfläche des Werkstücks (Quelle: ISG)

9.3 Modellierung und Simulation von Werkstücken, Werkzeugen und Magazinen

Um den Bearbeitungsprozess für die NC-Programmvalidierung im intelligenten Maschinenfenster realitätsnah mit Abtrags- und Kollisionssimulation abzubilden, müssen Werkstücke und Werkzeuge definiert, sowie deren Bewegungen durch die Maschine modelliert werden. Die Erstellung virtueller Modelle für das intelligente Maschinenfenster wird vom Maschinenhersteller vorgenommen. Er muss in der Lage sein, seine Maschine effizient virtuell abzubilden.

Für die **Definition von Werkstücken** wurden drei unterschiedliche Möglichkeiten geschaffen. Für die Erzeugung einfacher Rohteile können die Werkstücktypen Quader (typisch beim Fräsen) und Zylinder (typisches Drehwerkstück) ausgewählt und parametrisiert werden. Um komplexere Rohteile oder Halbzeuge zu definieren, können beliebige Werkstückkonturen aus einem CAD-Daten-Import, auch während der Simulation dynamisch, erzeugt werden.

Abbildung 9-6 zeigt ein manuell über die Symbolleistenfunktionen der Bedienoberfläche (rot umrandet) erzeugtes quaderförmiges Rohteil. Als Unterstützung bei der Analyse der Bearbeitungsergebnisse kann für bearbeitete Flächen eine eigene Farbe eingestellt werden.

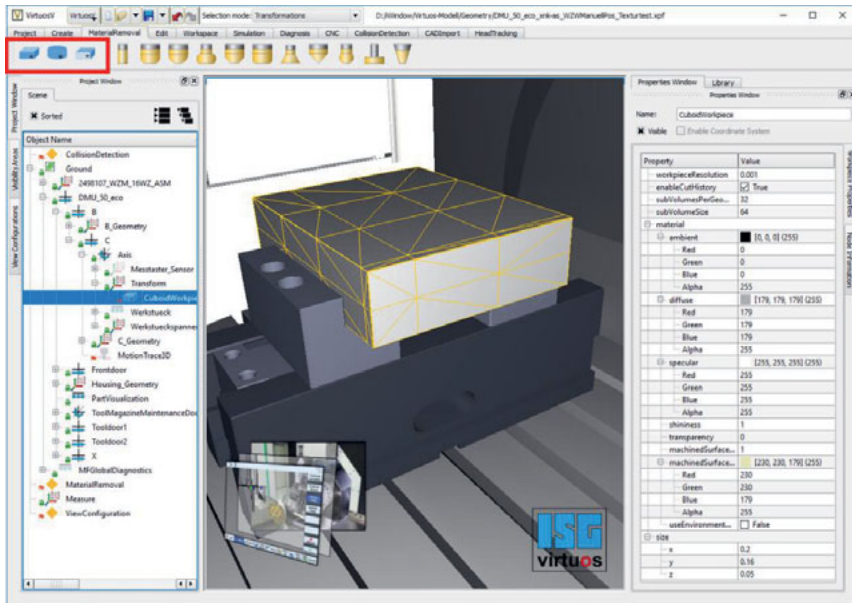


Abbildung 9-6: Quaderförmiges Werkstück in der iWindow-Demonstratormaschine (Quelle: ISG)

Für die **Definition von Werkzeugen** werden zum einen Templates für Standardfräs- und Bohrwerkzeuge (Schafffräser, Radiusfräser,...) verwendet. Zum anderen können beliebige rotationssymmetrische Profilwerkzeuge über einen 2D-Konturzug definiert werden. Die rotationssymmetrischen Geometrien können für eine realitätsnahe Darstellung mit detaillierten 3D-Geometrien überlagert werden. Abbildung 9-7 zeigt über Templates (rot umrandet) definierte Werkzeuge mit überlagerter 3D-Geometrie; die Template-Geometrie (gelbe Gitterdarstellung) kann während der Simulation unsichtbar geschaltet werden, sodass ein sehr realistischer Bearbeitungsvorgang visualisiert wird.

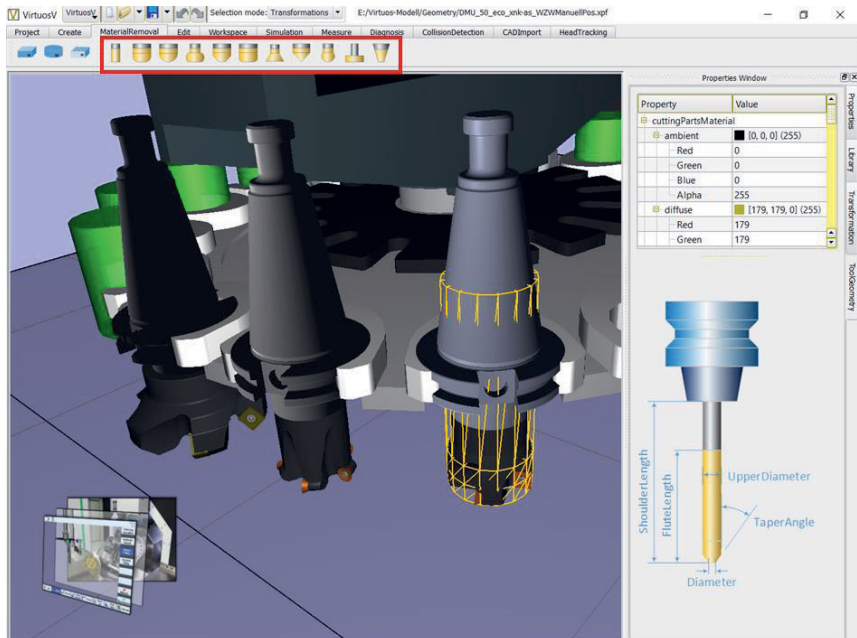


Abbildung 9-7: Werkzeugdefinition über Templates und überlagerte 3D-Geometrie (Quelle: ISG)

Die Werkzeuge werden mit Hilfe der Werkzeughandling-Bibliothek in die Magazine und weitere Werkzeugplätze in der Maschine (bspw. Spindel) eingelastet. Der Werkzeugwechsel erfolgt auf Grundlage der Steuerungssignale; im Falle der Demonstratormaschine wird das gesamte Tellermagazin durch einen Pneumatikzylinder in den Arbeitsraum der Maschine eingerückt und durch einen Elektromotor gedreht. Die Übergabe eines Werkzeugs von einem Werkzeugplatz des Tellermagazins in die Spindel wird ausgeführt, wenn der Spindel-Werkzeugplatz sich über einem Werkzeugplatz des Magazins befindet und das Steuerungssignal zum Spannen der Spindel anliegt.

Die Belegung des Werkzeugmagazins und der Spindel zu Beginn der Simulation wird über OPC UA mit der Steuerung synchronisiert. Ebenso wird, wenn ein Werkzeug (bspw. aufgrund des Ablaufes der Standzeit) aus der Maschine entnommen und ein neues eingewechselt wurde, eine Synchronisation durchgeführt. Hierfür werden die Werkzeugnummern für die jeweiligen Werkzeugplätze von der Steuerung in einem Array bereitgestellt.

9.4 Technische Integration & Anwendung von Mehrwertinformationen

9.4.1 Abtragsimulation

Ziel dieses Mehrwertdiensts besteht darin, Fehler im NC-Teileprogramm frühzeitig während des Einrichtbetriebs zu erkennen sowie den Bearbeitungsfortschritt auf dem intelligenten Maschinenfenster anzuzeigen.

Für die Abtragsimulation wurde der in Kapitel 8.2 beschriebene Abtragskern des Projektpartners ModuleWorks in ISG-virtuos integriert. Da die Berechnung des Materialabtrags nicht in Steuerungszeit möglich ist, wurde sie in die Visualisierungskomponente ISG-virtuosV integriert (Abbildung 9-8). Um sicherzustellen, dass jeder Steuerungssollwert in der Abtragsimulation berücksichtigt wird, musste der Kommunikationskanal zwischen Simulationsrechnern und Visualisierung um geeignete Puffermechanismen erweitert werden.

Für die Versorgung des Abtragskerns mit den nötigen Daten sind neben den in Kapitel 8.2 beschriebenen Maschinendefinitionen und den in Kapitel 9.3 beschriebenen Werkzeug- und Werkstückmodellen weitere Parameter zu übergeben. Die „Workpiece Resolution“ beispielsweise gibt die Vernetzungsdichte der bei der Abtragsberechnung kontinuierlich neu erzeugten Werkstückgeometrie an und hat großen Einfluss auf die benötigte Rechenleistung.

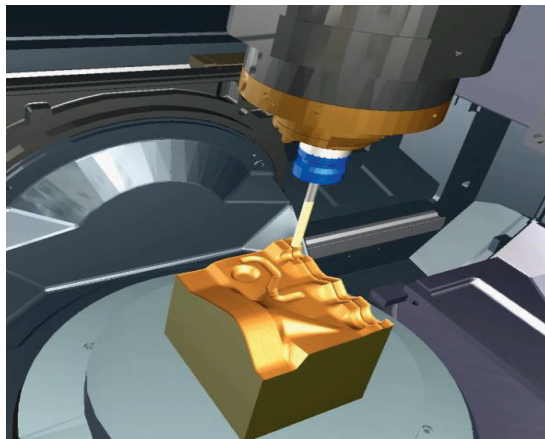


Abbildung 9-8: Visualisierung des Materialabtrags (Quelle: ISG)

9.4.2 Spurbildung

Ziel dieses Mehrwertdiensts besteht darin, Fehler im NC-Teileprogramm frühzeitig während des Einrichtbetriebs zu erkennen. Hierfür wird die Bahn des bereits zurückgelegten beziehungsweise des noch bevorstehenden Bearbeitungspfads in den virtuellen Maschineninnenraum eingeblendet. Die Berechnung des bereits zurückgelegten bzw.

aktuellen Bearbeitungspfads basiert auf den Positionssollwerten aus der Maschinensteuerung, während der noch bevorstehende Bearbeitungspfad z.B. aus dem NC-Programm abgeleitet werden kann.

Die Spurbildung (Bahnvisualisierung) ist in die Informationsebene „Maschineninnenraum“ integriert. Die folgenden Abbildungen (Abbildung 9-9, Abbildung 9-10) zeigen die resultierende Bahnvisualisierung mit eingestelltem Linientyp „durchgezogene Linie“ und geschwindigkeits-, bzw. prozessparameterabhängigem Farbwechsel.

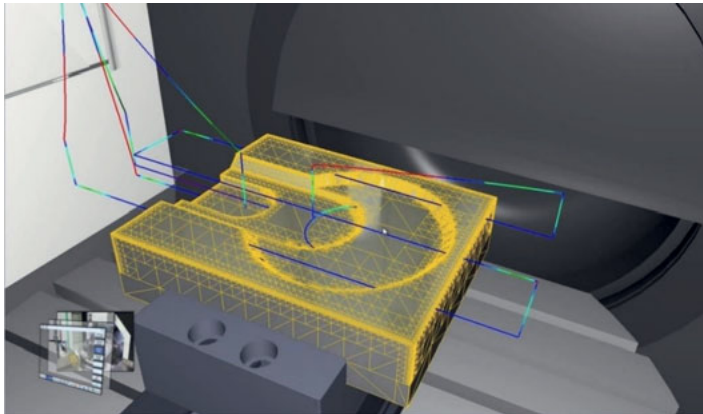


Abbildung 9-9: Bahnvisualisierung als durchgezogene Linie mit geschwindigkeitsabhängigem Farbverlauf (Quelle: ISG)

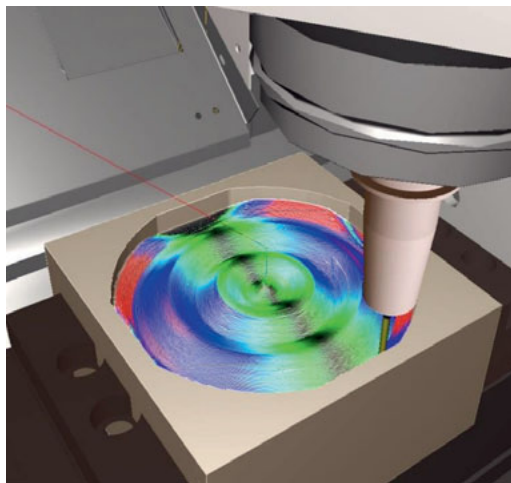


Abbildung 9-10: Abgetragenes Werkstück mit überlagertem prozessparameterabhängigem Farbverlauf (Quelle: ISG)

9.4.3 Kollisionserkennung

Bei der Erkennung von Kollisionen wird zwischen Offline- und Online-Erkennung unterschieden. Offline-Kollisionserkennung findet vor Beginn der realen Bearbeitung statt. Online-Kollisionserkennung findet mit einem nur geringen zeitlichen Vorsprung vor der realen Bearbeitung statt, sodass zumindest das rechtzeitige Anhalten der Maschine gewährleistet ist. Die Steuerung muss hierfür neben den aktuellen Achs-Sollpositionen auch zukünftige Positionen berechnen und der Kollisionserkennung rechtzeitig bereitstellen.

Ebenso wie die Abtragsimulation ist die Kollisionserkennung mit umfangreichen CAD-Daten nur mit starken Restriktionen in Steuerungszeit rechenbar. Sie wurde deshalb nicht in den Echtzeitrechnern, sondern in die Visualisierungskomponente ISG-virtuosV integriert. Auf dem Markt verfügbare Kollisionsbibliotheken unterscheiden sich hauptsächlich bezüglich Rechenperformanz der verwendeten Algorithmen, Toleranz gegenüber Fehlern in den CAD-Daten, Fähigkeit Berechnungen auf der GPU auszuführen, Fähigkeit Geometrien dynamisch zu verändern (Berücksichtigung des Abtragsfortschritts) und Multicore-Fähigkeit. Abhängig vom jeweiligen Anwendungsfall (bspw. Simulation einer Werkzeugmaschine vs. Simulation einer Roboterzelle für Karosseriebearbeitung) eignen sich unterschiedliche Bibliotheken am besten. In ISG-virtuos wurde deshalb eine bibliotheksunabhängige Schnittstelle integriert, welche unterschiedliche Bibliotheken einheitlich anbindet. Am iWindow-Demonstrator kommt die Bibliothek des Projektpartners ModuleWorks zum Einsatz. Die hierbei verwendeten Datenformate und Schnittstellen sind in Kapitel 8.2 beschrieben.

Bei der Umsetzung der Kollisionsüberwachung für die Simulation im Parallelbetrieb hat sich gezeigt, dass für die großen Datenmengen üblicher Maschinenmodelle die Ausführung auf der realen Zeitachse eine Herausforderung ist und einer Optimierung bedarf. Hier kommt sowohl eine Beschleunigung der Berechnungen durch Auslagerung auf die GPU in Frage, als auch die Reduktion der CAD-Geometrien der Kollisionsobjekte durch geeignete Reduktionsalgorithmen.

Für die Auslagerung der Berechnungen auf die GPU wurde die Kollisionsberechnung in die NVidia-PhysX-Engine portiert. Die Evaluierung des Ansatzes liefert vielversprechende Ergebnisse, die voraussichtlich mit dem nächsten Release der PhysX-Engine noch übertroffen werden, da Teile der PhysX-Engine aktuell noch auf der CPU anstatt auf der GPU ausgeführt werden.

Für die Reduktion der Geometrien wurden mehrere Ansätze evaluiert und zwei davon für die Umsetzung ausgewählt. Als für einen industrietauglichen Einsatz entscheidende Eigenschaften wurden folgende Kriterien identifiziert:

1. Die Algorithmen müssen in der Lage sein, die Geometrien einstellbar um bis zu 75% zu reduzieren

2. Die Ergebnis-Geometrie der Reduktion darf einen maximalen Abstand zur ursprünglichen Geometrie nicht überschreiten, damit falsch-positive Kollisionsereignisse erst ab einem definierten Abstand der tatsächlichen/ursprünglichen Geometrien auftreten.
3. Die Ergebnis-Geometrie der Reduktion darf nur außerhalb der ursprünglichen Geometrie liegen, damit bei der Kollisionsberechnung mit der reduzierten Geometrie jede Kollision erkannt wird, welche auch mit der ursprünglichen Geometrie erkannt würde.
4. Die Ergebnis-Geometrie der Reduktion sollte in mehrere Teilgeometrien zerlegt sein, um in der „Narrow Phase“ der Kollisionsberechnung von (massiver) Parallelisierung profitieren zu können.

Zum einen wurde der VHACD- Algorithmus (Volumetric-Hierarchical Approximate Convex Decomposition) [5] umgesetzt, welcher die Kriterien eins, drei und vier erfüllt. Zum anderen wurde der ModuleWorks-Reduktionsalgorithmus angebunden, der die Kriterien eins und drei erfüllt.

Um sicherzustellen, dass jeder Steuerungssollwert in der Kollisionssimulation berücksichtigt wird, kommen die bereits in Abschnitt 9.4.1 erwähnten Puffermechanismen zum Einsatz.

Abbildung 9-11 zeigt die Visualisierung einer Kollision zwischen Werkzeugschneide und Spanblock der iWindow-Demonstratormaschine während des Bearbeitungsvorgangs.

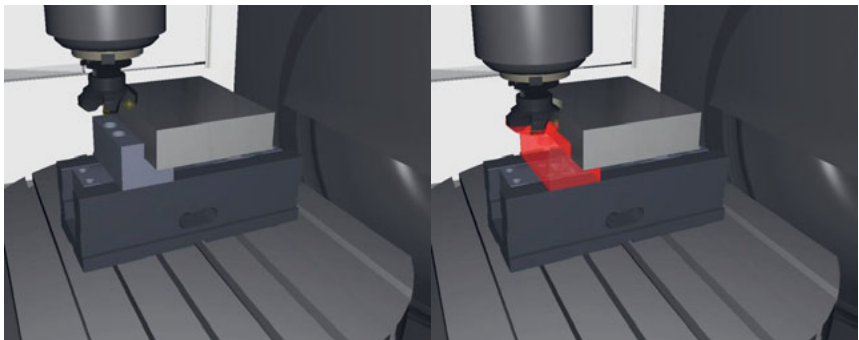


Abbildung 9-11: Kollisionsvisualisierung der Demonstratormaschine in ISG-virtuos
(Quelle: ISG)

9.4.4 Schnelle Kommunikation

Zur Kopplung der innerhalb und außerhalb der Steuerungsechtzeit gewonnenen Informationen benötigt es eine schnelle Kommunikation zwischen Maschineninnenraumvisualisierung und der Verhaltenssimulation, welche die Werkzeuge verwaltet, die Werkzeugmagazine realisiert und auch den Materialfluss steuert. Dieser echtzeitfähige Kommunikationskanal transportiert alle Informationen aus der Abtragsimulation von ModuleWorks im zeitlichen Kontext. Weiterhin kann die Information

über eine Kollision – welches Objekt hat zu welchem Zeitpunkt mit welchen Objekt kollidiert – in Echtzeit transportiert werden.

Die Herausforderung bei der sehr aufwändigen Umsetzung des Kommunikationskanals besteht darin, dass er aufgrund der Unterschiedlichkeit der zu transportierenden Daten – Werkstückgeometrien, Kollisionsinformationen, Abtragsinformationen bspw. beim Trennen von Teilen, Werkstück-Materialfluss, Werkzeug-Materialfluss – eine große Flexibilität bezüglich des Übertragungsinhalts erfordert. Gleichzeitig sind die Anforderungen bezüglich der Echtzeitfähigkeit komplex, da sowohl echtzeitfähige Komponenten des Simulationssystems (Simulationsrechenkerne), als auch nicht-echtzeitfähige Komponenten miteinander kommunizieren müssen. In der realisierten Umsetzung kommen hierfür zwei Kommunikationsarten zum Einsatz. Zum einen eine Kommunikation nach dem Producer/Consumer-Schema (bspw. zur Übertragung von Simulationszuständen aus dem Echtzeitrechenkern zur Visualisierung); zum anderen eine ereignisorientierte Kommunikation (beispielsweise zur Übermittlung von Teile-Trenn-Informationen oder Kollisionsrückmeldung aus der im Visualisierungsteil berechneten Abtragsimulation in den Echtzeitrechenkern). Für die geforderte Echtzeitfähigkeit der Kommunikationsseite des Simulationsrechenkerns haben sich folgende Mechanismen als notwendig herausgestellt:

- Puffermechanismen
- Echtzeitfähiges Speichermanagement (Speicherallokation und -Freigabe)
- Echtzeitfähiges Hinzufügen und Löschen von Kommunikations-Nodes
- Übertragung von Referenzen zwischen den übertragenen Kommunikations-objekten (verzeigerte Field-Container), vernetzte Strukturen ähnlich wie bei OPC UA.

9.5 Literaturverzeichnis

- [1] NVIDIA Corporation; *Quadro Quad Buffered Professional Stereo Technology*. Online: http://www.nvidia.de/object/quadro_stereo_technology_uk.html
- [2] Dodgson, Neil A., et al.: *Variation and extrema of human interpupillary distance*. In: *Proceedings of SPIE*, 2004, S. 36-46.
- [3] Gateau, Samuel; Neuman, Robert; Salvati, Marc: *Stereoscopy, From XY to Z*. Siggraph 2011 Stereoscopy Course, 2011. Online: http://developer.download.nvidia.com/assets/gamedev/docs/Siggraph2011-Stereoscopy_From_XY_to_Z-SG.pdf
- [4] Advanced Realtime Tracking GmbH: *SMARTTRACK Produktbeschreibung*. Online: <https://ar-tracking.com/products/tracking-systems/smarttrack/>
- [5] Mamou, Khaled; Ghorbel, Faouzi: *A simple and efficient approach for 3d mesh approximate convex decomposition*. In: *Proceedings of the 16th IEEE international conference on Image processing ICIP'09*, s. 3465-3468, Piscataway, NJ, USA, 2009.

10 3D-Rekonstruktion

W. Schotte

10.1 Einleitung und Zielsetzung

Ziel der 3D-Rekonstruktion im Projekt ist es, 3D-Teilobjekte im Arbeitsraum der Werkzeugmaschinen wie beispielsweise Spannmittel oder das Werkstück zu erkennen und bezüglich der Werkzeugmaschinenkoordinaten zu rekonstruieren und zu verorten. Spannmittel sind üblicherweise nicht in der Konstruktion der Werkstücke enthalten bzw. werden vom Bediener der Werkzeugmaschine angebracht, ohne dass der Werkzeugmaschine Position und Orientierung bekannt gegeben werden. Diese Spannmittel stellen eine potentielle Gefahr für den Werkzeugkopf bei Kollision dar. Zudem scheint es interessant auch weitere Objekte im Arbeitsraum der Werkzeugmaschinen zu erkennen, wie beispielsweise Fremdkörper oder sogar Personen bzw. den Eingriff des Benutzers in die Maschine, um beispielsweise die Sicherheit für Maschine und Bediener weiter zu verbessern.

Für die 3D-Rekonstruktion gibt es zahlreiche technische und methodische Ansätze, aus unterschiedlichen Informationen und bildgebenden Aufnahmeverfahren, eine Rekonstruktion der Objekte im Sichtbereich zu ermöglichen (Abbildung 10-1) [1,2,3].



Abbildung 10-1: Stereokamera oCamS-1CGN-U (links), ToF-Kamera (Mitte), Structured Light Scanner (rechts) (Quelle: [1,2,3])

Allgemein betrachtet, basieren die Verfahren, die zweidimensionale Bildgebung benutzen, darauf, gemeinsame Features in Bildern mit unterschiedlicher Perspektive zuzuordnen (Kapitel 10.2.1), um so eine Tiefeninformation der Features bei bekannter Kameraposition zu berechnen. 3D Aufnahmeverfahren wie die Time-of-Flight-Kameras (ToF; Kapitel 7.2 und Kapitel 10.2.2) nutzen die durch den Sensor ermittelte Tiefe, um ein dreidimensionales Bild der Objekte vor der Kamera zu erstellen. Verfahren zur Bestimmung der Tiefeninformation sind neben dem ToF-Prinzip beispielsweise auch Structured Light System, die ein Muster in die Szene projizieren und die auf die Objekte projizierte Verformung dieses Muster nutzen, um Tiefeninformation und Objektform zu rekonstruieren. Die Möglichkeiten der Erfassung und Qualität der Bildaufnahme ist wesentlich für die 3D-Rekonstruktion.

Die gegebenen Bedingungen in der Werkzeugmaschinen schränken die Aufnahmeverfahren teils sehr ein bzw. reduzieren die Auswahl auf Spezialkameras, die beispielsweise bei kleinen Arbeitsräumen das Messprinzip zur Gewinnung der Tiefe sehr nahe der Kamera ermöglichen.

Die geometrische Anordnung von LEDs und ToF-Sensor bei ToF-Kameras oder von Projektor und Kamera bei Structured Light Scanner gibt hier neben den messtechnischen Prinzipien bereits die Arbeitsdistanzen vor. Eine zweite Einschränkung sind die in der Werkzeugmaschine zu erwartenden Materialien. Üblicherweise werden hier rohe oder blanke Metallteile als Rohteil in die Maschine zur weiteren Verarbeitung gebracht, deren Aufnahmen beispielsweise durch spiegelnde Lichtreflexion erschwert sind.

Um die 3D-Rekonstruktion auch von bewegten Objekten, wie zum Beispiel dem Werkzeugkopf, Werkstücke die in die Werkzeugmaschine eingebracht oder entnommen werden, zu ermöglichen, muss die Bildaufnahme im Kamerasystem sowie die eigentlich Rekonstruktion in annähernd Echtzeit erfolgen. Die Bildaufnahme und Verarbeitung mehrerer Bildströme in Echtzeit erfordert ein Rechnernetzwerk und einen Rechner mit entsprechend hoher Bandbreite. Selbst moderne Workstation-PCs sind mit der gleichzeitigen Aufnahme von mehreren HD Bildströmen schnell überfordert. Gängige Verfahren zur 3D-Rekonstruktion (vgl. Kapitel 10.4) nutzen ebenfalls deutlich performantere Rechnersysteme bzw. GPU-Systeme, wie sie normalerweise nicht in einer Werkzeugmaschine verbaut werden.

10.2 Evaluierung der Methoden zur 3D-Rekonstruktion

10.2.1 Stereo-Kameras

Wie bereits erwähnt, beruht das Prinzip der Stereokameras darauf, Features, die in beiden Perspektiven der Stereokameras detektiert worden sind, zuordnen zu können und mit Hilfe der gegebenen Orientierung beider Kameras zueinander, die Tiefe zu berechnen. In einer Reihe von Versuchen sind Stereokamerakonfigurationen am Prototypen und insbesondere an der Palette der TRUMPF Laserschneidmaschine getestet worden (Abbildung 10-2).

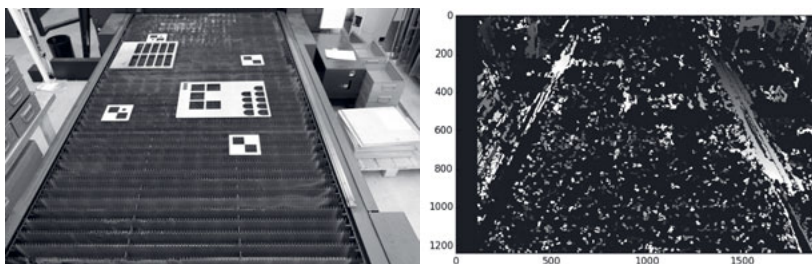


Abbildung 10-2: Rechtes Bild einer Stereokamera (links) Ergebnis eines Stereokameraalgorithmus (rechts) (Quelle: TRUMPF, HLRS)

Die Palette der Laserschneidmaschine, auf der die Bleche aufgelegt und nach der Bearbeitung absortiert werden sollen, besteht aus einzelnen Kupferlamellen, die in 10 cm Abstand voneinander quer entlang der Palette montiert sind. Man kann in den Bildern erkennen, dass das Verfahren zur Rekonstruktion dort gut arbeitet, wo einfache Geometrien im Bild zu finden sind. Im Bereich der Palette liefern die Verfahren zum Teil ein Rauschen oder im schlechtesten

Fall gänzlich fehlerhafte Ergebnisse (Abbildung 10-2). Features lassen sich im Bild nur schwer dort erkennen, wo beispielsweise unbearbeitetes Blech liegt, weil dieses über keinerlei bzw. wenig Struktur verfügt. Dort, wo Features zum Beispiel auf der Palette erkannt werden, ist die Zuordnung der Features in den beiden Bildern schwierig, da sich die Muster im Bild wiederholen.

Das Verfahren zur Rekonstruktion der Oberfläche der Palette lässt sich nach Feinjustierung einiger Parameter für den gegebenen Fall anwenden, eine zuverlässige Rekonstruktion in den verschiedenen Fällen mit unterschiedlichen oder wenig Blechen funktionierte in den Versuchen nicht zuverlässig.

10.2.2 ToF-Kameras

Das Messprinzip der ToF-Kameras wurde bereits in Kapitel 7.2 erläutert. Nach einer Kalibrierung der ToF-Kamera funktioniert die Messung der Tiefe mit Hilfe der in Kapitel 7.2 angesprochenen diskreten Korrekturfunktion f relativ gut. Ein großes Problem stellen aber die im Projekt benutzten Materialien dar, die detektiert und vermessen werden sollen. Die Technik der ToF-Kameras nutzt üblicherweise modulierte Infrarot-Licht, um aus der Phasenverschiebung von ein- und ausgehenden Licht, die Tiefe des Objekts, das das Licht zurückreflektiert, zu messen. Abbildung 10-3 zeigt das Frequenzspektrum des im BlueTechnix MTK-ETH Developer Board verbauten Time-of-Flight Sensormodules sowie den IR-Durchlassbereich des verbauten Filters.

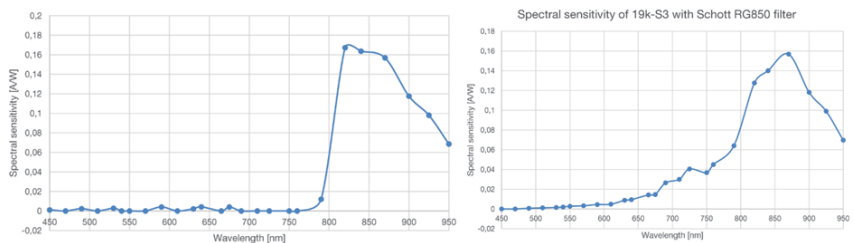


Abbildung 10-3: Frequenzspektrum des verwendeten ToF-Sensors (links) und Filter (rechts) des BlueTechnix MTK-ETH Developer Boards (Quelle: BlueTechnix)

Das ToF-Sensormodul hat eine Empfindlichkeit für Infrarotlicht ab etwa 800 nm. Die beste Empfindlichkeit und Durchlassbereich des genutzten Filters liegt um die Wellenlänge von 850 nm, was ebenfalls ein typischer Wellenlängenbereich für gängige IR-LEDs ist und so auch in anderen ToF-Kameras zum Einsatz kommt. Dagegen zeigt Abbildung 10-4 das Absorptionsspektrum verschiedener Metalle, wie sie beispielsweise in der Blechverarbeitung mittels Laserschneiden eingesetzt werden.

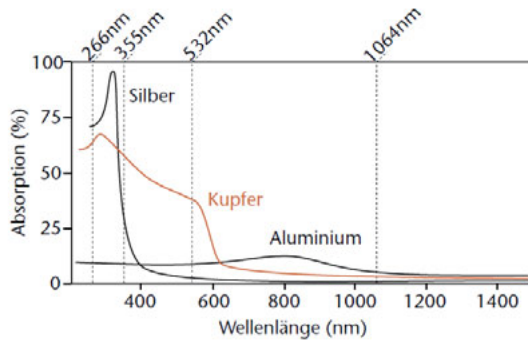


Abbildung 10-4: Absorptionsspektrum verschiedener Metalle (Quelle [4])

Wenn für Transmissionsgrad τ , Absorptionsgrad α_r und Reflexionsgrad ρ gilt $\tau + \alpha_r + \rho = 1$ und man annehmen darf, dass für Bleche der Transmissionsgrad $\tau = 0$ ist, kann man aus Abbildung 10-4 das Reflexionsverhalten der verschiedenen Materialien im Wellenbereich 850 nm ablesen. In einer Versuchsreihe bei TRUMPF wurde dies mit einer Reihe im Laserschneiden verwendeter Blechmaterialien verifiziert (Abbildung 10-5).

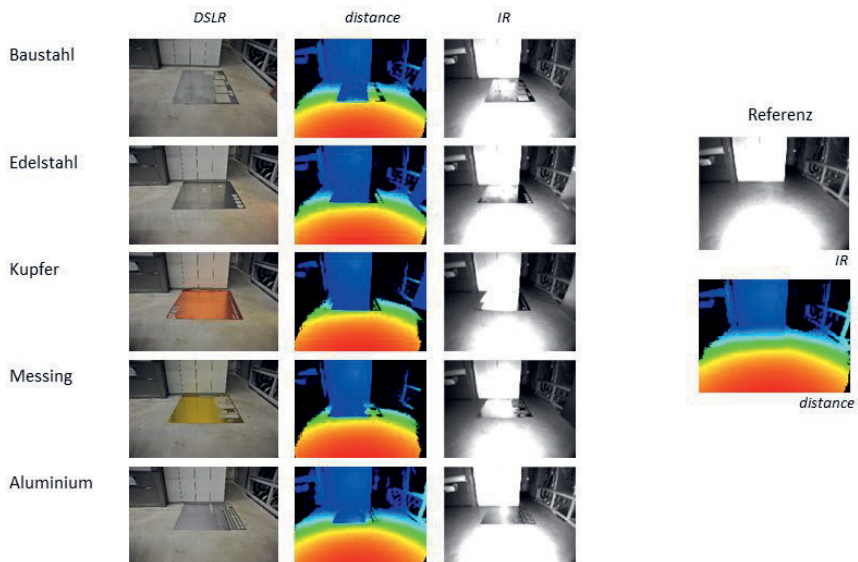


Abbildung 10-5: Reflexionen verschiedener Materialien im IR-Licht / Sichtbarkeiten für den ToF-Sensor (Quelle: TRUMPF, HLRS)

Die unterschiedlichen Bleche erscheinen im Bild der Spiegelreflexkamera (DSLR) sowie im IR-Rohbild des ToF-Sensors noch gut wahrnehmbar, der ToF-Sensor misst allerdings die Entfernung nicht zum Hallenboden sondern die Entfernung zur gespiegelten, dahinter liegenden Wand. Die Materialien bzw. das Abbild vom Boden erscheinen in den Tiefendaten nicht bzw. nur durch vereinzelte Pixel/Tiefendaten. Einen weiteren Versuch zur Vermessung eines aus Metall gefrästen Werkstücks zeigt Abbildung 10-6.



Abbildung 10-6: ToF-Testaufnahme (Tiefenbild links, IR Mitte) und 3D-Rekonstruktion (rechts) eines gefrästen Testobjekts (Quelle: HLRS)

Deutlich zu erkennen, sind die Reflexionen des LED-Lichts in den Raum (schwarz/ohne Messwert) als auch die direkten Reflexionen zurück zur Kamera (weiß). Die rekonstruierte Punktwolke zeigt keine stetige Oberfläche sondern deutliche Ausreißer und Fehler in den Messergebnissen des ToF-Sensors.

Die ToF-Sensoren der getesteten Kameras zeigen, dass sich die Aufnahmen von blanken oder selbst matten Metallen wie erwartet als schwierig gestalten. Messfehler entstehen durch die Materialoberfläche selbst (Kapitel 7.2) wie auch durch den hohen Reflexionsgrad des Materials. Dennoch zeigt sich, dass einzelne Tiefenwerte bzw. Pixel auf dem Material zu finden sind, eine Rekonstruktion von Blechen und metallenen Objekten nicht ausgeschlossen werden kann. Der zu erwartende Aufwand der Fehlerkorrektur und Verbesserung sowie die Genauigkeit der auf wenig rekonstruierten Pixel beruhenden Vermessung (vgl. Abbildung 10-7 rechts) erscheint fraglich.

10.2.3 Structured Light Scanner

Das Projizieren von beispielsweise Streifenlicht- oder Punktmustern in die zu rekonstruierende Szene, um über die Verzerrung der Muster auf Form und Entfernung der Objekte zurückrechnen zu können, ist ein häufiges und seit langem benutztes Prinzip zur Vermessung von Szenen und kleinen sowie großen Objekten. Einzelne Aufnahmen werden mittels Rekonstruktionsalgorithmen zusammengesetzt, die versuchen, aus bewegten Bildern zunächst auf die Kamerabewegung zurückzuschließen und dann die aus den Aufnahmen generierten Punktwolken zusammenzusetzen (Simultaneous Localization and Mapping - SLAM) [5,6]. Abbildung 10-7 (links) zeigt die Ergebnisse der 3D-Rekonstruktion der Palette einer Laserschneidmaschine mit darauf liegenden Blechen zu Testzwecken.

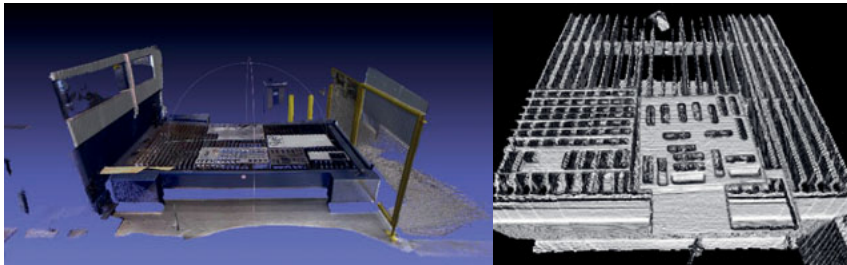


Abbildung 10-7: 3D-Rekonstruktion der Palette einer Laserschneidmaschine bei TRUMPF mittels Structured Light Tiefenkamera (links) und ToF-Kamera (rechts) (Quelle: HLRS)

Mittels schneller GPUs in der verarbeitenden Workstation lassen sich nahezu Echtzeitbedingungen bzw. eine akzeptable Verzögerung von wenigen Sekunden bei der Rekonstruktion erreichen. Deutlich sind auch hier Messrauschen sowie Messfehler aufgrund von unerwünschten Lichtreflexion zu erkennen. Bei langwelliger Strahlung ist die Farbe der Oberfläche weitgehend unbedeutend, Abbildung 10-7 (links) zeigt selbst auf lackierten Oberflächen Reflexionen, und Abbildung 10-8 zeigt die fehlenden Tiefenwerte auf glänzend metallenen Objekten.

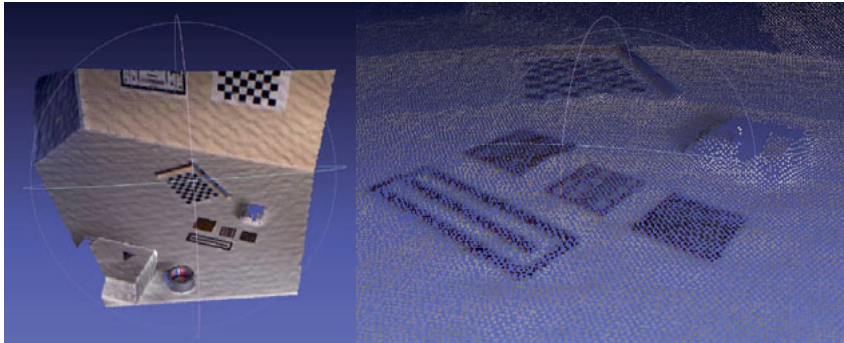


Abbildung 10-8: 3D-Rekonstruktion am HLRS Prototyp mittels Structured Light Tiefenkamera; Kameraausschnitt (links); Detailansicht der Punktwolke (rechts) (Quelle: HLRS)

Structured Light Scanning ist eine gute und etablierte Technik, die auf performanter Hardware nahezu Echtzeitfähigkeit erreicht, die Scanner bzw. Kameras haben aufgrund von Bauweise und Messprinzip je nach Kamera oder Einstellung eine Mindestmessdistanz von 20-80 cm, was deren Einsatz in sehr kleinen Bauräumen von Werkzeugmaschinen erschwert. Diese Scanner bzw. Kameras nutzen die Projektion von Mustern im sichtbaren Licht oder auch, ähnlich wie die ToF-Kameras, im infraroten Lichtfrequenzband.

10.3 Framework für Computer Vision

Die in die Werkzeugmaschinen verbauten Kamerasysteme sollen je nach Anforderung unterschiedlich viele Kameras umfassen. In der SW Maschine im Anwendungsfall bei Bosch Rexroth wurde nur eine Kamera verbaut (vgl. Kapitel 2.5), in der Laserschneidmaschine (vgl. Kapitel 2.3) sind bereits zwei Kameras verbaut worden. Eine Bildverarbeitung (BV) mit zwei Kameras kann für gewöhnlichen noch von einem Rechner bewältigt werden, für den anfallenden Datenstrom bei zwei IP Kameras reicht auch ein normales Gigabit-Ethernet aus (vgl. Kapitel 7.3.2). Beim Vorhaben von TRUMPF, das Verfahren auch auf große Werkzeugmaschinen zu übertragen, deren Bauraum dann mit 5 oder mehr Kameras abgebildet wird oder die Rekonstruktion auf Basis eines Mehrkamerasystems mit 8 oder mehr Kameras, liegt hier die Herausforderung. Einen Bilddatenstrom bei HD-Auflösung mit 15 fps und einer mpeg Kompressionsrate von 1:5 benötigt bereits 150Mbit/s Bandbreite. Man kann sich ausrechnen, dass bei 5 oder mehr Kameras ein Gigabit-Ethernet schnell ausgelastet ist und auch ein einzelner Rechner die Bildverarbeitung von 5 oder mehr HD-Bildströmen nicht mehr bewältigen kann. Zudem kommt die steigende Latenz, die die serielle Verarbeitung der Bilder folgen lässt (Abbildung 10-9).

1 CPU-Kern, 2 Kameras:

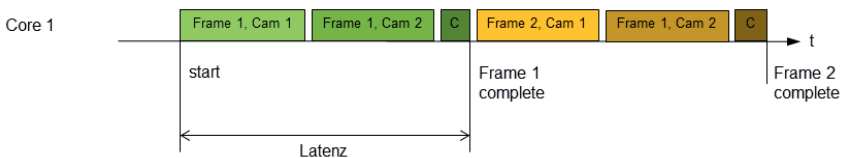


Abbildung 10-9: Sequentielle Bildverarbeitungs-Pipeline; 1 CPU-Kern, 2 Kameras (Quelle: HLRS)

In diesem Beispiel steht zur Veranschaulichung nur ein CPU-Kern zur Verfügung. Dieser führt Bildaufnahme und Bildverarbeitung sequentiell für alle Kameras durch und macht dann ein Compositing (C) aller Bilder beispielsweise zur Generierung eines Übersichtsbildes. Die Latenz, die ein Ergebnisbild von Aufnahme der Einzelbilder bis zur Fertigstellung des Compositing benötigt, ist direkt abhängig von der Anzahl der Kameras.

Die Bildverarbeitung kann auf mehrere Kerne der CPU sowie auf verschiedene CPUs in einem Rechner oder mehreren Rechnern verteilt werden. Die Verarbeitungsschritte bleiben zunächst die gleichen (Abbildung 10-10).

2 CPUs mit je 2 CPU-Kerne, 2 Kameras:

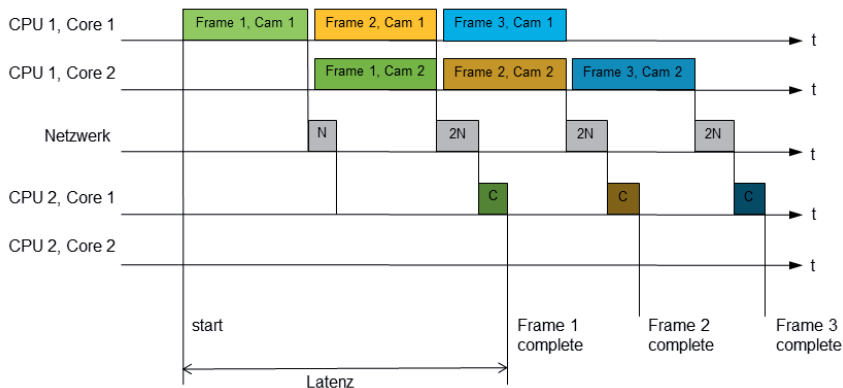


Abbildung 10-10: Verteilte sequentielle Bildverarbeitungs-Pipeline (Ausschnitt); 2x2 CPU-Kerne, 2 Kameras (Quelle: HLRS)

Durch die Verteilung der sequentiellen Schritte auf verschiedene Rechner, kann in diesem Beispiel der erste Rechner die Bildverarbeitung auf mehrere CPU-Kerne verteilen, während zum Beispiel ein zweiter Rechner das Compositing berechnet. Die Latenz der Einzelergebnisse steigt geringfügig, da nun auch die Netzwerklatenz miteinbezogen werden muss. Die mögliche Bildwiederholrate bzw. der Bilddurchsatz steigt dagegen deutlich an, da die Bildaufnahme und Verarbeitung parallel erfolgen kann.

Es sei erwähnt, dass die Bildaufnahme und Verarbeitung nicht beliebig verteilt werden können, da beispielsweise USB-Kameras ohne weiteres nur von dem Rechner ausgelesen werden können, an den sie angeschlossen sind. Der Ausleseprozess eines Rechners und einer Kamera ist tatsächlich auch der einzige Prozess, der sich nicht verteilen lässt. Davon ausgehend, werden nun auch die Verarbeitungsschritte aus Abbildung 10-10 weiter unterteilt (Abbildung 10-11).

In diesem Beispiel sei das Videocapturing (V), das Processing (P), der Detector (D), ein RANSAC-Filter (R) und das Compositing (C) in die jeweiligen diskreten Prozesse zerlegt, die über eine Shared-Memory-Architektur Bilddaten miteinander austauschen. So ist es möglich, nur solche Prozesse auf dem Rechner zu belassen, die die tatsächliche kameraspezifische Verarbeitung übernehmen und andere Prozesse im Rechnersystem zu verteilen. Zudem kann der Rechner, an dem die Kamera angeschlossen ist, sämtliche kameraspezifische Verarbeitung übernehmen und diese zur Weiterverarbeitung an entsprechend andere Rechner mit anderen Aufgaben übertragen. Trotz zuzüglicher Netzwerklatenz kann sich die Verarbeitungslatenz eines Frames deutlich reduzieren, da sich die diskreten Einzelprozesse nun parallelisieren lassen. Dieser Ansatz ermöglicht in Abhängigkeit von den durchzuführenden Schritten der Bildverarbeitung eine Verbesserung der Bildrate sowie der Einzelbildlatenz und erlaubt zudem die Verteilung der aufgabenspezifischen Prozesse auf dedizierte Rechner, wie zum Beispiel

Embedded-Systeme für die Bildaufnahme, wie sie heute oft in Industrie- oder IP-Kameras zum Einsatz kommen.

2 CPUs mit je 2 CPU-Kerne, 2 Kameras:

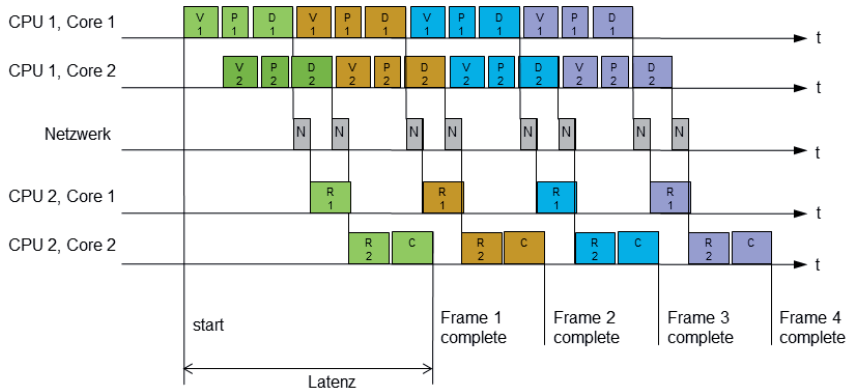


Abbildung 10-11: Parallele Bildverarbeitungspipeline; 2x2 CPU-Kerne, 2 Kameras (Quelle: HLRS)

Abstrahiert man den eigentlichen Bildverarbeitungsprozess, so lassen sich nach diesem Prinzip beliebige Bildverarbeitungspipelines auf ein verteiltes Rechnersystem abbilden (Abbildung 10-12).

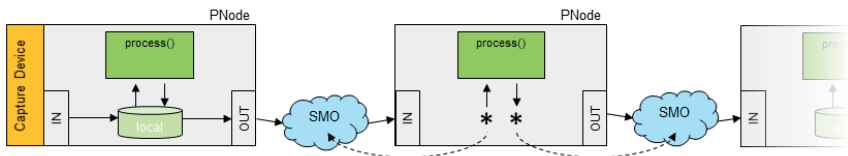


Abbildung 10-12: Allgemein Bildverarbeitungspipeline (Quelle: HLRS)

Die einzelnen Prozesse arbeiten dabei je nach implementierten BV-Algorithmus auf einem lokalen Speicherbereich oder direkt in ein Shared-Memory-Objekt (SMO), das nach Durchlauf des Prozesses an den nächsten Prozess weitergegeben werden kann. Das Signaling kann dabei über die Shared-Memory-Objekte oder über Sockets erfolgen.

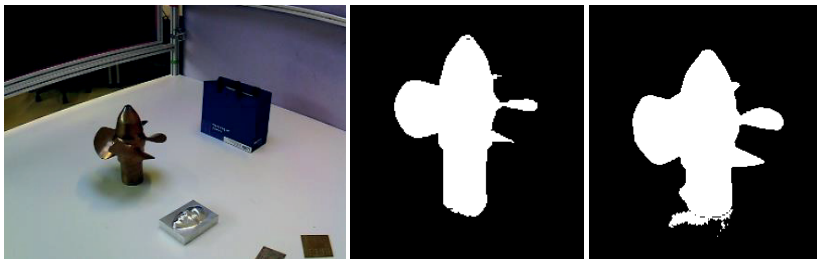
Dieser Ansatz erlaubt das quasi beliebige Zusammenfügen einer Bildverarbeitungspipeline und deren Verteilung auf ein Rechnernetzwerk entsprechend der angeschlossenen Kameras, Ressourcen und der zur Verfügung stehenden Rechnerleistung. Beispielsweise können die Rechner, die mit den Kameras direkt verbunden sind, die Entzerrung der Bilder und eine Vorverarbeitung übernehmen, um so den Rechner, der die Bilder zusammenfügt, zu entlasten. Außerdem kann durch Senden der vorverarbeiteten Daten, anstatt der rohen Bilddaten, die Netzauslastung und -latenz deutlich reduziert werden. Ist an der Werkzeugmaschine eine

erweiterte Bildverarbeitung gewünscht, kann diese optional als zusätzliches Rechnermodul zuzüglich den Kameras, die eventuell auch schon vorhanden sind, verbaut werden, ohne dass der Rechner zur Maschinensteuerung mit zusätzlichen Aufgaben belastet wird.

10.4 3D-Rekonstruktion in der Werkzeugmaschine

Für die Entwicklung eines robusten und in nahe Echtzeit arbeitendem 3D-Rekonstruktionsverfahren, zeigten die bisherigen Ansätze deutliche Nachteile hinsichtlich der Auswertungsmöglichkeit bzw. der Messfehler, die auf der Schwierigkeit beruhen, metallene Objekte zu erkennen und Reflexionen darin ausschließen zu können. Auch die Geschwindigkeit, insbesondere von photogrammetrischer Verfahren, die eine wesentlich höheren Zeitbedarf in der Bildfusion und Rekonstruktion aufweisen, ist in Bezug auf die genannten Anforderungen zu langsam.

Ein Bildaufnahmeverfahren, das auf einem Bilddifferenzverfahren und einer Konturerkennung basiert (vgl. Kapitel 3.2), verspricht wesentlich robuster und performanter zu sein. Mit Hilfe eines dynamischen Schwellwertverfahren und gleitender Bewegungsdetektion, können Bewegungen bzw. Veränderungen im Arbeitsraum der Werkzeugmaschine schnell und zuverlässig erkannt werden. Das Differenzbild bezüglich der letzten Aufnahme vor dieser Detektion enthält die Kontur des bewegten Objekts (Abbildung 10-13).



**Abbildung 10-13: Original Kamerabild und zwei Differenzbilder im Prototypen
(Quelle: HLRS)**

Zur Rekonstruktion wird die Technik Shape from Silhouette (SFS) genutzt, die die von den Kameras sichtbare Hülle (visual hull) eines Körpers berechnet. Diese stellt den größtmöglichen, diskretisierten Körper dar, der entsprechend der Kameraperspektiven die durch diese gegebenen Silhouetten nachbilden kann. Der Volumenbasierte Ansatz, das volume oder space carving [7], nimmt zunächst den zu erfassenden Raum als einen diskreten Raum an, der mit Volumenelementen (voxel) vollständig gefüllt ist. Überall dort, wo die Kamera kein Objekt wahrnimmt, also außerhalb der erkannten Kontur, werden die Voxel entfernt. Unter Verwendung genügend vieler unterschiedlicher Perspektiven kann so das Objekt entsprechend fein aufgelöst werden [8].

Jedoch ist auch bei diesem Verfahren, die Qualität der Rekonstruktion direkt abhängig von der Anzahl und Positionen der Perspektiven, die man vom Arbeitsraum erhält. Eine ungünstige Orientierung des zu erfassenden Objekts bei zu wenig Perspektiven kann zu entsprechend starken Verzerrungen des Objekts führen (Abbildung 10-14).

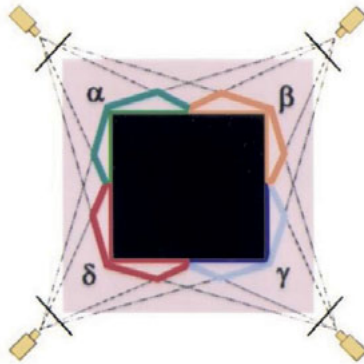


Abbildung 10-14: Voxel Carving / Space Carving Prinzip (Quelle: [7])

Die Rekonstruktion per SFS zeigt sich als sehr robust gegenüber fehlerhaft erkannten Konturen im Bild bzw. Pixeln, die in der Maske erscheinen, aber nicht vom Objekt stammen. Sind diese lediglich in einzelnen Bildern vorhanden, werden sie durch die Überdeckung der Sichtkegel der Kameras durch die Konturrekonstruktion einer anderen Kamera entfernt. Außerdem ist das Verfahren sehr robust gegenüber Lichtreflexionen oder sogar Spiegelungen im Objekt. Materialoberflächen spielen keine Rolle, da das Differenzbild lediglich die Information enthält, wo sich das Objekt befindet bzw. nicht befindet. Diese Information ist zudem reduziert auf eine Maske bzw. ein S/W-Bild, was die erforderliche Netzwerkbandbreite im verteilten Kamera- und Bildverarbeitungssystem deutlich reduziert (vgl. Kapitel 10.3). Das Verfahren ist schnell und eine Portierung auf GPUs bereits verfügbar.

Als sehr interessante Eigenschaft erscheint die Skalierbarkeit des Systems. Je mehr unterschiedliche Kameraperspektiven zur Verfügung stehen, desto feiner können die Objekte aufgelöst werden. Das heißt, ist eine Stelle im Arbeitsraum von besonderer Bedeutung, muss diese zum Beispiel höher aufgelöst werden als andere Stellen, so können hier entsprechend mehr Kameras angebracht werden.

10.5 Literaturverzeichnis

- [1] WithRobot – oCamS-1CGN-U, USB Stereokamera
<http://withrobot.com/en/camera/ocams-1cgn-u/>
 (01. Feb. 2018)
- [2] BlueTechnix ToF-Kameras und Sensormodule
<https://www.bluetechnix.com/de/products/tof-3d-cameras/>
 (01. Feb 2018)
- [3] NeoMatrix Technologies, Inc
<http://3dscanningservices.net/evixscan-3d-scanners/>
 (01. Feb. 2018)

-
- [4] Frede, M. & Moalem, A. (2012). *Neue Prozesssicherheit durch Vorpulslaser*. SMM - Schweizer Maschinenmarkt 4 (113), ff. 72–75, Februar 2012.
 - [5] Whelan, T. et al. (2014). *Real-time Large Scale Dense RGB-D SLAM with Volumetric Fusion*. In: The International Journal of Robotics Research 2014 (IJRR '14).
 - [6] Whelan, T. et al. (2015). *ElasticFusion: Dense SLAM Without A Pose Graph*. In: Proceedings of Robotics: Science and Systems XI, July 13 - July 17, 2015, Rome, Italy.
 - [7] Kutulakos, K. & Seitz, S. (2000). *A Theory of Shape by Space Carving*. In: International Journal of Computer Vision 38(3), pp 199–218, 2000.
 - [8] Liu, X. & Gavrilova, M. (2009) *High Quality Visual Hull Reconstruction by Delaunay Refinement*. In: Proceedings of the Sixth International Symposium on Voronoi Diagrams (ISVD), 23-26 Juni, 2009, Copenhagen, Denmark.

Online-Buchshop für Ingenieure

■ ■ VDI nachrichten

BUCHSHOP

Online-Shops



**Fachliteratur und mehr -
jetzt bequem online recher-
chieren & bestellen unter:
www.vdi-nachrichten.com/
Der-Shop-im-Ueberblick**



**Täglich aktualisiert:
Neuerscheinungen
VDI-Schriftenreihen**



Im Buchshop von vdi-nachrichten.com finden Ingenieure und Techniker ein speziell auf sie zugeschnittenes, umfassendes Literaturangebot.

Mit der komfortablen Schnellsuche werden Sie in den VDI-Schriftenreihen und im Verzeichnis lieferbarer Bücher unter 1.000.000 Titeln garantiert fündig.

Im Buchshop stehen für Sie bereit:

VDI-Berichte und die Reihe **Kunststofftechnik**:

Berichte nationaler und internationaler technischer Fachtagungen der VDI-Fachgliederungen

Fortschritt-Berichte VDI:

Dissertationen, Habilitationen und Forschungsberichte aus sämtlichen ingenieurwissenschaftlichen Fachrichtungen

Newsletter „Neuerscheinungen“:

Kostenfreie Infos zu aktuellen Titeln der VDI-Schriftenreihen bequem per E-Mail

Autoren-Service:

Umfassende Betreuung bei der Veröffentlichung Ihrer Arbeit in der Reihe Fortschritt-Berichte VDI

Buch- und Medien-Service:

Beschaffung aller am Markt verfügbaren Zeitschriften, Zeitungen, Fortsetzungsreihen, Handbücher, Technische Regelwerke, elektronische Medien und vieles mehr – einzeln oder im Abo und mit weltweitem Lieferservice

VDI nachrichten

BUCHSHOP

www.vdi-nachrichten.com/Der-Shop-im-Ueberblick

Die Reihen der Fortschritt-Berichte VDI:

- 1 Konstruktionstechnik/Maschinenelemente
 - 2 Fertigungstechnik
 - 3 Verfahrenstechnik
 - 4 Bauingenieurwesen
- 5 Grund- und Werkstoffe/Kunststoffe
 - 6 Energietechnik
 - 7 Strömungstechnik
- 8 Mess-, Steuerungs- und Regelungstechnik
 - 9 Elektronik/Mikro- und Nanotechnik
 - 10 Informatik/Kommunikation
 - 11 Schwingungstechnik
- 12 Verkehrstechnik/Fahrzeugtechnik
 - 13 Fördertechnik/Logistik
- 14 Landtechnik/Lebensmitteltechnik
 - 15 Umwelttechnik
 - 16 Technik und Wirtschaft
- 17 Biotechnik/Medizintechnik
- 18 Mechanik/Bruchmechanik
- 19 Wärmetechnik/Kältetechnik
- 20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
 - 21 Elektrotechnik
 - 22 Mensch-Maschine-Systeme
- 23 Technische Gebäudeausrüstung

ISBN 978-3-18-369702-1