Jürgen Gausemeier
Michael Grafe
Friedhelm Meyer auf der Heide (Hrsg.)

Augmented & Virtual Reality in der Produktentstehung

- Grundlagen, Methoden und Werkzeuge
- Interaktions- und Visualisierungstechniken
- Virtual Prototyping intelligenter technischer Systeme mit AR/VR

12. Paderborner Workshop
Augmented & Virtual Reality in der Produktentstehung
23. und 24. April 2015
Heinz Nixdorf MuseumsForum, Paderborn

Bibliografische Information Der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar

©Heinz Nixdorf Institut, Universität Paderborn – Paderborn – 2015

ISSN 2195-5239

Das Werk einschließlich seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung der Herausgeber und des Verfassers unzulässig und strafbar. Das gilt insbesondere für Vervielfältigung, Übersetzungen, Mikroverfilmungen, sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Als elektronische Version frei verfügbar über die Digitalen Sammlungen der Universitätsbibliothek Paderborn.

Satz und Gestaltung: Polina Decheva und Franziska Reichelt

Hersteller: Hans Gieselmann Druck und Medienhaus GmbH & Co. KG
Druck · Buch · Verlag
Bielefeld

Printed in Germany

Vorwort

Augmented Reality und Virtual Reality eröffnen neue Perspektiven: Virtuelle funktionale Prototypen ermöglichen die immersive und interaktive Präsentation von Gestalt und Verhalten eines zukünftigen Produkts. Das verbessert die Kommunikation in Entwicklungsprozessen, hilft Entwicklungsfehler zu vermeiden und trägt dazu bei, die Anzahl realer Prototypen zu verringern. Unternehmen sparen so Zeit und Geld.

Das Heinz Nixdorf Institut der Universität Paderborn veranstaltet zweijährlich unter dem Dach „Wissenschafts- und Industrieforum – Intelligente Technische Systeme“ den Workshop „Augmented & Virtual Reality in der Produktentstehung“. Wir richten uns an Fachleute aus Industrie und Forschungsinstituten, die sich maßgeblich mit der Entwicklung von intelligenten technischen Systemen befassen. Die Veranstaltung bietet ihnen eine Plattform zur Diskussion und zum Erfahrungsaustausch.

Für die Begutachtung und Auswahl der Beiträge danken wir den Mitgliedern des Programmkomitees herzlich. Deren Expertise hat wesentlich zur hohen Qualität der Tagung beigetragen.

Prof. Dr.-Ing. M. Abramovici,
Universität Bochum

Prof. Dr.-Ing. R. Anderl,
TU Darmstadt

Prof. Dr. B. Brüderlin,
TU Ilmenau

Prof. Dr. G. Domik,
Universität Paderborn

Prof. Dr. Ch. Geiger,
Fachhochschule Düsseldorf

Prof. Dr. O. Herzog,
Universität Bremen

Prof. Dr.-Ing. habil. G. Höhne,
TU Ilmenau

Prof. Dr.-Ing. R. Koch,
Universität Paderborn

Prof. Dr.-Ing. S. Kopp,
Universität Bielefeld

Prof. Dr.-Ing. F.-L. Krause,
TU Berlin

Prof. Dr. rer. nat. T. Kuhlen,
RWTH Aachen

Prof. Dr.-Ing. U. Lindemann,
TU München

Prof. Dr.-Ing. habil. P. Nyhuis,
Universität Hannover

Prof. Dr.-Ing. J. Ovtcharova,
TU Karlsruhe

Prof. Dr.-Ing. G. Reinhart,
TU München

Prof. Dr.-Ing. J. Roßmann,
RWTH Aachen

Prof. Dr.-Ing. habil. Dr.-Ing. E. h. M. Schenk,
Fraunhofer IFF

Prof. Dr.-Ing. R. Stark,
TU Berlin

Prof. Dr.-Ir. F.J.A.M. van Houten,
University of Twente

Dr.-Ing. H. Waßmann-Kahl,
VW AG

Prof. Dr.-Ing. Ch. Weber,
TU Ilmenau

Paderborn, im April 2015

Prof. Dr.-Ing. J. Gausemeier, M. Grafe und
Prof. Dr. math. F. Meyer auf der Heide

Inhaltsverzeichnis

VR-Systeme und Anwendungen

M. Oberhauser, D. Dreyer, P. Kovačič
Rapid Prototyping in einem funktionalen, virtuellen Flugzeugcockpit 9

S. Husung, A. Siegel, C. Weber, B. Müller, M. Behrendt, D. Landes, A. Albers
Audio-visuelle Simulation der Fahrzeugvorbeifahrt unter Nutzung von VR und Prüfstandsmessungen 21

W. Steger, E. Steindecker, R. Stelzer, M. Gebert, T. Kim
Individuelle Funktionen für Teilnehmer an einem VR Design Review 33

AR/VR in der Produktentstehung

V. Paelke, S. Heymann
Benutzungsschnittstellen für Cyber-Physical-Systems – Herausforderungen und Lösungsansätze 49

J. Berssenbrügge, O. Wiederkehr, C. Jähn, M. Fischer
Anbindung des Virtuellen Prototypen an die Partialmodelle intelligenter technischer Systeme 65

K. Wrasse, E. Brandenburg, H. Hayka, R. Stark
Visualisierung von Anforderungen an physische Prototypen mit Augmented Reality 79

Rendering und 3D-Produktdarstellung

J. Meder
Depth Image Based Rendering zur Entkopplung langsamer Renderprozesse von der Darstellung 93

C. Jähn, M. Fischer, M. Gerges, J. Berssenbrügge
Automatische Ableitung geometrischer Eigenschaften von Bauteilen aus dem 3-D-Polygonmodell 107

T. Preidel, S. Cersowsky, R. Krüger, M. Färber
Punktebasiertes hierarchisch-probabilistisches Level of Detail 121

VR für Fabrikplanungsprozesse

S. Pick, S. Gebhardt, B. Hentschel, T. Kuhlen, R. Reinhard, C. Büscher, T. al Khawli, U. Eppelt, H. Voet, J. Utsch
Ein Konzept zur Integration von Virtual Reality Anwendungen zur Verbesserung des Informationsaustauschs im Fabrikplanungsprozess 139

S. Gebhardt, S. Pick, B. Hentschel, T. Kuhlen, R. Reinhard, C. Büscher
Ein Ansatz zur Softwaretechnischen Integration von Virtual Reality Anwendungen am Beispiel des Fabrikplanungsprozesses 153

R. Langmann, C. Geiger
Web-basierte Integrationsplattform für dynamische 3D-Anlagenvisualisierung 167

AR für mobile Assistenzsysteme

M. Neges, M. Wolf, M. Abramovici
Zugriffsgesicherte Augmented Reality Lösung zur mobilen Instandhaltungsunterstützung mit zustandsorientierten Arbeitsanweisungen 181

S. Adler, A. Kernchen, T. Reipsch, E. Bayrhammer, U. Schmucker
Mobile Assistenzsysteme für sicheren Betrieb und Wartung von Maschinen und Anlagen 197

Fahrsimulation und Fahrerassistenzsysteme

B. Hassan, J. Gausemeier, K. Abdelgawad, J. Berssenbrügge, M. Grafe
Systematik zur Entwicklung von rekonfigurierbaren Fahrsimulatoren 213

J. Berssenbrügge, A. Trächtler, C. Schmidt
Visualisierung von Scheinwerfersystemen für das Virtual Prototyping von lichtbasierten Fahrerassistenzsystemen 231

H. Amelunxen
Entwicklung eines Closed-Loop Testsystems für fortgeschrittene Fahrerassistenzsysteme 247

D. Zimmermann, A. Kohlstedt, S. Gausemeier, A. Trächtler
Entwicklung eines prädiktiven Motion-Cueing-Verfahrens für den ATMOS-Fahrsimulator 261

VR-Systeme und Anwendungen

Rapid Prototyping in einem funktionalen, virtuellen Flugzeugcockpit

M.Sc. Matthias Oberhauser, Dr-Ing Daniel Dreyer
Airbus Group Innovations
Willy-Messerschmitt-Straße 1, 81663 München
Tel. +49 (0) 89 607 / 21 236, Fax. +49 (0) 89 607 / 21 240
E-Mail: matthias.oberhauser@airbus.com

Primož Kovačič
Technische Universität Graz
Rechbauerstraße 12, 8010 Graz
E-Mail: primoz.kovacic@student.tugraz.at

Zusammenfassung

Beim Entwicklungsprozess von Cockpits und Cockpitkomponenten sollten in einer möglichst frühen Phase Nutzerstudien durchgeführt werden, um sowohl ergonomische als auch kognitiv-psychologische Aspekte in der Mensch-Maschine-Interaktion zu berücksichtigen. Für Ergonomieuntersuchungen werden bereits auf digitale Mock-ups und Virtual Reality Systeme zurückgegriffen. Um das Verhalten des Nutzers auch in einer dynamischen Cockpitumgebung untersuchen zu können, wurde der Virtual Reality Flight Simulator (VRFS) entwickelt. Hier wird ein Virtual Reality Cockpit mit einem Flugsimulator verbunden. Im Gegensatz zu ähnlichen VR-Flugsimulatoren liegt bei diesem System der Fokus auf Rapid Prototyping von Cockpitkomponenten und deren Evaluation mit integrierten kognitiv-psychologischen Methoden. In dieser Arbeit wird gezeigt, wie neue Cockpitkomponenten in diesen Simulator integriert werden können und wie mit diesen Elementen interagiert werden kann. Als Basis für die Geometrie dienen hierbei CAD-Daten oder auch 3D-Scans. Die Interaktion kann mit virtuellen Tastern, Hebeln, Drehschaltern oder über simulierte Touchscreens erfolgen. Durch die Integration von einfachen Platten, 3D-gedruckten Bauteilen oder sogar funktionalen Hardwarekomponenten an der entsprechenden räumlichen Position, kann haptisches Feedback erzeugt und die Bedienbarkeit deutlich verbessert werden. Durch diese Flexibilität kann der virtuelle Flugsimulator in jeder Phase des Cockpitdesignprozesses eingesetzt werden.

Schlüsselworte

Virtual Reality, Flugsimulation, Cockpit Design, Rapid Prototyping

Rapid Prototyping in a Functional Virtual Cockpit

Abstract

In the development process of flight decks, user studies should be conducted as early as possible to gather information on ergonomics as well as cognitive aspects of the human-machine-interaction (HMI). Virtual Reality is a well-known tool for exploring the ergonomics of a cockpit. Yet, for conducting human factors studies, a dynamic, usable flight deck is necessary. The Virtual Reality Flight Simulator (VRFS), developed by Airbus Group Innovations, provides this functionality by connecting a digital mock-up with a commercially available flight simulation. In contrast to similar VR flight simulations, human factors methods have been integrated to conduct user studies. This research presents the integration of conceptional as well as existing HMI components into the VRFS. CAD data or 3D scans can be used as a basis for this. The interaction with these elements can be fully virtual, aided by using plates, or combined with functional hardware elements. Due to this flexibility, the VRFS can be used in all phases of the cockpit development process.

Keywords

Virtual Reality, Flight Simulation, Cockpit Design, Rapid Prototyping

1 Einführung

Die technischen Systeme in modernen Flugzeugen sind komplex, miteinander vernetzt und zum Teil hochgradig automatisiert. Um diese Komplexität für einen Piloten beherrschbar zu machen, müssen klare und intuitive Mensch-Maschine-Schnittstellen (MMS) bereitgestellt werden. Der Entwicklungsprozess von Cockpits und Cockpitkomponenten sollte deshalb bereits in einer möglichst frühen Phase durch Nutzerstudien unterstützt werden, um ergonomische und kognitive Aspekte in der Mensch-Maschine-Interaktion zu berücksichtigen. Außerdem können nötige Modifikationen dadurch rechtzeitig erkannt, und mit wenig Aufwand bzw. niedrigen Kosten umgesetzt werden [RN04].

Für diese Studien bieten sich verschiedenste Prototypen an, die von der Simulation einzelner Komponenten bis hin zu komplexen Full-Flight-Simulatoren reichen. Die Entwicklung und der Bau solcher Cockpit-Mock-ups ist mit einem erheblichen Aufwand verbunden. Dasselbe gilt für Änderungen an einem solchen Hardware System. Deshalb haben sich Virtual Reality Systeme im Cockpitentwicklungsprozess für die Evaluation ergonomischer Aspekte etabliert. Diese virtuellen Systeme bilden die räumliche Anordnung des Cockpits ab und es können Erreichbarkeiten und die Außensicht getestet werden. Um auch kognitive Aspekte bei der Interaktion mit dem Cockpit untersuchen zu können, wurde bei Airbus Group Innovations der Virtual Reality Flight Simulator (VRFS) entwickelt. Hier wird ein Virtual Reality Cockpit mit einem Flugsimulator verbunden und es können realitätsnahe Szenarien simuliert werden. Ein ähnliches System wurde an der TU Darmstadt für das Training von Piloten entwickelt [HK05]. Das Enhanced Virtual Environment (EVE) von Airbus Helicopter [Bau14-ol] wird unter anderem für die Flugunfalluntersuchung verwendet und ein weiterer virtueller Helikopter Simulator wurde von der Universität Istanbul vorgestellt [YKT+09]. Im Gegensatz zu diesen VR-Flugsimulatoren liegt bei dem hier vorgestellten System der Fokus auf Rapid Prototyping von Cockpit Elementen und deren Evaluation mit integrierten kognitiv-psychologischen Methoden.

2 Der Virtual Reality Flight Simulator

2.1 Grundlagen

Der Virtual Reality Flight Simulator basiert auf einer kommerziell verfügbaren Flugsimulationssoftware, die durch Virtual Reality Fähigkeiten ergänzt wurde. Der Kern dieses Systems besteht aus einem optischen Tracking System, das die Position des Kopfes und der Finger des Nutzers erfasst (siehe Bild 1). Diese Koordinaten werden transformiert und an die grafische Ausgabe der Simulation weitergegeben. Diese Ausgabe wird dann an ein Head-Mounted-Display (HMD) übertragen. Da in der Flugsimulationssoft-

ware ein 3D-Cockpit modelliert ist, bekommt der Nutzer somit einen immersiven Eindruck der Cockpitumgebung [ADP+14]. Die Hand des Nutzers wird in der Simulation durch eine vereinfachte 3D-Geometrie dargestellt. Durch eine Kollisionsberechnung der Hand- bzw. Fingerkoordinaten mit der Cockpitgeometrie kann mit dem Cockpit - und somit mit der Simulation - interagiert werden.

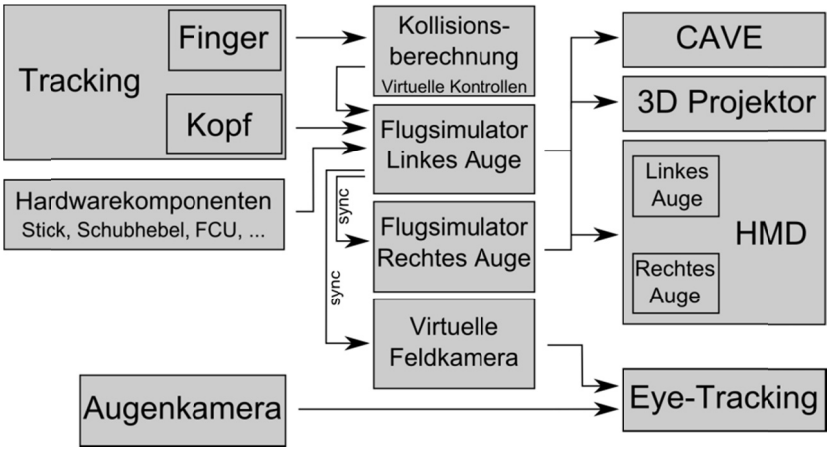


Bild 1: Die Architektur des VRFS nach Dreyer et al. [DBO14]

Das System besteht aus zwei Computern auf denen je eine Instanz der Flugsimulatorsoftware läuft, während ein separater Computer Tracking Daten empfängt und verarbeitet. Bei der Verwendung von Eye Tracking kann eine weitere synchronisierte Instanz der Flugsimulatorsoftware verwendet werden.

2.2 Interaktionsmechanismen

Der Virtual Reality Flight Simulator bietet verschiedenste Möglichkeiten zur Interaktion (siehe Bild 2). Diese reichen von Bedienelementen die vollvirtuell dargestellt und bedient werden, über einfaches haptisches Feedback bis zur Integration echter Hardwareelemente, dem sogenannten „Mixed Mock-up“.

2.2.1 Vollvirtuelle Bedienelemente

Einfache Taster und Schalter (Push Buttons) werden mithilfe der Kollisionsberechnung bedient. Hierbei wird ein einzelnes Signal an die Simulation gesendet sobald eine Fingergeometrie mit einer entsprechenden Geometrie eines Kollisionsvolumens kollidiert. Um Doppel-Aktivierungen zu vermeiden, ist jeder Taster/Schalter mit einer kurzen Totzeit ausgestattet [HK05].

Die vollvirtuelle Bedienung von Hebeln ist ebenfalls möglich. Hierbei ist es nötig, in das Kollisionsvolumen einzudringen. Ab diesem Moment ist der virtuelle Hebel mit dem jeweiligen Finger verknüpft und er kann nach vorne oder nach hinten verschoben

werden. Bei Verlassen des Kollisionsvolumens in Richtung der Z-Achse bleibt der Hebel in der jeweiligen Stellung stehen oder springt in eine definierte Rasterstellung.

Die vollvirtuelle Bedienung von Drehschaltern funktioniert ähnlich. Nach der Kollision mit einem Drehschalter ist die Rotation mit der Fingerkoordinate verknüpft. Dies wird zusätzlich durch die Einblendung eines Kreissegmentes signalisiert. Durch das Zurückziehen des Fingers in Richtung der Hochachse des Drehschalters wird diese Verknüpfung wieder gelöst und der Schalter verbleibt in der gewählten Stellung.



Bild 2: Interaktionsmechanismen mit dem digitalen Mock-up: L.o.: Virtueller Push Button, r.o.: Flight Control Unit im Mixed Mock-up, l. u.: Virtueller Drehschalter, r.u.: Virtueller Hebel.

2.2.2 Das Mixed Mock-up

Bei vollvirtuellen Bedienelementen fehlt das haptische Feedback. Dadurch ist die Bedienbarkeit dieser Elemente eingeschränkt und der Nutzer benötigt mehr Zeit für die Aktivierung. Zusätzlich sind viele Bedienelemente im Cockpit nicht nur optisch gekennzeichnet, sondern unterscheiden sich auch in ihrer Haptik, um Fehlbedienungen zu vermeiden. Um dieses wichtige zusätzliche Feedback zu erzeugen, kann ein Hardwareelement an die exakte räumliche Position entsprechend des digitalen Mock-ups verbaut werden. Dadurch entsteht das sogenannte „Mixed Mock-up“. Im einfachsten Fall sind diese Hardwareelemente Platten, die haptisches Feedback erzeugen. Die Aktivierung des Elements erfolgt dabei weiterhin virtuell über die Kollisionsberechnung. In einem komplexeren Fall können auch funktionale Hardwareelemente verbaut werden. Diese sind in der Regel direkt an die Flugsimulation angeschlossen und bieten eine leichte und intuitive Bedienung.

2.2.3 Integration von Touchscreens

Touchscreens können ebenfalls vollvirtuell emuliert werden. In einem entsprechend großen Kollisionsvolumen können X- und Y-Koordinaten der eingedrungenen Fingergeometrie berechnet und weiterverarbeitet werden.

Um die Bedienbarkeit zu erhöhen, kann aber auch hier das „Mixed Mock-up“ Konzept verwendet werden indem ein touchfähiges Gerät an der entsprechenden räumlichen Position in den Simulator eingebunden wird. Dieses Gerät kann nun die X- und Y-Koordinaten und eventuell Informationen über Gesten an die Simulation senden oder direkt selbst verarbeiten und interpretieren.

2.3 Human Factors Methoden

Um den Virtual Reality Flight Simulator für Nutzerstudien verwenden zu können, wurden mehrere Human Factors Methoden integriert [HOD+14]. In der Human Factors Forschung wie auch in der Marktforschung sind Eye-Tracking Systeme weit verbreitet (siehe Bild 3). Besonders für die Evaluation moderner Glass-Cockpit Systeme oder Head-Up Displays sind Informationen über die Blickbewegung unerlässlich. Deshalb wurde in den Virtual Reality Flight Simulator ebenfalls ein Eye-Tracking System integriert [LIE13].



Bild 3: Eye Tracking Ansichten v.l.n.r.: AR Marker, Areas of Interest und Heatmap

Eine besondere Schwierigkeit stellt hierbei das Head-Mounted Display dar. Optische Eye Tracking Systeme benötigen eine freie Sicht auf mindestens ein Auge des Nutzers. Deshalb muss die Eye-Tracking Kamera in das HMD integriert werden. Dazu wird diese an einem modularen Stecksystem an einem Ring um die Optik des HMDs angebracht und kann somit an die Physiologie verschiedenster Probanden angepasst werden.

Neben Eye-Tracking Analysen können auch andere, in der Human Factors Forschung gebräuchliche Methoden verwendet werden. Dazu zählen Fragebögen wie der NASA Task Load Index (NASA-TLX) [HS88] zur Erfassung der mentalen Beanspruchung oder die Situation Awareness Rating Technique (SART) [ST90] zur Erfassung des Situationsbewusstseins. Diese Fragebögen können schriftlich außerhalb der Virtual Reality

Umgebung, beispielsweise nach einem Experiment bearbeitet oder direkt in die VR eingeblendet werden.

Eine weitere Methode zur Erfassung des Situationsbewusstseins ist die Situation Awareness Global Assessment Technique (SAGAT) [End95]. Hierbei wird die Simulation an einem bestimmten Punkt angehalten und Elemente werden aus dem digitalen Mockup ausgeblendet. Der Nutzer wird nun über aktuelle Systemparameter wie Geschwindigkeit oder Höhe befragt [MDO14].

3 Produktintegration und Rapid Prototyping

3.1 Pipeline für die Integration von 3D-Komponenten

Als Basis für 3D-Komponenten dienen CAD-Daten. Somit kann diese Geometrie nicht nur für die Evaluation in Virtual Reality verwendet werden, sondern auch für die weitere Entwicklung und Fertigung. Um die 3D-Geometrie möglichst realistisch darzustellen, muss diese mit Texturen versehen werden, die, um Ressourcen zu schonen, bereits vorschattiert sein sollten (siehe Bild 4).

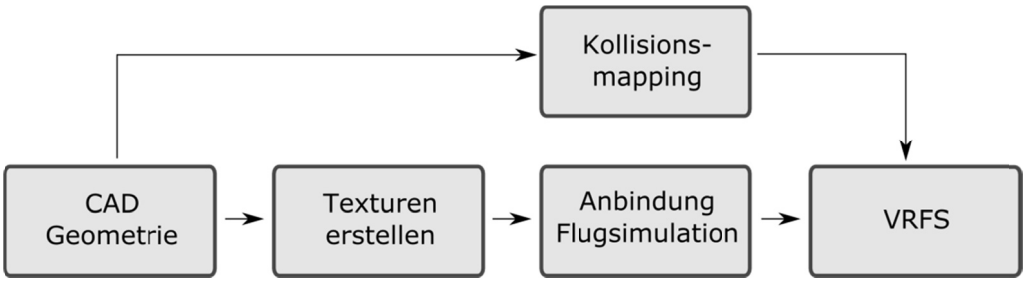


Bild 4: Arbeitsschritte für die Integration einer 3D-Komponente

Diese texturierte Geometrie muss nun noch mit der Flugsimulation verbunden werden. Dazu werden Elemente wie Taster oder Hebel mit den Simulationsdaten verknüpft, um Animationen wie Translation oder Rotationen zu erzeugen. Die Änderung der Simulationsdaten eines Elements wirkt sich somit in einer entsprechenden Änderung in der Position der Geometrie aus. Neben diesen Animationen im Cockpit sind auch Displayinhalte von großer Bedeutung. Diese müssen, entweder durch die Flugsimulation selbst, oder durch ein externes Modul erzeugt und einer Geometrie im Cockpit zugeordnet werden. Dies führt zu einem Cockpit, das durch Hardwareelemente, die an den Flugsimulator angeschlossen sind und somit die Simulationsdaten verändern, bedient werden kann. Vollvirtuelle Elemente müssen über ein Modul zur Kollisionserkennung eingebunden werden. Dazu wird die Geometrie verwendet. Die einzelnen vollvirtuellen Bedienelemente werden mit einem einfachen Hüllkörper umgeben, der für die Kollisionsberechnung verwendet wird. Dieser Hüllkörper wird mit einem Identifier verknüpft, der als Signal an die Flugsimulation gesendet wird.

3.2 Integration existierender MMS-Komponenten

Für die Integration bestehender Komponenten in den virtuellen Flugsimulator müssen diese, sollten keine Geometriedaten vorhanden sein, nachmodelliert werden. Eine einfache Möglichkeit ist die Zuhilfenahme von 3D-Scans. Eine simple und ausreichend genaue Methode hierfür bietet zum Beispiel die Benutzung einer Kinect Kamera.



Bild 5: Nachmodellierung eines Sitzes mit überlagertem 3D-Scan

Die Punktwolke eines 3D-Scans kann als Basis für die CAD-Modellierung verwendet werden, wobei in vielen Fällen keine detailgetreue CAD-Nachbildung gefragt ist. Vielmehr ist es für manche Elemente, wie zum Beispiel für den Sitz in Bild 5 wichtig, nur markante Maße, wie die Sitzfläche, die Lehnen und die Oberkante der Kopfstütze exakt zu modellieren.

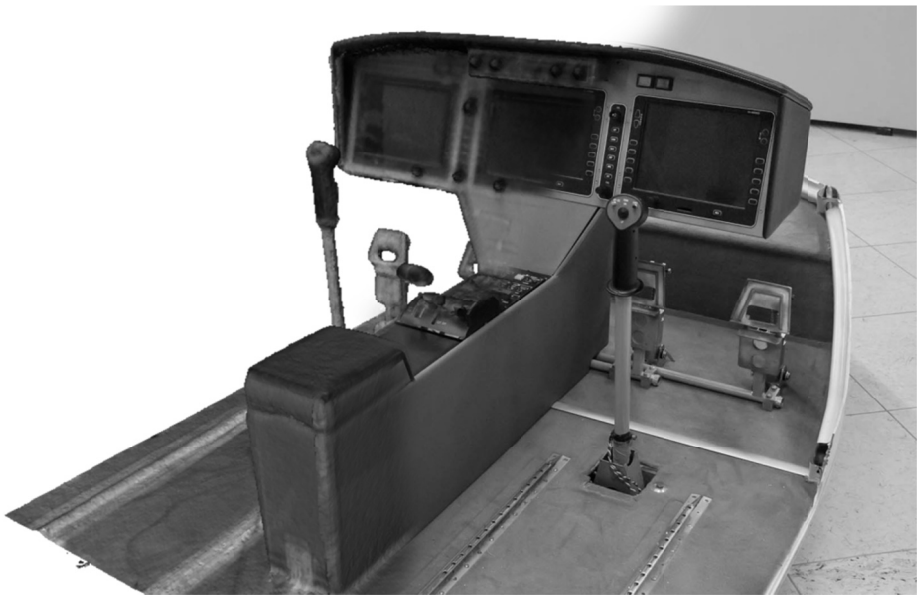


Bild 6: 3D-Scan eines existierenden Cockpits

In anderen Fällen wiederum ist es nötig, eine Geometrie komplett und detailgetreu zu modellieren (siehe Bild 6). Hier kann die Punktwolke des 3D-Scans ebenfalls als Unterstützung für eine CAD-Modellierung verwendet werden.

3.3 Integration konzeptioneller MMS-Komponenten

Da die Bedienung vollvirtueller Elemente unter Umständen schwierig sein kann, sollte für wichtige und oft benutzte Elemente das „Mixed Mock-up“ Konzept verwendet werden. Bei konzeptionellen MMS Komponenten sind oft noch keine Hardware Mock-ups vorhanden und die Verwendung vereinfachter Elemente (wie z.B. Acrylglasplatten) reicht oftmals nicht aus, um die benötigte Haptik zu erzeugen.

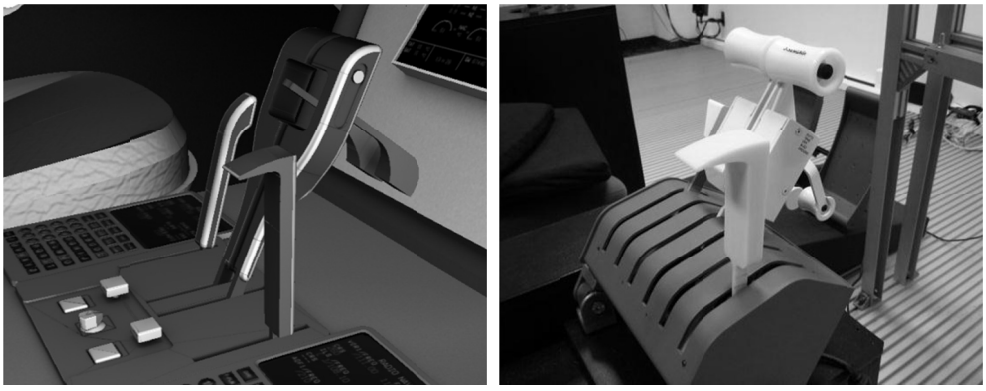


Bild 7: 3D-Druck eines Landeklappenhebels

Eine schnelle Alternative bietet sich durch die Möglichkeiten des 3D-Drucks. So können konzeptionelle Komponenten gedruckt und als „Mixed Mock-up“ in die Simulation eingebunden werden. Bedienelemente, wie zum Beispiel ein Hebel zur Bedienung der Landeklappen (siehe Bild 7), können somit schnell und mit der entsprechenden Haptik eingebunden werden.

4 Integration des virtuellen Flugsimulators in den Cockpitentwicklungsprozess

Der virtuelle Flugsimulator kann in mehreren Stufen der Cockpitentwicklung eingesetzt werden. Zu Beginn des Entwicklungsprozesses sind viele Merkmale der Nutzerschnittstellen noch nicht vollständig definiert. In dieser Phase werden meist Ergonomiestudien über Erreichbarkeiten von Bedienelementen und die Sicht nach außen durchgeführt. Hierfür bietet sich ein vollvirtuelles Cockpit an, da in dieser Phase noch keine Haptik benötigt wird. Später können Hardware-Flugkontrollen und Platten für ein einfaches haptisches Feedback hinzugefügt werden, um dem Nutzer die Möglichkeit zu geben, die Ergonomie des Cockpits im Flug zu testen. In einem solchen einfachen Setup lassen sich bereits in einer frühen Entwicklungsphase kognitiv-psychologische Aspekte evaluieren.

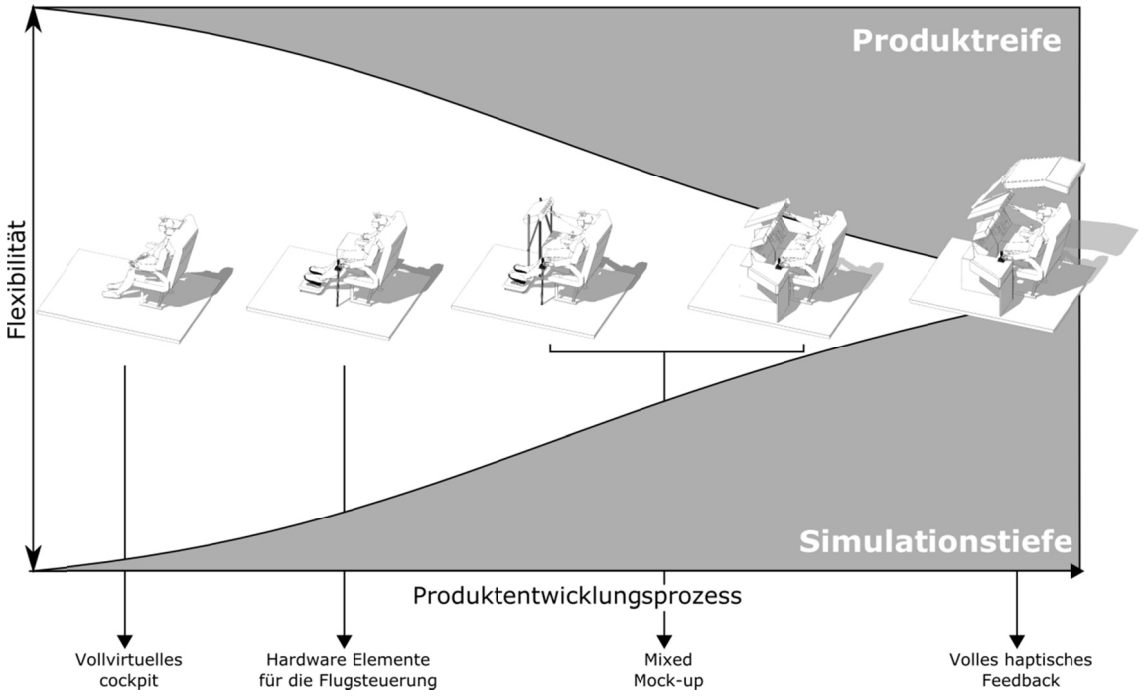


Bild 8: Integration des VRFS in den Cockpitentwicklungsprozess

Durch steigende Produktreife und somit determinierte Produktmerkmale werden erste Prototypen von einzelnen Systemen entwickelt. Diese lassen sich als funktionale Hardwareelemente in das virtuelle Cockpit integrieren. Dadurch steigt die Simulationstiefe erheblich und umfangreiche, realitätsnahe Szenarios lassen sich simulieren. Diese Erweiterung mit Hardwarekomponenten kann bis zum Erreichen eines kompletten Hardware Mock-ups vorangetrieben werden (siehe Bild 8).

Im Rahmen des Alicia (All Condition Operations and Innovative Cockpit Infrastructure) Projektes wurde der VRFS als Protoyping- und Evaluierungsplattform eingesetzt. Es wurden einzelne MMS Komponenten in ein konzeptionelles virtuelles Cockpit integriert und in einem operationellen Szenario mit Piloten getestet. Um den Einfluss der neuartigen Testumgebung auf die Ergebnisse zu minimieren, wurden nur Vergleichsstudien mit bereits existierenden Komponenten durchgeführt. Es wurden unter anderem eine neuartige Head-Up Display Symbologie [DBO14] sowie ein optimiertes Overhead Panel evaluiert.

5 Resümee und Ausblick

Der Virtual Reality Flight Simulator stellt eine wertvolle und kostengünstige Erweiterung der Prototypen dar, die im Entwicklungsprozess von Cockpits verwendet werden. Durch eine vollvirtuelle Darstellung und entsprechende Interaktionsmechanismen können Cockpits mit dem VRFS in einem funktionalen Simulationsumfeld bereits in einer sehr frühen Entwicklungsphase getestet werden. Durch die Anreicherung des Systems mit funktionalen Hardwarekomponenten kann der Simulator mit dem Entwicklungspro-

jekt wachsen und die Simulationstiefe entsprechend erhöht werden. Trotzdem kann auch ein Mixed Mock-up System nicht als vollwertiger Ersatz für einen Hardware Simulator verwendet werden, da weiterhin Einschränkungen durch die Virtual Reality Technologie, wie zum Beispiel eingeschränktes Sichtfeld oder Ungenauigkeiten im Tracking, bestehen. Weitere Entwicklungen zielen auf die einfachere Integration von MMS Komponenten ab ebenso wie auf die Erhöhung der Simulationstiefe durch verbesserte Hard- und Software.

Literatur

[RN04] REUZEAU, F.; NIBBELKE, R.: Flight deck design process. In: D. Harris (Hg.): Human Factors for Civil Flight Deck Design. Hampshire: Ashgate, S. 33-55, 2004.

[HK05] HÜSGEN, S.; KLINGAUF, U.: Interaktionsmechanismen zur Verbesserung der Mensch-Maschine-Schnittstelle am Beispiel eines virtuellen Flugsimulators. In: Deutscher Luft- u. Raumfahrtkongress, 2005.

[Bau14-ol] BAUER, M.: The Enhanced Virtual Environment (EVE): Learning lessons to improve safety, unter: http://www.highflyer.airbus-group.com/02_2014_tackling_the_cyber_threat.html, 1. März 2015

[YKT+09] YAVRUCUK, I.; KUBALI, E.; TARIMCI, O.; YILMAZ, D.: A Low Cost Flight Simulator Using Virtual Reality Tools. In: AIAA Modeling and Simulation Technologies Conference, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, Reston, 2009

[ADP+14] ASLANDERE, T.; DREYER, D.; PANTKRATZ, F.; SCHUBOTZ, R.: A Generic Virtual Reality Flight Simulator. In: Virtuelle und Erweiterte Realität, 11. Workshop der GI-Fachgruppe VR/AR. Shaker Verlag, S. 1-13, Aachen, 2014.

[DBO14] DREYER, D.; BANDOW, D.; OBERHAUSER, M.: HUD Symbology evaluation in a Virtual Reality Flight Simulation. In: HCI-Aero, 2014

[HOD+14] HILLEBRAND, A.; OBERHAUSER, M.; DREYER, D.; MEISTER D.: Modulare Erweiterbarkeit von Flugsimulationen für Human-Factors-Experimente. In: 3. Interdisziplinärer Workshop Kognitive Systeme: Mensch, Teams, Systeme und Automaten, Magdeburg, 2014

[LIE13] LIESECKE S.: Eye-Tracking in Virtual Reality. Mastersthesis. Universität der Bundeswehr Institut für Flugsysteme, München, 2013

[HS88] HART, S.; STAVELAND, L.: Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In: Advances in psychology 52, S. 139-183, 1988

[ST90] SELCON, S.; TAYLOR, R.: Evaluation of the situational awareness rating technique (SART) as a tool for aircrew systems design. In: Situational Awareness in Aerospace Operations (AGARD-CP-478). Neuilly Sur Seine, 1990

[End95] ENDSLEY, M.: Measurement of situation awareness in dynamic systems. In: Human Factors: The Journal of the Human Factors and Ergonomics Society 37 (1), S. 65-84, 1995

[MDO14] MAMESSIER, S.; DREYER, D.; OBERHAUSER, M.: Calibration of online Situation Awareness Assessment Systems using Virtual Reality. In: Proceedings of HCI-International 2014, Heraklion 2014

Autoren

M.Sc. Matthias Oberhauser arbeitet seit 2012 bei Airbus Group Innovations. Er schloss das Masterstudium Maschinenwesen an der Technischen Universität München mit den Schwerpunkten „Elektronik und Informatik“ und „Luftfahrttechnik“ 2014 ab und arbeitet seitdem an seiner Promotion an der Technischen Universität Graz. Seine Forschungsthemen sind Mensch-Maschine-Interaktion im Cockpit und die Nutzung von Virtual Reality für Human Factors Studien.

Dr.-Ing. Daniel Dreyer arbeitet seit 2003 bei Airbus Group Innovations. Er studierte und promovierte an der Technischen Universität München (Luft- und Raumfahrttechnik/ Ergonomie). Der Titel seiner Dissertation lautet „Kontrastschwellensimulation für Sichtbarkeitsuntersuchungen an Displays“.

Primož Kovačič studiert Maschinenbau an der Technischen Universität Graz und beschäftigt sich zurzeit mit der 3D Modellierung des Flugsimulators am Institut für Mechanik für die Integration in einen virtuellen Flugsimulator.

Dieser Beitrag ist in der Online-Version leider nicht verfügbar.

Anbindung des Virtuellen Prototypen an die Partialmodelle intelligenter technischer Systeme

Jan Berssenbrügge, Olga Wiederkehr
Heinz Nixdorf Institut
Universität Paderborn
Fürstenallee 11, 33102 Paderborn
Tel. +49 (0) 5251 / 60 62 32, Fax. +49 (0) 5251 / 60 62 68
E-Mail: {Jan.Berssenbruegge|Olga.Wiederkehr}@hni.upb.de

Claudius Jähn, Matthias Fischer
Heinz Nixdorf Institut und Institut für Informatik
Universität Paderborn
Fürstenallee 11, 33102 Paderborn
Tel. +49 (0) 5251 / 60 64 66, Fax. +49 (0) 5251 / 60 64 82
E-Mail: {claudius|mafi}@upb.de

Zusammenfassung

Die fachdisziplinübergreifende Systembeschreibung Intelligenter Technischer Systeme erfolgt anhand einer Reihe von Partialmodellen. Die einzelnen Partialmodelle beleuchten jeweils detailliert spezifische Aspekte des Gesamtsystems, wie die Wirkstruktur oder die Gestalt. Die Partialmodelle sind untereinander stark vernetzt. Im Bereich des Partialmodells Gestalt sind die Verknüpfungen zwischen Gestalt und anderen Partialmodelle zwar bekannt, jedoch nicht in Form einer visuellen Darstellung umgesetzt. Der Virtuelle Prototyp, als digitales Abbild des Gesamtsystems, bildet einen geeigneten Ansatzpunkt für die Weiterentwicklung des Gestaltmodells. Ziel ist eine Darstellung der Zusammenhänge zwischen den Partialmodellen im Rahmen der Gestalt, um das Systemverständnis zu erleichtern. Hierzu sollen Aspekte des Gesamtsystems am Virtuellen Prototypen aufgezeigt und verortet werden. Anhand der Spezifikation eines Regalbediengeräts im Rahmen eines Transferprojektes aus dem Spitzencluster it´s OWL stellt der Beitrag die Anbindung einzelner Partialmodelle an den Virtuellen Prototypen und die prototypische Umsetzung vor.

Schlüsselworte

Partialmodelle, Wirkstruktur, Umfeldmodell, Aktivitätsdiagramm, Virtueller Prototyp, Intelligente Technische Systeme

Linking the Virtual Prototype with the Partial Models of Intelligent Technical Systems

Abstract

The interdisciplinary specification of intelligent technical systems is carried out based on a set of partial models. The individual partial models highlight specific aspects of the overall system respectively, e.g. the active structure or the shape. The partial models are highly cross-linked among each other. In fact, for the partial model shape these cross-links between the shape and other partial models are known, however, they are not visually translated. The virtual prototype, as a digital model of the overall system, forms a suitable starting point for the further development of the shape model. The aim is an illustration of the interrelation between the partial models in the context of the shape, in order to facilitate the system understanding. For this, aspects of the overall system are to be depicted and localized at the virtual prototype. Based on the specification of a storage handling system in the context of a transfer project in the leading edge cluster it´s OWL the paper introduces a link between individual partial models an the virtual prototype along with the implementation prototype.

Keywords

Partial Models, Active Structure, Activity Diagram, Virtual Prototype, Intelligent Technical Systems

1 Einleitung

Virtual Prototyping, d.h. die Simulation eines Prototypen anhand des digitalen Produktmodells, ist heute ein wesentlicher Bestandteil bei der Entwicklung von Intelligenten Technischen Systemen (ITS). Der Virtuelle Prototyp besteht dabei im Wesentlichen aus dem 3D-CAD-Modell des ITS, welcher im Rahmen von Virtuellen Design Reviews anhand von Abmaß,- Toleranz-, Baubarkeit-, Kinematikanalysen und weiteren Untersuchungen auf seine geometrischen Eigenschaften überprüft wird. Das 3D-CAD-Modell als Grundlage für das Virtual Prototyping beschreibt jedoch nur die Gestalt und ist damit nur eines von mehreren Elementen, welche zur Entwicklung von Intelligenten Technischen Systemen aufgestellt werden. Denn, bei der Entwicklung von ITS ist bekanntermaßen eine Vielzahl unterschiedlicher Fachdisziplinen involviert; hierzu zählt die Mechanik, Elektrik/ Elektronik, Regelungstechnik sowie die Softwaretechnik. Der Begriff Mechatronik bringt dies zum Ausdruck.

Die zunehmende Multidisziplinarität stellt dabei einen wesentlichen Komplexitätstreiber in der Entwicklung dar. Methoden des Systems Engineering können einen wesentlichen Beitrag leisten diesen Herausforderungen zu begegnen; sie fördern die interdisziplinäre Zusammenarbeit und betrachten das System als Ganzes [INC13], [FMS12]. Die modelltechnische Abbildung der Zusammenhänge adressiert das Themenfeld Modelbased Systems Engineering (MBSE). Im Rahmen des MBSE wird ein Systemmodell in den Mittelpunkt der Entwicklung multidisziplinärer Systeme gestellt. Das Systemmodell bildet alle relevanten Aspekte von Beginn an ab und ist die wesentliche Grundlage für die Kommunikation und Kooperation der involvierten Fachleute. Ein Ansatz zur Erarbeitung eines Systemmodells, bildet die fachdisziplinübergreifende Spezifikationstechnik CONSENS (CONceptual design Specification technique for the ENgineering of mechatronic Systems) [GFD+08]. Sie bildet die wesentlichen Elemente zur Entwicklung eines ITS in Form von Partialmodellen ab, beispielhaft seien an dieser Stelle die Wirkstruktur, das Umfeldmodell, Anwendungsszenarien oder eben auch die Gestalt genannt.

Die Partialmodelle sind untereinander stark vernetzt, im Bereich des Partialmodells Gestalt gibt es jedoch Weiterentwicklungspotentiale: So sind die Verknüpfungen zwischen der Gestalt und anderen Partialmodelle zwar bekannt, jedoch nicht in Form einer visuellen Darstellung umgesetzt. Konkret geht es um die Visualisierung weiterer Partialmodelle der Systemspezifikation im Rahmen des Gestaltmodells unter Zuhilfenahme von Methoden der Virtual Reality. Der resultierende Virtuelle Prototyp ermöglicht durch die Veranschaulichung und Verortung der Wirkflächen und -prinzipien eine visuelle Analyse von Wirkzusammenhängen zwischen den unterschiedlichen Partialmodellen. Dies unterstützt die multidisziplinären Entwicklungsteams und fördert das Systemverständnis. Hier setzt der vorliegenden Beitrag an.

Ziel des Beitrags ist die Visualisierung der Partialmodellen Wirkstruktur, Umfeld und Anwendungsszenarien auf Basis der Gestalt durch Anbindung an einen virtuellen Proto-

typ. Der Beitrag stellt daher die Anbindung aus Sicht des virtuellen Prototypen in Richtung der Partialmodelle über die Anreicherung des 3D-CAD-Modells durch funktionale Proxy-Objekte (engl. Proxy für Stellvertreter) her. Durch die aspektorientierte Konfiguration dieser Objekte können Funktionen, wie Animationen oder physikalische Eigenschaften, und Verknüpfungen zwischen den Modellen flexibel umgesetzt werden. Der funktionale Nachweis erfolgt anhand der prototypischen Implementierung im Rahmen des Transferprojektes iRegabi des Spitzenclusters it´s OWL. Untersuchungsgegenstand ist ein intelligentes Regalbediengerät eines Herstellers für Lagersysteme aus der Region OWL.

Der folgende Abschnitt 2 gibt zunächst einen Überblick über die Partialmodelle zur Beschreibung der verschiedenen Aspekte eines ITS im Rahmen der fachdisziplinübergreifenden Systemspezifikation. Es folgt die Visualisierung ausgewählter Elemente der weiteren Systemspezifikation im Gestaltmodell in Form eines Virtuellen Prototypen und die Vorstellung der Anwendung im Rahmen des Transferprojektes aus den Spitzencluster it´s OWL in Abschnitt 3. Abschnitt 4 thematisiert die prototypische Umsetzung der Anbindung anhand des Partialmodells Anwendungsszenario an den Virtuellen Prototypen auf Basis des PADrend-Systems. Der Beitrag schließt mit einem Ausblick auf die nächsten Arbeiten.

2 Fachdisziplinübergreifende Systemspezifikation

Die Leistungsfähigkeit der Spezifikationstechnik CONSENS sowie ihr Beitrag zur Verbesserung der Kommunikation und Kooperation der involvierten Fachleute konnte bereits in zahlreichen Industrieprojekten unter Beweis gestellt werden. Zur Beschreibung des Systems stellt CONSENS die folgenden sieben Aspekte zur Verfügung: Umfeld, Anwendungsszenarien, Anforderungen, Funktionen, Wirkstruktur, Gestalt und Verhalten (siehe Bild 1).

Nachfolgend werden die einzelnen Partialmodelle näher kurz erläutert [GLL12], [GTS14]:

- Das **Umfeldmodell** beschreibt die Einbettung des zu entwickelnden in sein Umfeld. Dabei wird das System zunächst als Black Box betrachtet und mit den Elementen, mit denen es interagiert, den sog. Umfeldelementen, in Beziehung gesetzt.
- **Anwendungsszenarien** charakterisieren Betriebssituationen, für die das System auszulegen ist und spezifizieren gleichzeitig, in welcher Art und Weise sich das System in dieser Situation verhalten soll. Somit charakterisieren Anwendungsszenarien für einen bestimmten Fall das zu lösende Problem.
- Bei dem Partialmodell **Anforderungen** handelt es sich um die rechnerinterne Repräsentation der Anforderungen, deren Basis die Anforderungsliste ist. Diese gelten während der gesamten Produktentwicklung als „Messlatte“ für das zu

entwickelnde Produkt, der es genügen muss. Jede Anforderung wird durch Attribute und deren Ausprägung konkretisiert.

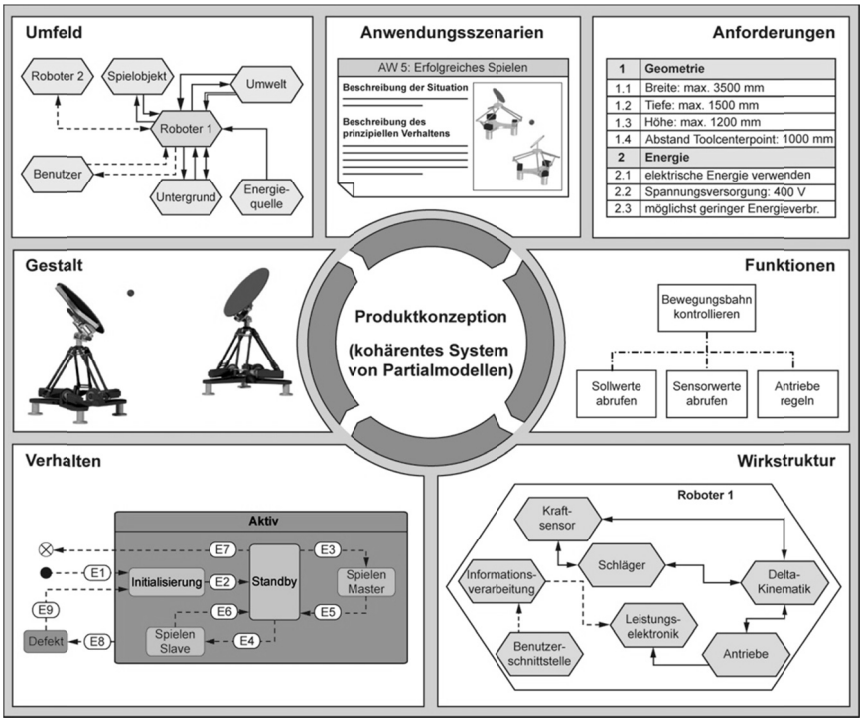


Bild 1: Partialmodelle der Spezifikationstechnik CONSENS [GTS14]

- **Funktionen** beschreiben die hierarchische Aufgliederung der Funktionalität des zu entwickelnden Systems. Eine Funktion ist der allgemeine und gewollte Zusammenhang zwischen Eingangs- und Ausgangsgrößen mit dem Ziel, eine Aufgabe zu erfüllen. Bei der Erstellung der Funktionshierarchie wird die Hauptfunktion so lange in Subfunktionen untergliedert bis sinnvolle Lösungen gefunden werden können – die Erstellung derartiger Funktionshierarchien kann durch Funktionskataloge unterstützt werden.
- Ziel der **Wirkstruktur** ist die Abbildung der grundsätzlichen Struktur und der prinzipiellen Wirkungsweise des Systems. Beschrieben werden Systemelemente, deren Merkmale sowie die Beziehungen der Systemelemente zueinander. Systemelemente repräsentieren Systeme, Module, Bauteile oder Softwarekomponenten; Stoff-, Energie- und Informationsflüsse sowie logische Beziehungen beschreiben die Wechselwirkungen zwischen den Systemelementen.
- Während des Systementwurfs sind erste Festlegungen der **Gestalt** des Systems vorzunehmen. Diese sind ebenfalls Teil der Systemspezifikation. Dieses Partialmodell umfasst Angaben über Anzahl, Form, Lage und Anordnung und Art der Wirkflächen und Wirkorte des Systems. Die rechnerinterne Repräsentation erfolgt mit Hilfe gängiger 3D-CAD-Systeme.

- Das Partialmodell **Verhalten** umfasst die Modellierung von Aktivitäten, Zuständen und Zustandsübergängen sowie deren Auswirkungen auf die Wirkstruktur.

Die genannten Aspekte stehen zueinander in Beziehung und ergeben ein konsistentes Ganzes; daher besteht das Systemmodell aus einem kohärenten System von Partialmodellen. Die Aspekte und die entsprechenden Partialmodelle sind im Wechselspiel zu bearbeiten, wenngleich es eine gewisse Reihenfolge gibt [GFD+08], [GTS14].

3 Visualisierung der Systemspezifikation am Virtuellen Prototypen

Die Multidisziplinarität und die Produktkomplexität Intelligenter Technischer Systeme stellen die involvierten Fachdisziplinen im Rahmen ihrer Entwicklung vor eine Vielzahl an Herausforderungen. In diesem Zusammenhang kommt der zielgerichteten Interaktion und Kooperation der am Produktentstehungsprozess beteiligten Akteure eine besondere Rolle zuteil. Dies setzt das rasche Erlernen und Verstehen insbesondere neuer bzw. disziplinfremder Informationen voraus. Eine Möglichkeit dieser Herausforderung zu begegnen stellt der Einsatz von Virtual Reality-Methoden im Kontext der Systemspezifikation dar. Diese ermöglichen bspw. eine visuelle Analyse von Wirkzusammenhängen zwischen den unterschiedlichen Partialmodellen. Abweichungen zwischen Partialmodellen können dargestellt und notwendige Änderungen sofort initiiert werden. Multidisziplinäre Entwicklerteams werden in die Lage versetzt, disziplinübergreifende Synergien und Widersprüche frühzeitig zu erkennen und so die entwickelte Lösung zu verbessern.

Ausgangsbasis ist der Virtuelle Prototyp als ein rechnerinternes Modell des ITS. Der Virtuelle Prototyp visualisiert auf der Basis des Gestaltmodells zunächst lediglich die Baugruppen und Komponenten des technischen Systems und simuliert das Verhalten der Systemelemente. Damit bildet der Virtuelle Prototyp eine vielversprechende Plattform, indem er die Zuordnung und räumliche Verortung der einzelnen Aspekte aus den verschiedenen Partialmodellen direkt an den Baugruppen und Systemelementen des Virtuellen Prototypens ermöglicht. Diese Veranschaulichung und Verortung der Wirkflächen und -prinzipien unterstützt die multidisziplinären Entwicklungsteams und fördert das Systemverständnis.

Das Fernziel ist die informationstechnische Anbindung des Virtuellen Prototypen an CONSENS-basierte Modellierungs-Tools. In dieser Arbeit wird als erster Schritt in diese Richtung eine methodische Anbindung des Virtuellen Prototypen an ausgewählte Partialmodelle auf der Basis des Renderingsystems PADrend umgesetzt.

Im Rahmen des Transferprojekts iRegabi des Spitzenclusters it´s OWL entsteht hierfür eine erste prototypische Implementierung. Den Ausgangspunkt stellt ein intelligentes Regalbediengerät eines Herstellers für Lagersysteme aus der Region OWL. Im Rahmen

des Projektes wird das Regalbediengerät anhand ausgewählter Partialmodelle spezifiziert und diese anschließend an den Virtuellen Prototypen angebunden.

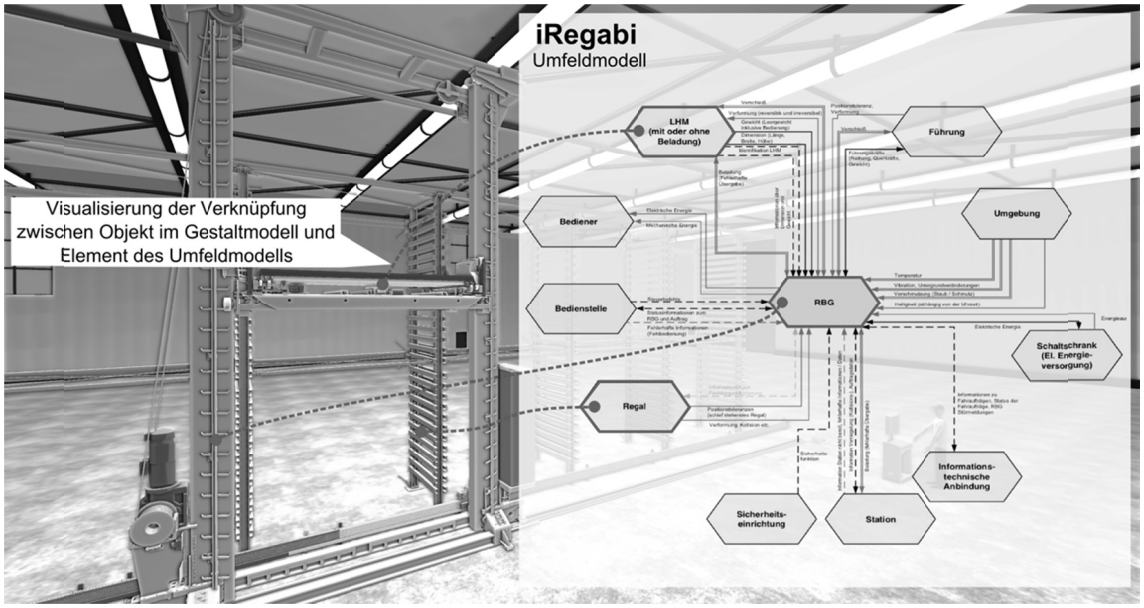


Bild 2: Anbindung des Virtuellen Prototypen (Partialmodell der Gestalt) des Regalbediengeräts an das Partialmodell zur Darstellung der Wechselwirkung mit dem Umfeld (Quelle: it´s OWL QP MMI und Friedrich Remmert GmbH)

Im Rahmen der Systemspezifikation wird zunächst das Umfeld des Regalbediengerätes durch die **Umfeldelemente** definiert, mit denen das Regalbediengerät interagiert. Die einzelnen Umfeldelemente werden durch Links zu den entsprechenden Baugruppen am Virtuellen Prototyp verortet (siehe. Bild 2).

Anschließend werden in **Anwendungsszenarien** die Prozessabläufe in typischen Betriebssituationen beschrieben. Diese Szenarien werden im Rahmen des Projekts u.a. mittels Aktivitätsdiagrammen dokumentiert. Am Virtuellen Prototypen werden z.B. die entsprechenden Bewegungsabläufe des ausgewählten Aktivitätschrittes animiert (vgl. Bild 3). Die Aktivitätsdiagramme gliedern die Prozessabläufe zeitlich und umfassen neben den Aktivitäten auch die damit verbundenen Systemzustände.

Das Partialmodell **Wirkstruktur** beschreibt die grundsätzliche Struktur des ITS und bildet Stoff-, Energie- und Informationsflüsse ab. Am Regalbediengerät sind dies z.B. konkrete Ablaufsimulationen der beweglichen Systemelemente (Stoffffluß) und Informationsflüsse der im Regalbediengerät verbauten Sensorik.

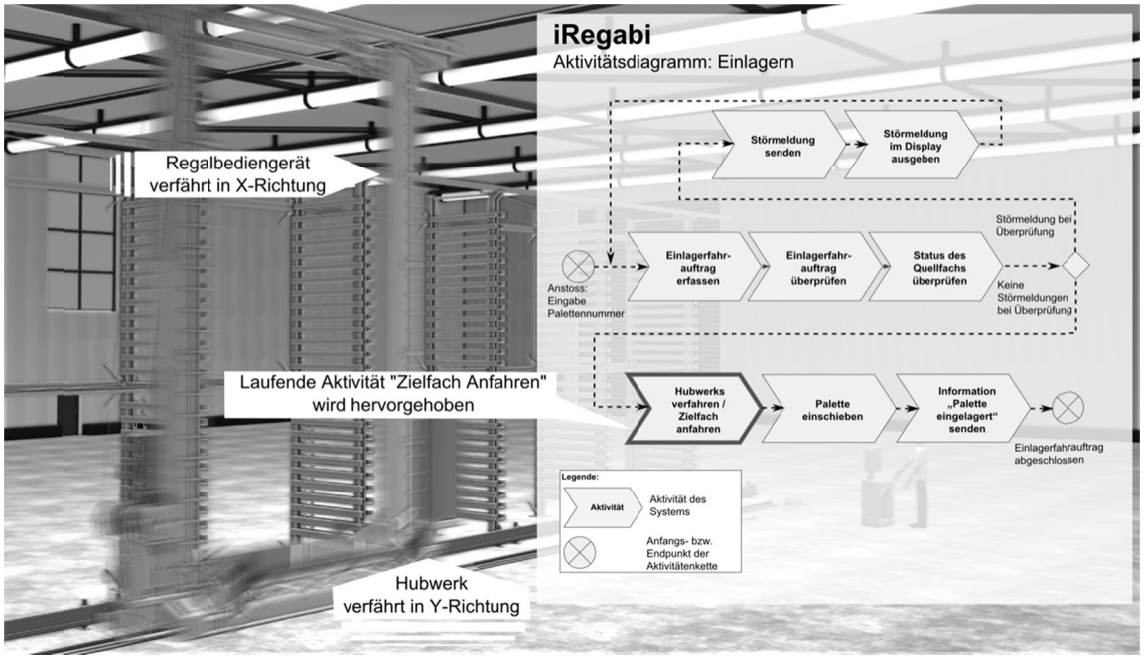


Bild 3: Anbindung des Virtuellen Prototypen des Regalbediengeräts an das Aktivitätendiagram zur Beschreibung der Anwendungsszenarien (Quelle: it´s OWL QP MMI und Friedrich Remmert GmbH)

Zwischen den einzelnen Partialmodellen existieren Querbezüge. So spezifiziert beispielsweise das Aktivitätsdiagramm der Anwendungsszenarien die zeitliche Abfolge eines Prozesses. Die einzelnen Prozessschritte können wiederum Stoff-, Energie- oder Informationsflüssen zugeordnet werden und sind damit im Partialmodell Umfeld oder Wirkstruktur zu verorten (siehe Bild 4). Am Virtuellen Prototypen können diese Querbezüge anschaulich dargestellt werden.

Wir blenden die Partialmodelle über den Virtuellen Prototypen im PADrend-System ein. Durch Auswahl von Elementen der Partialmodelle werden die korrespondierenden Elemente im Virtuellen Prototypen hervorgehoben und der Bezug visualisiert. So werden beispielsweise Aktivitäten, die im Partialmodell abstrakt spezifiziert sind, am Virtuellen Prototypen ausgeführt, anschaulich dargestellt und können somit leichter nachvollzogen werden.

Umgekehrt können am Virtuellen Prototypen einzelne Systemelemente ausgewählt werden, um die Verbindungen zu den korrespondierenden Elementen in den entsprechenden Partialmodellen darzustellen. Hierdurch fungiert der Virtuelle Prototypen als eine Art zusätzliches Bindeglied zur einfachen Navigation durch die beteiligten Partialmodelle. Hierdurch werden Querbezüge zwischen den einzelnen Partialmodellen sofort deutlich und Wirkzusammenhänge können schneller nachvollzogen werden. Dieses fördert das Systemverständnis.

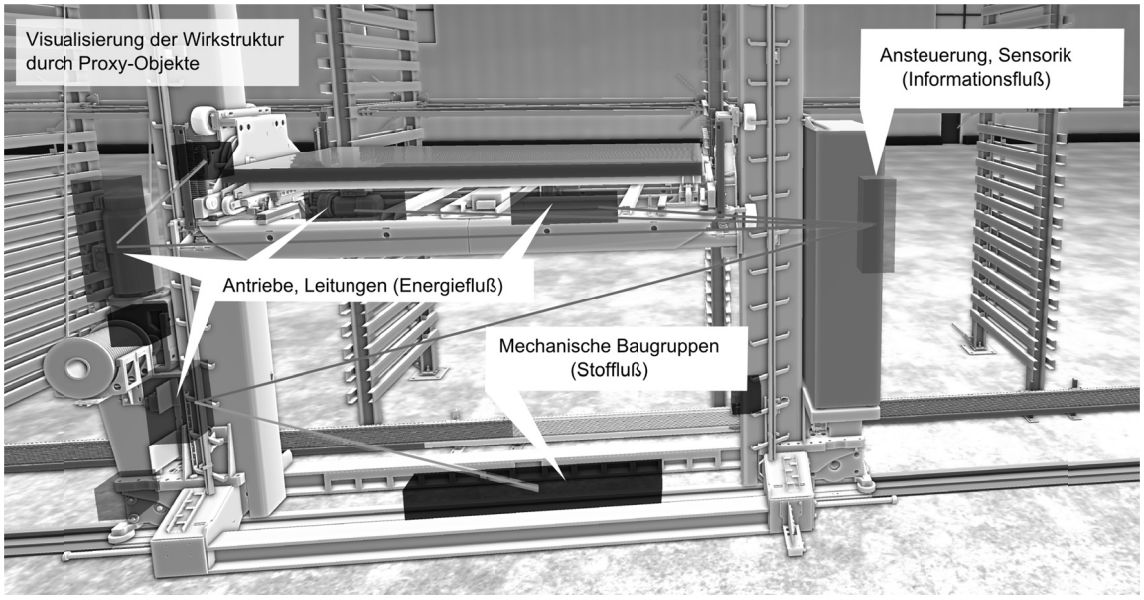


Bild 4: Anbindung des Virtuellen Prototypen des Regalbediengeräts an das Partialmodell Wirkstruktur (Quelle: it´s OWL QP MMI und Friedrich Remmert GmbH)

4 Umsetzung der Verknüpfungen mit funktionalen Proxy-Objekten in PADrend

Im Folgenden stellen wir die prototypische Umsetzung auf Basis des PADrend-Systems [PAD14] vor, welches für das Design Review virtueller Prototypen von ITS im Rahmen des Querschnittprojekts Mensch-Maschine-Interaktion des Spitzenclusters it´s OWL weiterentwickelt wird und in Transferprojekten für das Virtual Prototyping von ITS bereits eingesetzt wird. Wir beschreiben, wie die Anbindung der Partialmodelle an den Virtuellen Prototypen durch *funktionale Proxy-Objekte* umgesetzt wurde. Der Fokus liegt dabei auf der Umsetzung des Verhaltens des Virtuellen Prototypen, das in den Aktivitätsdiarammen definiert wird und durch die Anbindung des Aktivitätsdiagrammes ausgelöst werden soll. Im Folgenden beziehen wir uns konkret auf das Beispiel der Aktivität „Fach anfahren“, bei dem das Regalbediengerät sich vor einem vorgegebenen Fach des Regallagers positioniert.

4.1 Beschreibung der Eingabe

Als Basis für die Darstellung des Virtuellen Prototypen im VR-System werden trianguläre Oberflächenmodelle der Bauteile benötigt. Diese können durch viele 3D-CAD-Systeme direkt exportiert werden, oder können über externe Softwarelösungen aus den parametrisch definierten 3D-CAD-Daten des Prototypen erzeugt werden. Bei der Konvertierung sollte die Benennung der Bauteile erhalten bleiben, um eine eindeutige Zuordnung von virtuellem Teilobjekt zu seinem zugrundeliegenden Bauteil zu erhalten. Auch wenn in 3D-CAD-Systemen Kinematiken und Animationen mit modelliert wer-

den können, sind diese Daten nicht notwendig für die Anreicherung des Virtuellen Prototypen – und werden in vielen Fällen in der Praxis auch nicht erstellt. Wir gehen wir davon aus, dass nur die statische Struktur des Prototypen vorliegt.

4.2 Proxy-Objekte als Schnittstelle

Sobald die triangulierten Daten in das VR-System geladen werden, bilden die Baugruppen und Bauteile die Knoten eines Szenengraphen, der die hierarchische Baustruktur des Regalbediengerätes nachbildet. Die Funktionen, mit der der Prototyp angereichert wird, werden nicht direkt mit den Knoten der Baustruktur verknüpft, sondern mit sogenannten Proxy-Objekten. Dies hat im Wesentlichen zwei Gründe: Zum einen ist die Struktur, die sich aus dem Teilebaum ergibt, oft ungeeignet für Animationen (z. B. können alle elektrischen Leiter in einem gemeinsamen Teilbaum liegen, über die Grenzen von unabhängig bewegbaren Teilen hinweg). Zum anderen wird versucht, die importierten Daten so wenig wie nötig anzupassen, um mit geringem Aufwand einen aktualisierten 3D-CAD-Datensatz in die existierende, virtuelle Szene integrieren zu können, ohne die Anpassung an die Partialmodelle erneut vornehmen zu müssen.

Proxy-Objekte sind im Normallfall unsichtbare Knoten des Szenengraphen, die stellvertretend für die tatsächlichen Objekte animiert und mit Eigenschaften angereichert werden (ähnlich z. B. mit Dummy-Objekten in 3ds Max von Autodesk). Ein Proxy-Objekt kann mit einer Reihe von Objekten des Virtuellen Prototyps verbunden werden, und kann diese dann auf spezifizierte Weise beeinflussen (bewegen, ausblenden, hervorheben, etc.). Die Referenzierung zwischen den Proxy-Objekten und den zugehörigen Bauteilen erfolgt über die Benennung, so dass die Beziehung bei einer Aktualisierung der CAD-Daten erhalten bleibt (wenn die Baugruppe bei der Aktualisierung nicht verändert wurde).

4.3 Komposition durch funktionaler Aspekte

Die Flexibilität des von uns gewählten Ansatzes liegt in der Möglichkeit begründet, das Verhalten und die Funktionalität eines Proxy-Objektes, durch funktionale Aspekte nach Bedarf komponieren zu können. Zur Laufzeit kann so ein Proxy-Objekt mit zahlreichen, unterschiedlichen Aspekten aus einem Katalog von Aspekten ausgestattet werden. Die Konfiguration der Parameter erfolgt über ein automatisch generiertes GUI-Interface, dass durch die Aspekte definiert wird (siehe Bild 5). Die folgende Aufzählung gibt einen Überblick über einige der verfügbaren Aspekte:

Verknüpfungen: Dieser Aspekt stellt die grundlegende Möglichkeit bereit, andere Knoten des Szenengraphen (also sowohl Bauteile aus dem CAD-Modell, als auch weitere Proxy-Objekte) zu referenzieren. Dieser Aspekt wird von vielen weiteren Aspekten ausgenutzt. Die Verknüpfungen lassen sich als farbige Linien zwischen den Objekten visualisieren und können so Beziehungen zwischen Objekten darstellen. Dies wird auch

dafür genutzt, Flüsse (wie Energie und Information) der Wirkstruktur zwischen einzelnen Teilen des virtuellen Prototypen darzustellen. Durch den zusätzlichen Aspekt des Bewegungsproxies, lässt sich die Bewegung des Objektes konfigurierbar auf verknüpfte Objekte übertragen.

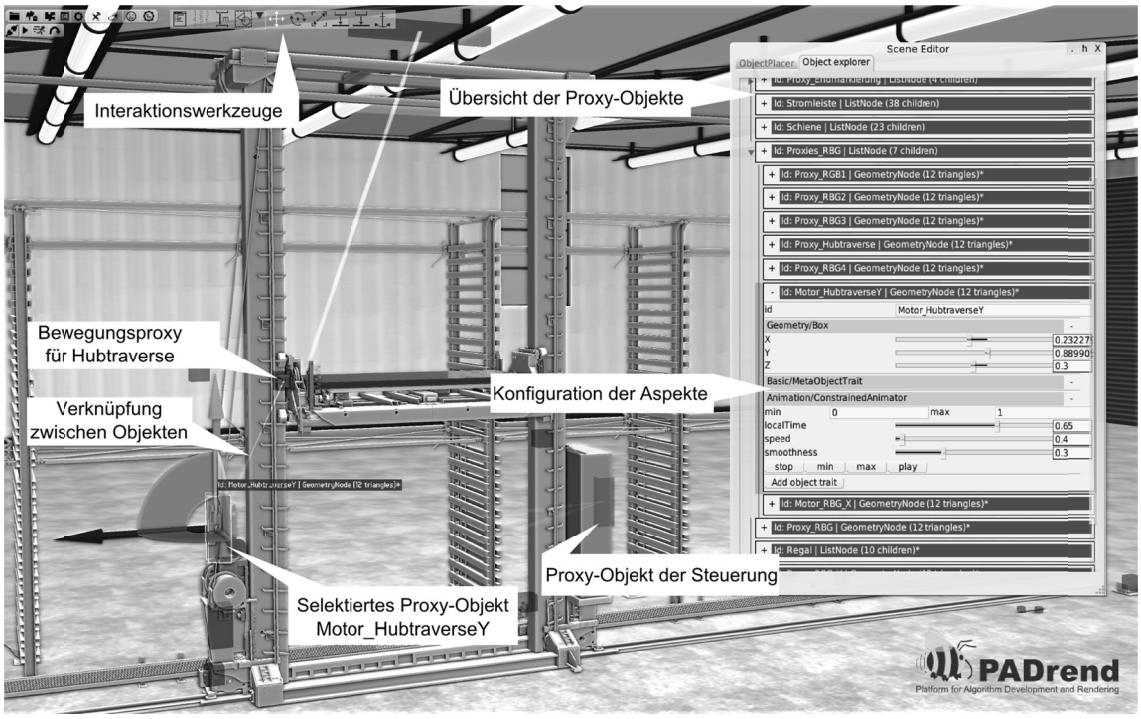


Bild 5: Benutzungsschnittstelle im PADrend-System zur interaktiven Generierung und Konfiguration von Proxy-Objekten

Zeitgeber/ Animationen: Durch verschiedene Zeitgeberaspekte, können Proxy-Objekte zeitlich gesteuertes Verhalten auslösen. Objekte mit Animationsaspekten, wie Rotationen oder Key-Frame-Animationen werden mit einem Zeitgeber verbunden. Der Zeitgeber entspricht damit einem virtuellen Motor, der die verbundenen animierten Objekte (wie Zahnräder) antreibt. Die Parameter des Motors (Beschleunigung, Maximalgeschwindigkeit, etc.) werden über Parameter des Zeitgebers konfiguriert. Das Übersetzungsverhältnis der verknüpften Bauteile wird mit den Animationsaspekten konfiguriert. Über Schalteraspekte können Objekte als interaktiven Schalter verwendet werden. Durch ein Klicken mit der Maus, einer Bestätigung mit einem Flystick in VR oder durch Bestätigung des Schalters im automatisch generierten Webinterfaces, können die verknüpften Zeitgeber gesteuert werden.

Im Beispiel werden die beteiligten Motoren zum Verfahren des Regalbediengeräts als Ganzes (X-Achse) und der Bewegung der Hubtraverse (Y-Achse) jeweils durch ein Proxy-Objekt mit Zeitgeberaspekt nachgebildet. Bewegungsproxies mit Key-Frame-Animationen (mit den Endpositionen) für das Regalbediengerät als Ganzes und für die Hubtraverse, werden mit den Objekten des CAD-Modells verbunden und bestimmen die Bewegung der Baugruppen. Auch die an der Bewegung beteiligten Zahnräder werden

auf diese Weise mit dem Rotationsaspekt in die Animation eingebunden. Durch virtuelle Schalter am Bedienterminal lassen sich die Zeitgeber manuell steuern, wodurch der Handfahrbetrieb des Geräts nachgebildet wird.

Physikalische Aspekte: Um komplexere Abläufe mit überschaubarem Aufwand realistisch darstellen zu können, werden Aspekte verwendet, die den Objekten physikalische Eigenschaften zuweisen, die dann von einer Physics Engine [Bul13-ol] zur Laufzeit simuliert werden. Im Beispiel werden die Ziehnocken des Regalbediengeräts, die Auflagen der Regalpaletten und des Regals mit physikalischen Aspekten angereichert. Dadurch werden durch eine realistische Animation des Regalbediengeräts und der Ziehnocken, automatisch die Bewegungen der Paletten simuliert. Dadurch lassen sich auch komplexe Umlagervorgänge mit mehreren Paletten umsetzen, ohne zusätzlichen Aufwand zur Animation der Paletten.

Erweiterte Kontrollaspekte: Dank der zugrundeliegenden Skripting-Engine [ES14-ol], lassen sich auch komplexere Funktions- und Steuerungsabläufe für eine konkrete Anwendung zur Laufzeit als Aspekt an ein Objekt binden oder ändern. Teile der Verhaltenslogik des Virtuellen Prototypen können als spezialisierte Aspekte nachimplementiert werden oder es können über Schnittstellen (wie Netzwerk, RS232, etc.) externe Software- oder Hardwarelösungen angeschlossen werden. Rückmeldung des Virtuellen Prototypen lassen sich durch Sensoraspekte realisieren, die beispielsweise als virtuelle Lichtschranken funktionieren. Das Ziel einer solchen Simulation bleibt jedoch die Darstellung des Verhaltens; belastbare Simulationsergebnisse einer virtuellen Inbetriebnahme liegen nicht im Fokus dieser Arbeit.

Im Beispiel wird eine vereinfachte Version der grundlegenden Steuerung des Regalbediengerätes als spezialisierter Aspekt zu einem Steuerungsobjekt hinzugefügt. Das Objekt wird räumlich in den Schaltschrank der Anlage platziert, so dass die Verknüpfungen des Objektes direkt den logischen Verbindungen des Informationsflusses des Virtuellen Prototypen aus der Wirkstruktur entsprechen. Zur Steuerung erlaubt der Aspekt die Eingabe einer Fachkoordinate, woraufhin die Repräsentanten der Motoren entsprechend angesteuert werden. Durch die hinterlegten Animationen und Bewegungsproxies bewegt sich das virtuelle Regalbediengerät an die gewünschte Position und kann über eine virtuelle Lichtschranke den Belegtstatus des Faches prüfen. Eine ggf. geladene Palette wird automatisch mit dem Regalbediengerät verfahren, da sie durch die Physikeigenschaften auf der Hubtraverse aufliegt.

2D- / 3D-Verknüpfung: Um die Beziehung der Proxy-Objekte in 3D und der Elementen der Partialmodelle in der 2D-Darstellung herzustellen, existiert ein Aspekt, der eine Verbindung vom 3D-Raum zu einem eingeblendeten Element im 2D-Raum visualisiert. Durch Auswahl des Objektes in 3D oder des Elementes in 2D werden beide Seiten hervorgehoben (farbig und verdeckte Elemente werden sichtbar) und die Verbindung wird durch eine Linie visualisiert. Dadurch können beispielsweise die Elemente der Wirkstruktur mit einer direkten Repräsentation im virtuellen Prototypen dargestellt

werden. In der prototypischen Implementierung erfolgt die Zuordnung der 2D-Flächen noch manuell, da bis jetzt nur eine graphische Repräsentation der Partialmodelle im VR-System verwendet wird.

5 Resümee und Ausblick

Die Grundlagen für die in diesem Beitrag vorgestellte Anbindung von Partialmodellen an den Virtuellen Prototypen eines intelligenten technischen Systems zur Förderung des Systemverständnisses und Unterstützung der an der Entwicklung beteiligten Akteure wurden im Rahmen des Querschnittsprojektes Mensch-Maschine-Interaktion erarbeitet und prototypisch implementiert. Eine erste Evaluation anhand der Übertragung der Grundlagen auf ein konkretes Anwendungsbeispiel erfolgte im Rahmen des Transferprojektes iRegabi durch die Anbindung der Partialmodelle Wirkstruktur, Umfeld und Verhalten an den Virtuellen Prototypen eines Regalbediengerätes. Die erreichten Resultate erscheinen vielversprechend und das Feedback seitens der Projektpartner ist positiv. Es ist daher geplant, die bestehende Anbindung an den Virtuellen Prototypen auf die verbleibenden Partialmodelle auszuweiten und ggf. an CONSENS-basierte Modellierungs-Tools anzubinden.

Danksagung

Dieses Forschungs- und Entwicklungsprojekt wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) im Rahmen des Spitzenclusters „Intelligente Technische Systeme OstWestfalenLippe (it´s OWL)“ gefördert und vom Projektträger Karlsruhe (PTKA) betreut. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Wir danken den Projektpartnern von der Friedrich Remmert GmbH für die freundliche Bereitstellung von Informationen und Daten zu dem betrachteten Regalbediengerät und die gute Zusammenarbeit im Rahmen des it´s OWL-Transferprojektes iRegabi.

Literatur

[Bul13-ol] BULLET: Physics Engine. Unter: http://bulletphysics.org, abgerufen am 4. Dezember 2014

[ES14-ol] ESCRIPT: Extensible Script Language. Unter: http://github.com/EScript, abgerufen am 4. Dezember 2014

[EJP11] EIKEL, B.; JÄHN, C.; PETRING, R.: „PADrend: Platform for Algorithm Development and Rendering“. In: Augmented & Virtual Reality in der Produktentstehung. Hrsg. von J. Gausemeier, M. Grafe und F. Meyer auf der Heide. Bd. 295. HNI-Verlagsschriftenreihe. Heinz Nixdorf Institut, Universität Paderborn, Mai 2011, S. 159–170.

[FMS12] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A Practical Guide to SysML – The Systems Modeling Language. Morgan Kaufmann, Waltham, 2. Auflage, 2012

[GFD+08] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus. In: Konstruktion, Ausgabe 7/8-2008 und 9/-2008, Springer VDI-Verlag, Düsseldorf, 2008

[GLL12] GAUSEMEIER, J.; LANZA, G.; LINDEMANN, U.: Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung. Hanser Verlag, München, 2012

[GTS14] GAUSEMEIER, J.; TRÄCHTLER, A.; SCHÄFER, W.: Semantische Technologien im Entwurf mechatronischer Systeme – Effektiver Austausch von Lösungswissen in Branchenwertschöpfungsketten. Hanser Verlag, München, 2014

[INC13] INCOSE: Systems Engineering Vision 2020. unter: http://www.incose.org/Products_Pubs/pdf/SEVision2020_20071003_v2_03.pdf, 2007, abgerufen am 20. August 2013

Autoren

Dr.-Ing. Jan Berssenbrügge ist wissenschaftlicher Mitarbeiter in der Fachgruppe Produktentstehung am Heinz Nixdorf Institut der Universität Paderborn. Seine Forschungsschwerpunkte liegen in den Bereichen Virtual & Augmented Reality, Virtual Engineering sowie Virtual Prototyping und Simulation. Herr Berssenbrügge studierte Informatik an der Universität Paderborn und promovierte am Heinz Nixdorf Institut mit einem Thema zum Virtual Prototyping von Scheinwerfersystemen.

Dipl. Wirt.-Ing. Olga Wiederkehr studierte Wirtschaftsingenieurwesen mit der Fachrichtung Innovations- und Entwicklungsmanagement an der Universität Paderborn. Nach Abschluss ihres Studiums war Frau Wiederkehr Beraterin bei der UNITY AG, einer technologieorientierten Unternehmensberatung. Seit 2012 ist sie wissenschaftliche Mitarbeiterin am Heinz Nixdorf Institut der Universität Paderborn bei Prof. Dr.-Ing. Jürgen Gausemeier. Ihre Tätigkeitsschwerpunkte liegen in den Bereichen Systems Engineering sowie Entwicklungsmethodik Mechatronik.

Dipl.-Inform. Claudius Jähn studierte Informatik an der Universität Paderborn und ist wissenschaftlicher Mitarbeiter in der Fachgruppe Algorithmen und Komplexität am Heinz Nixdorf Institut. Seine Forschungsinteressen liegen im Bereich Computergrafik bei der Darstellung hochkomplexer Szenen und der Analyse von Szeneneigenschaften.

Dr. rer. nat. Matthias Fischer studierte Informatik an der Universität Paderborn und ist wissenschaftlicher Mitarbeiter in der Fachgruppe Algorithmen und Komplexität am Heinz Nixdorf Institut. Er promovierte dort über verteilte virtuelle Szenen. Seine Forschungsinteressen liegen im Bereich Computergrafik, Echtzeit-Renderingalgorithmen und verteiltes Rechnen.

Visualisierung von Anforderungen an physische Prototypen mit Augmented Reality

Kevin Wrasse, Elisabeth Brandenburg, Dr. Haygazun Hayka, Prof. Dr.-Ing. Rainer Stark
Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik (IPK)
Pascalstr. 8-9, 10587 Berlin
Tel. +49 (0) 30 / 39 00 62 18, Fax. +49 (0) 30 / 39 30 246
E-Mail: kevin.robert.wrasse@ipk.fraunhofer.de

Prof. Dr.-Ing. Rainer Stark
Fachgebiet Industrielle Informationstechnik (IIT),
Technische Universität Berlin
Pascalstr. 8-9, 10587 Berlin
Tel. +49 (0) 30 / 31 42 54 14
E-Mail: rainer.stark@tu-berlin.de

Zusammenfassung

Tests an physischen Prototypen sind in der Produktentwicklung ein wesentlicher Bestandteil um Bauteile oder Baugruppen gegen Fehlfunktionen und zugunsten ihrer Anforderungskonformität abzusichern. Betrachtet man nun den Fall einer Absicherung an einem physischen Prototyp, so ist die Beziehung eines Bauteils zu den zugrundeliegenden Anforderungen nicht explizit an dem Prototyp sichtbar. Der Entwickler ist durch den Medienbruch mit einem erheblichen Aufwand konfrontiert, die Beziehungen zwischen den physischen Bauteilen und den textuellen Anforderungen herzustellen. Dieser zusätzliche Aufwand kann durch den Einsatz von Augmented Reality (AR) verringert werden. Dazu werden die vorhandenen Zusammenhänge zwischen Anforderung, Geometriemodell und dem physisches Bauteil so verwendet, dass die Beziehungen eines Bauteils oder einer Baugruppe zu den Anforderungen in das Sichtfeld des Entwicklers über ein mobiles Endgeräteingeblendet werden können. Im Rahmen dieses Beitrags wird die Untersuchung der Machbarkeit dieser Lösung beschrieben. Dafür wurden die Komponenten spezifiziert, konzipiert und prototypisch umgesetzt, die zu einer erfolgreichen Lösung des Problems beitragen.

Schlüsselworte

Augmented Reality, Traceability, Virtuelle Absicherung, Requirements Engineering, Produktentstehung

Visualization of Requirements on Physical Prototypes with Augmented Reality

Abstract

Tests on physical prototypes are an important part within product development because they ensure verification of components or assemblies against malfunctions and hence support their requirement conformity. In case of product verification with a physical prototype, the relationship between components and underlying requirements cannot be seen explicitly on the prototype. The develop engineer is confronted with significant efforts in order to rebuild the relationship between the physical prototype and the textual requirements due to media discontinuity. This additional effort can be decreased by use of Augmented Reality (AR). Therefore, the existing context between requirements, geometrical model and physical component are used in order to display the relationship of a component or assembly to its requirements. The engineering perspective is ensured by the appropriate assistance of mobile devices. In this paper, it is described how the feasibility of this solution can be explored. The components are specified, designed and implemented as prototypes which contribute to solve the problem successfully.

Keywords

Augmented Reality, Traceability, Virtual Validation, Requirements Engineering, Product Design

1 Einleitung

Eines der ersten Ergebnisse im Produktentwicklungsprozess ist die Anforderungsspezifikation. Gegenstand der Anforderung sind Eigenschaften, Funktionalitäten und Qualitäten, die ein Produkt bekommen soll. Mit diesen wird beschrieben, was genau von einem Produkt hinsichtlich der Beschaffenheit, der Funktion, der Nutzbarkeit etc. erwartet wird. Für eine erfolgreiche Nutzung der Anforderungen ist ein Anforderungsmanagement erforderlich, welches dabei hilft diese schriftlich zu dokumentieren, zu strukturieren und für die Verwendung in den nachfolgenden Entwicklungsphasen nutzbar zu machen [Gra11]. Insbesondere bei komplexer werdenden Produktstrukturen bis hin zu domänenübergreifenden Systemen, deren Komplexität vorzugsweise mit den Methoden des Model-Based Systems Engineering (MBSE) [Est07] bearbeitet wird, kommt dem Anforderungsmanagement eine besondere Aufgabe zu. Die Anforderungen werden sowohl von den unterschiedlichen Domänen im Verlauf des Entwicklungszyklus herangezogen, um die Systemmodelle in ihren fortschreitenden Detaillierungsgraden und die Prototypen abzusichern als auch bei der Verifikation des daraus resultierenden physischen Prototypen. In diesem Beitrag wird ein Anwendungsfall betrachtet, bei welchem die Anforderungen an einen physischen Prototyp abgesichert werden. Dabei werden die anhand der Anforderungen definierten Test-Cases abgearbeitet und entsprechend der Ergebnisse bewertet. In diesem systematischen Ablauf können jedoch durch die direkte Interaktion und durch das Erleben des physischen Prototyps Erkenntnisse entstehen, welche keinem Test Case zugeordnet sind, und somit keinen Anforderungen explizit zugewiesen sind. Im Besonderen sind dabei die Anforderungen relevant, welche für Artefakte der mechanischen Domäne entstehen, die vor der Integration in das Gesamtsystem verifiziert werden müssen.

2 Anforderungen in der Produktentstehung

Die Anforderungen an das zukünftige Produkt sind das erste Produktartefakt, welches im Smart Systems Engineering Entwicklungsprozess entsteht und dient als Grundlage für die weitere Produktentstehung. Anforderungen sollten stets schriftlich verfasst werden und ggf. hierarchisch strukturiert sein. In einem Dokument oder Datenbank hinterlegt, sind sie ein dynamisches Werkzeug für die anforderungskonforme Entwicklung [Gra11].

2.1 Smart Systems Engineering

Die Entwicklung von komplexen Produktsystemen wird durch die Verwendung von Vorgehensmodellen vereinfacht. Ein Beispiel hierfür ist das V-Modell des Smart Systems Engineering (eine Ausprägung des MBSE), welches in Bild 1 dargestellt ist. Die Erweiterung des Systems Engineering zum Smart Systems Engineering beinhaltet eine erweiterte Sicht auf die Absicherung der Entwicklungsartefakte. Hier können virtuelle,

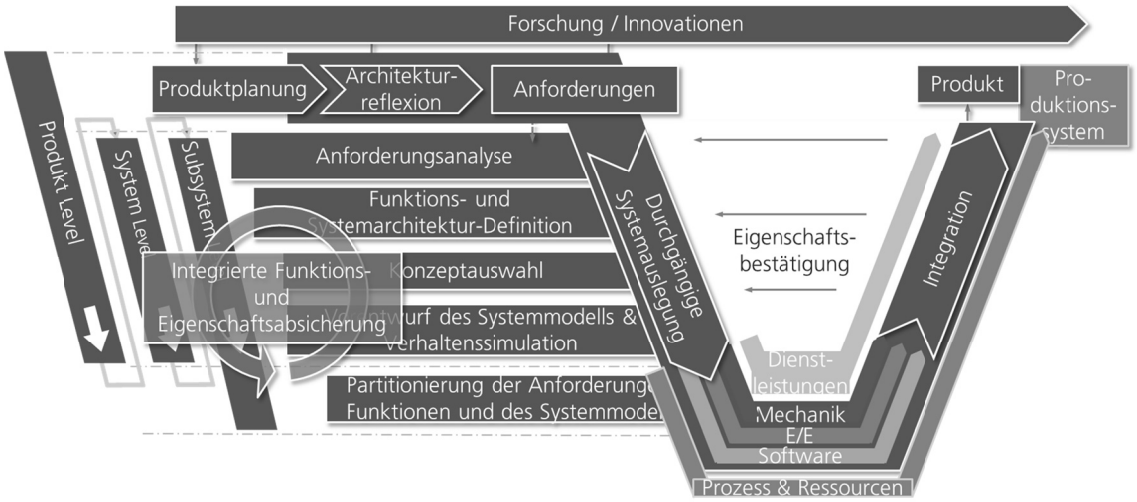


Bild 1: Erweitertes V-Modell des Smart Systems Engineering nach R. Stark [BRW+12]

sowie physische Absicherungsplattformen, als auch Kombinationen von Beiden zur Absicherung verwendet werden. Nach den Phasen der Produktplanung und Architekturreflexion werden die Anforderungen formuliert und entsprechend den Methoden des Anforderungsmanagements in ein Ordnungsschema gebracht. Im Anschluss daran werden die funktionalen Zusammenhänge des Gesamtsystems definiert. Hier kann die Beziehung zu den Anforderungen teilweise durch direkte Ableitung der Funktionen aus den Anforderungen hergestellt werden. Weitere Beziehungen werden mittelbar über die Zusammenhänge in funktionalen Teilsystemen erstellt. Wenn die domänenspezifischen Teillösungen erarbeitet wurden, beginnt deren Absicherung anhand der vorausgehenden virtuellen Produktmodelle. Schließlich werden alle Anforderungen verifiziert, die mit den Komponenten der Teilsysteme zusammenhängen [BRW+12]. Um die Zusammenhänge aller Teilkomponenten überprüfen zu können wird eine Nachverfolgbarkeit der Anforderungen über die weiteren Produktartefakte vorausgesetzt. Daher wird im Folgenden die Nachverfolgbarkeit (Traceability) beschrieben.

2.2 Requirement-Traceability

Diese Fähigkeit, Beziehungen zwischen den Objekten der einzelnen Produktmodelle bis zurück zu den Anforderungen vorzuhalten, wird Traceability genannt [Bei14], [Fig14]. Es ist eine Methode des Requirements Engineering und hilft dem Entwickler bei der Verifikation und Validierung der Anforderungen. Die Verbindungselemente, im Folgenden Tracelinks genannt, bilden die Beziehung zwischen den Elementen der aufeinanderfolgenden Produktmodelle ab. Die Verbindung werden zwischen den Anforderungen, den Funktionen im Funktionsmodell, dem Verhaltensmodell und den Teilen der Produktstruktur hergestellt. Über diese Kette können von einem Bauteil oder einer Baugruppe aus allen vorangegangenen Entwicklungsmodellen alle Elemente identifiziert werden, die einen Einfluss auf dessen Eigenschaften haben. Zum einen werden die Anforderungen verwendet, um die Modelle abzusichern und entsprechend Fehler zu ver-

meiden (Vorwärtstraceability). Zum anderen werden die Zusammenhänge genutzt, um bei Fehlern die Anforderungen ausfindig zu machen, welche nicht berücksichtigt, fehlerhaft oder fehlend sind (Rückwärtstraceability) [GF94]. Die Prüfung von Anforderungen erfolgt u. a. durch Prototypen, neben Inspektionen, Walkthroughs und Checklisten [Gra11]. Und diese Kombination aus Prototyp und Rückwärtstraceability soll in dieser Entwicklung genutzt werden, um nicht Test-Case spezifische Erkenntnisse nutzbar zu machen.

2.3 Absicherung von Anforderungen am physischen Prototyp

Ein Prototyp ist eine Annäherung an ein Produkt unter Betrachtung einer oder mehrerer Dimensionen des Interesses, das heißt es werden nur Teile aller Anforderungen umgesetzt. Der Vorteil ist, dass durch das „Arbeiten“ am Produkt, Aussehen, Funktion und Verhalten des Produktes erlebt werden können [Ulr12], [ALS13]. Zusätzlich entsteht bei der Verwendung von physischen Prototypen der Effekt, dass durch diese insbesondere unerwartete Eigenschaften des Produkts zum Vorschein kommen. Während virtuelle Prototypen sehr zielgerichtet zur Untersuchung einer oder mehrerer definierter Eigenschaften erstellt werden, so wirkt beim physischen Prototyp die Gesamtheit der physikalischen Umgebungseinflüsse, wodurch auch vorher unbeachtete Effekte zum Vorschein treten. Darüber hinaus kann durch einen physischen Prototyp ein größeres Spektrum der menschlichen Sinneswahrnehmung angesprochen werden, wodurch auch mögliche implizite Kundenanforderungen besser abgesichert werden können. In einem Projekt am Fachgebiet Industrielle Informationstechnik der Technischen Universität Berlin wurde das Smart-Tripelec, ein dreirädriges elektrisches Fahrrad, entwickelt und prototypisch produziert. An diesem physischen Prototyp des Tripelec wurden nun in ersten Tests funktionale Probleme deutlich. Die Fehler waren eine schwergängige Antriebskette und eine klemmende Lenkung. Um diese Fehler vollständig zu beheben, ist es notwendig alle Anforderungen, die zu der funktionalen Gruppe gehören, zu identifizieren. In einer Werkstattumgebung ist es nicht immer möglich auf kurzem Wege alle Entwicklungsdaten abzurufen.

2.4 Problemstellung und Lösungsansatz

Die Absicherung an physischen Prototypen ist ein erforderlicher Prozess, und die Einhaltung der Anforderungen ist notwendig für ein valides Produkt. Die Schwierigkeit des Prüfers besteht jedoch darin, dass an einem physischen Prototyp die Beziehung eines Bauteils zu den zugrundeliegenden Anforderungen nicht immer explizit sichtbar ist. Der

Prüfer ist durch den Medienbruch mit einem erheblichen Aufwand konfrontiert, die Beziehungen zwischen den physischen Bauteilen und den textuellen Anforderungen herzustellen. Hier wäre also ein Ansatz sinnvoll, welcher die beiden Informationen in ein Blickfeld integriert. Außerdem ist die Rückverfolgung der Verbindungen zwischen den Produktmodellen nur in einem Anforderungsmanagement-Tool möglich (bspw. Doors).

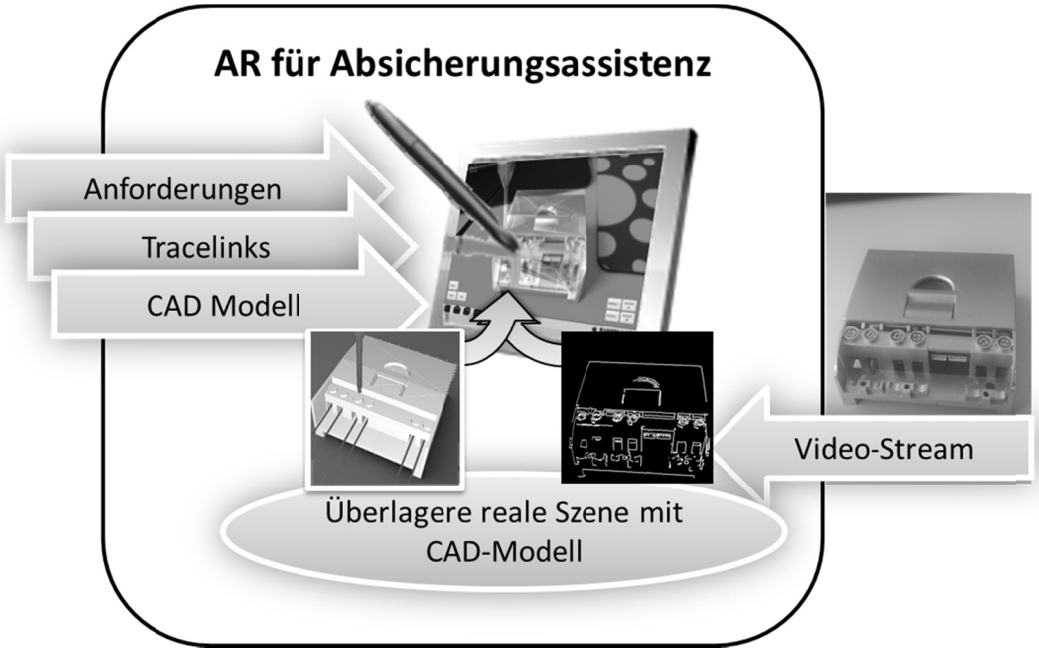


Bild 2: Übersicht der Daten und Prozesse für die Anforderungsvisualisierung

Es erfordert viel Erfahrung, diese Tools zu bedienen und kostet Zeit die Informationen zu transferieren. Hier wäre folglich eine Software sinnvoll, die die Traceabilityinformationen direkt in den Montage- und Untersuchungsbereich für den Prototyp bringt. Der Lösungsansatz ist die Verwendung von Augmented Reality (AR), mit deren Hilfe textuelle Informationen (die Anforderungen) an einem realen Objekt (der physische Prototyp) visualisiert und bearbeitet werden können. Im Folgenden wird der Lösungsansatz und AR genauer beschrieben.

3 Visualisierung von Anforderungen an physischem Prototyp

Für eine Vereinfachung der Validation von Anforderungen am Prototyp ist es also sinnvoll, diese über eine visuelle Schnittstelle miteinander zu verknüpfen. Als visuelle Schnittestelle eignet sich ein Tablet-PC, welches über eine Kamera verfügt, mit der das reale Objekt detektiert werden kann. Weiterhin müssen die erforderlichen Daten identifiziert und deren Verwaltung geklärt werden. Die Zusammenhänge der Daten und Prozesse, die in der Anwendung verarbeitet werden, sind in einer Übersicht in Bild 2 dargestellt. Benötigt werden die Anforderungen an das Produkt und die Verbindungen zwischen diesen Anforderungen und den Bauteilen, welche am physischen Prototypen sichtbar sind. Auf dieser Datengrundlage muss eine Gestaltungslösung konzipiert werden, welche folgende Anzeigen enthält:

- Realer physischer Prototyp,
- CAD-Modell, welches eine Unterscheidung der Baugruppen/ Bauteile zulässt,
- Produktanforderungen und
- Verbindungen von einem Bauteil zu den entsprechenden Anforderungen.

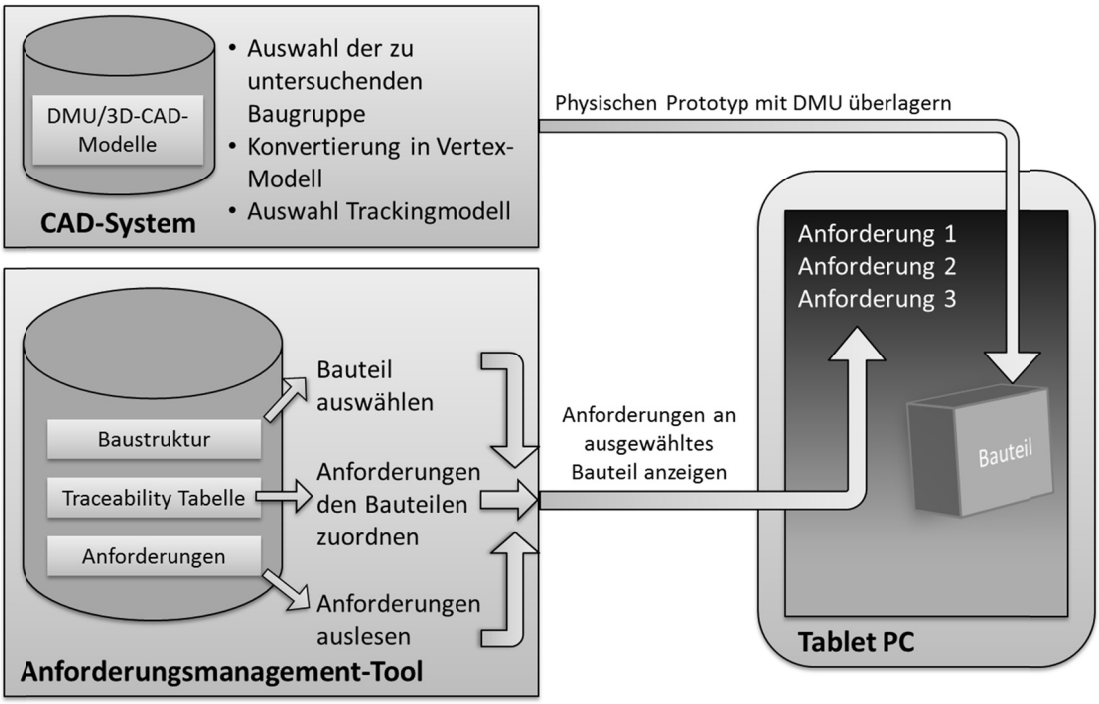


Bild 3: Konzept der Datenflüsse für die Anforderungsvisualisierung auf dem Tablet-PC

Dem Entwickler sollen bei der Absicherung am physischen Prototyp nicht nur die relevanten Daten übersichtlich dargestellt werden, sondern er soll auch in Form von Markierungen oder Texteingaben Änderungen in den Anforderungen vermerken können. Dies ist notwendig, um im nächsten Iterationsschritt der Überarbeitung die Anforderungen anzupassen und Fehlfunktionen zu beheben. Um die in diesem Lösungsansatz enthaltenden Komponenten zu ermöglichen wird ein entsprechend ein Tracking des realen Objekts, CAD-Daten, Anforderungsdaten, Verbindungsdaten (Tracelinks) und Gestaltungsrichtlinien für AR-Anwendungen benötigt. Das Datenmodell enthält neben diesen DMU/ CAD-Daten, auch die Anforderungen und Tracelinks. Die Anforderungen werden je nach IT-Lösung aus dem entsprechenden Anforderungsmanagement-tool extrahiert. Die Tracelinks zwischen Anforderungen und CAD-Daten werden auch aus der entsprechenden Datenbank in das Datenmodell eingefügt. Diese Abhängigkeiten werden von dem AR-Werkzeug aus dem Datenmodell herausgelesen und mit den dargestellten 3D-CAD-Modellen verknüpft. Eine Übersicht zu den Datenflüssen ist in Bild 3 zusehen.

3.1 Gestaltungsrichtlinien für AR-Informationen

Die Visualisierung der Informationen sollte neben dem einfachen Verständnis und der Lesbarkeit für den Betrachter, auch zielgerichtet auf die zu erfüllende Aufgabe sein. Dementsprechend variieren Gestaltungsrichtlinien je nach Medium (Monitor, Printmedien, VR oder AR), Anwender und dem Nutzungskontext (DIN EN ISO 9241 - 11). Diese Betrachtung von Mensch, Technologie und Aufgabe ist der Forschungsschwer-

punkt der Gestaltung von User Interfaces (UI). Die Gestaltung von AR-UIs muss neben allgemeinen Gestaltungsparametern die folgenden spezifischen AR-Parameter beachten (aus [Weg12]):

- Redundanz: Redundanz liegt dann vor, wenn die Informationen auf dem AR-Medium exakt den dahinterliegenden realen Komponenten entsprechen. Es sollten Redundanzen vermieden werden. Wenn redundante Visualisierung verwendet wird, dann sollten die realen Komponenten sichtbar sein.
- Realismus und Ähnlichkeit: Die dargestellten Informationen können sowohl abstrakt als auch realistisch abgebildet werden. Die Parameter Größe, Konturform und Farbe sollten nicht realistisch wiederholt werden, da ähnliche Darstellungen die Detekti
- onsleitung vermindern. Eine Restähnlichkeit sollte jedoch zur Bezugsbildung erhalten bleiben.
- Gruppierung: Gruppierung meint niedrige Distanzen zwischen den AR-Informationen untereinander. Diese Bündelung sollte vermieden werden [AH05].
- Dimensionalität: Die Information kann zwei- oder dreidimensional dargestellt werden. Bei der dreidimensionalen Visualisierung kann zusätzlich die Tiefenposition verändert werden. Beide Dimensionsparameter Form und Position sollten dem realen Pendant entsprechen.
- Transparenz: Dargestellte AR-Informationen können durchsichtig sein. Dies ermöglicht die Sichtbarkeit der realen Komponente.
- Konforme und nonkonforme Aufblendung: Bei der konformen Aufblendung liegen AR- und reale Informationen übereinander, bei der nonkonformen danebenstehend. Diese Positionierung, die die Sichtbarkeit der realen Komponente ermöglicht, ist zu bevorzugen, sollte jedoch mit einer Verbindungslinie unterstützt.

Diese Gestaltungsrichtlinien werden in der AR-Anwendung umgesetzt werden.

3.2 Implementierung der AR-Anwendung

Für virtuelle Überlagerung von Informationen an realen Objekten im Rahmen von Augmented Reality gibt es verschiedene Möglichkeiten. Die einfachste Möglichkeit ist das markerbasierte Tracking, wobei ein zweidimensionales Muster von der Kamera erkannt und dessen Pose im dreidimensionalen Raum berechnet wird. Dafür ist es erforderlich, dass der Marker einen definierten Abstand zu dem physischen Objekt einhält und sich die gesamte Zeit des Trackings vollständig im Kamerabild befindet. Dies kann insbesondere bei engen Bauräumen kompliziert werden, so dass hier nach einer Alternative gesucht werden muss. Das in einem CAD-System entworfene Geometriemodell lässt sich in einer komprimierten Form als Digital Mock-Up (DMU) speichern. Mit dem

kantenbasierten Trackingverfahren „RAPiD“ ist es möglich, dreidimensionale Objekte ausgehend von diesem DMU zu tracken [HS90]. Dieses Verfahren wird hier verwendet.

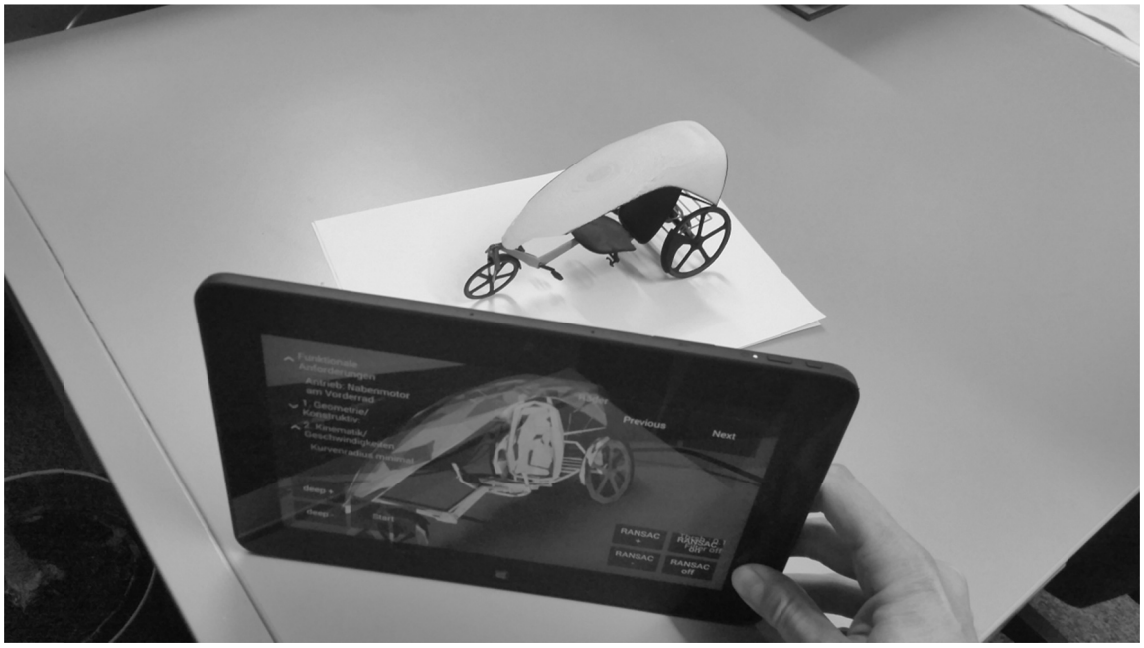


Bild 4: Nutzung eines Tablet-PCs zur Visualisierung der Anforderungen

Als Plattform für diese Software dient ein Tablet-PC mit einem Android-Betriebssystem. Der Trackingalgorithmus RAPiD wurde im Rahmen eines vorherigen Projekts implementiert, so dass hier vor allem die Konvertierung und Darstellung der 3D-Modelle und Anforderungen, sowie die Konzeption der Datenflüsse im Vordergrund stand. Ziel der Anwendung ist es, ein DMU-Modell über einen Videostream genau zu positionieren, so dass eine realistisch erscheinende 3D-Beziehung zwischen dem realen und virtuellen Modell hergestellt werden kann (Gestaltungsrichtlinie *Realismus)*. Dafür werden alle zu betrachtenden virtuellen 3D-Modelle der Bauteile in ein darstellbares Format konvertiert, so dass die räumliche Zuordnung der Baustruktur erhalten bleibt. Zusätzlich werden eine eindeutige Beziehung zwischen den Bezeichnungen der 3D-Modelle der Bauteile und den Einträgen in der Liste der Baustruktur hergestellt. Bei der Anwendung hält der Nutzer das mobile Endgerät so vor die zu betrachtende Baugruppe, dass dessen 3D-Modell und Videobild eine etwa gleiche Pose einnehmen, wie es in Bild 4 dargestellt ist. Nun kann die Baugruppe getrackt werden, so dass ein virtuelles und ein reales Bauteil immer deckungsgleich übereinander liegen. Durch die Bewegung des Tablet-PCs relativ zum Prototyp navigiert der Nutzer intuitiv zu dem zu untersuchenden Bauteil. Die Gestaltungsrichtlinie zur *Redundanz* war bei der Überblendung des Prototyps mit dem DMU zu beachten, wobei hier eine teilweise Redundanz zur Identifikation des Bauteils notwendig war, welche aber durch die Verwendung von *Transparenz* abschwächt wurde. Die Baustruktur der aktuell im Bildschirm befindlichen Baugruppe wird im Demonstrator über zwei Navigationsknöpfe (obere rechte Ecke) durchlaufen und das ausgewählte Bauteil farblich hervorgehoben. So kann das untersuchte Bauteil des physischen Prototyps ausgewählt werden. Aus der Traceability-

Tabelle werden die zugehörigen Anforderungen geladen und in der linken Bildschirmseite verkürzt angezeigt (*nonkonform, 2-dimensional).*

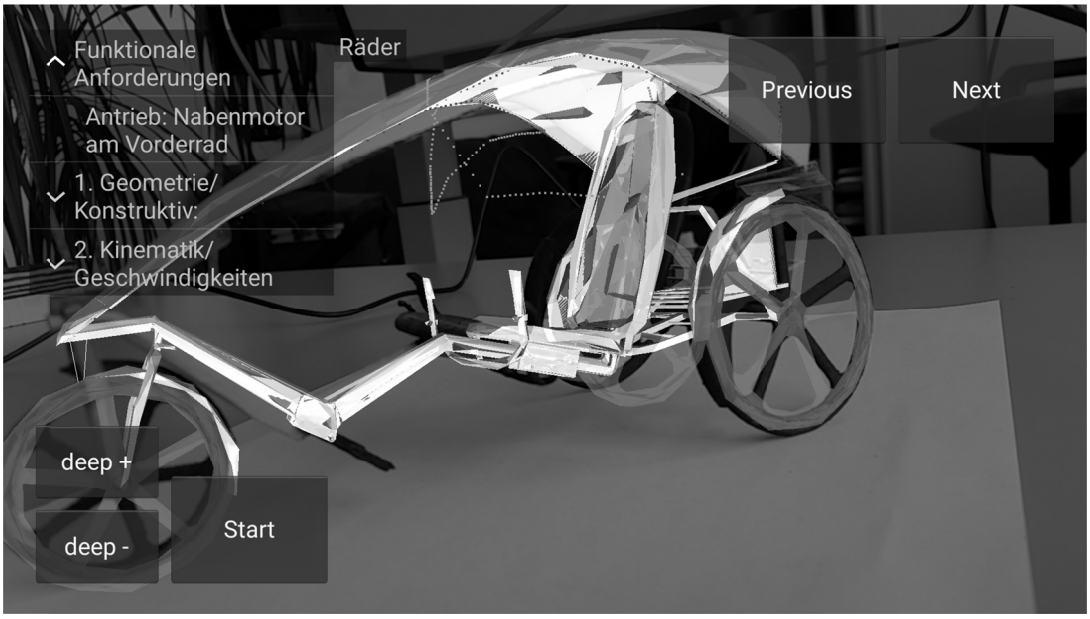


Bild 5: Screenshot der Anwendung bei Anzeige der Anforderungen ans Smart-Tripelec

Die Anforderungen können nun ausgewählt und kommentiert werden. Die Kommentare werden in einer Liste zusammen mit den Links zu den bezogenen Anforderungen gespeichert. Nach Abschluss des Absicherungssprozesses kann die Liste in Form einer Text-Datei zur Weiterverarbeitung auf den Rechner geladen werden, wo die Anforderungen im Anforderungsmanagementsystem geändert werden sollen. Bild 5 zeigt die Umsetzung der Anforderungsvisualisierung bei der Validation eines Prototyps.

4 Resümee und Ausblick

Im Rahmen dieser Forschungsarbeit konnte ein neues Konzept für die Visualisierung von Anforderungen prototypisch umgesetzt werden. Damit ist es möglich ein Bauteil eines physischen Prototyps zu identifizieren und die damit verbundenen Anforderungen anzuzeigen. Die Anforderungen sind mit dieser Anwendung ohne Medienbrüche bei der Untersuchung eines Prototyps verfügbar und helfen den Absicherungsprozess effizienter durchzuführen. An dieser Stelle muss die Ausarbeitung der Anzeige der Anforderungen für eine gute Lesbarkeit weiter verfeinert werden, da Anforderungen häufig in Form von langen Sätzen formuliert werden und so die wesentlichen Daten nicht immer sofort erkennbar sind. Zusätzlich kann nach Alternativen für die Lösung des Trackingalgorithmus gesucht werden, da dieser zwar die wesentlichen Anforderungen erfüllt, jedoch im Betrieb unter bestimmten Voraussetzungen instabil ist und eine Überlagerung bei ungünstigen Lichtverhältnissen schwierig ist.

Die nächsten Schritte beinhalten eine Evaluation der Machbarkeit einer Übertragung der Lösung auf AR-Brillen. Diese würden den Vorteil bieten, dass bei der Untersuchung die Hände des Akteurs frei sind, jedoch sind viele Randbedingungen in Betracht auf die Überlagerung von realen 3D-Objekten mit virtuellen Inhalten unter Beachtung der Einschränkungen von Optical-See-Through-Geräten zu berücksichtigen und zu erfüllen.

Literatur

[AH05] ARAGON, C. R.; HEARST, M. A.: Improving aviation safety with information visualization. In (van der Veer, G.; Gale, C. Hrsg.): the SIGCHI conference, 2005; S. 441.

[ALS13] AURICHT, M.; LINDOW, K.; STARK, R.: Erlebbare Entwicklung und Absicherung technischer Systeme. Stuttgarter Symposium für Produktentwicklung 2013, 2013.

[Bei14] BEIER, G.: Verwendung von Traceability-Modellen zur Unterstützung der Entwicklung technischer Systeme. Fraunhofer-Verl, Stuttgart, 2014.

[BRW+12] BEIER, G. et al.: Durchgängige Entwicklung mit erlebbaren Prototypen. Modellbasiertes Systems Engineering. In Digital Engineering, 2012, 11; S. 14-17.

[Est07] ESTEFAN, J. A.: Survey of Model-Based Systems Engineering (MBSE) Methodologies, 2007.

[Fig14] FIGGE, A.: Effiziente Erfassung und Pflege von Traceability-Modellen zur Entwicklung technischer Systeme. Fraunhofer Verlag, Stuttgart, 2014.

[GF94] GOTEL, O.; FINKELSTEIN, C. W.: An analysis of the requirements traceability problem: IEEE International Conference on Requirements Engineering, 1994; S. 94-101.

[Gra11] GRANDE, M.: 100 Minuten für Anforderungsmanagement. Kompaktes Wissen nicht nur für Projektleiter und Entwickler. Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden, Wiesbaden, 2011.

[HS90] HARRIS, C.; STENNET, C.: RAPID – A Video Rate Object Tracker, Romsey, Hampshire, England, 1990.

[Ulr12] ULRICH, K. T.: Product design and development. McGraw-Hill, New York, NY, 2012.

[Weg12] WEGERICH, A.: Nutzergerechte Informationsvisualisierung für Augmented Reality Anzeigen, Berlin, 2012.

Autoren

Kevin Wrasse ist seit 2013 Wissenschaftlicher Mitarbeiter des Fraunhofer Instituts für Produktionsanlagen und Konstruktionstechnik in Berlin. In der Abteilung Virtuelle Produktentstehung beschäftigt er sich mit den Forschungsschwerpunkten Product-Service Systems und Remote Monitoring zur Anwendung in der dezentralen Energieversorgung, sowie Informationsvisualisierung in der Produktentwicklung. Seinen Abschluss als Dipl.-Ing. hat er 2013 im Studiengang Informationstechnik im Maschinenwesen an der Technischen Universität Berlin gemacht.

Elisabeth Brandenburg ist seit 2014 Wissenschaftliche Mitarbeiterin des Fraunhofer Instituts für Produktionsanlagen und Konstruktionstechnik in Berlin. In der Abteilung Virtuelle Produktentstehung beschäftigt sie sich mit den Forschungsschwerpunkten

Human Factors im Model-Based Systems Engineering und Traceability. In den Jahren zuvor forschte sie an dem interdisziplinären Graduiertenkolleg prometei zu Informationsvisualisierung in immersiver Virtual Reality. Sie erwarb den Bachelor of Science in Informations- und Medientechnik und den Master of Science in Human Factors.

Dr.-Ing Haygazun Hayka ist Leiter der Abteilung Informations- und Prozesssteuerung (IPS) am Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik (IPK). Er studierte Elektrotechnik an der Technischen Universität Berlin, wo er auch promovierte. Seine Arbeitsschwerpunkte liegenin den Bereichen PDM/ PLM, Collaborative Engineering sowie Informationsmanagement in mechatronischer Systementwicklung.

Prof. Dr.-Ing. Rainer Stark ist seit 2008 Leiter des Fachgebietes Industrielle Informationstechnik der Technischen Universität Berlin und Direktor des Geschäftsfeldes Virtuelle Produktentstehung des Fraunhofer-Instituts für Produktionsanlagen und Konstruktionstechnik (IPK). Er studierte Maschinenbau an der Ruhr-Universität Bochum sowie der Texas A&M University (USA). Von 1989 bis 1994 war er Wissenschaftlicher Mitarbeiter am Lehrstuhl für Konstruktionstechnik/CAD der Universität des Saarlandes. Nach Erlangung des Grades Dr.-Ing. füllte er verschiedene Führungspositionen in der Automobilindustrie aus. Forschungsschwerpunkte sind die intuitive und kontextbezogene Informationsmodellierung, intuitiv bedienbare und funktional erlebbare virtuelle Prototypen, die funktionsorientierte virtuelle Produktentstehung sowie Entwicklungsprozesse und Methodiken für die Produktgestaltung.

Rendering und 3D-Produktdarstellung

Depth Image Based Rendering zur Entkopplung langsamer Renderprozesse von der Darstellung

Julian Meder
3Dinteractive GmbH
Am Vogelherd 10, 98693 Ilmenau
Tel. +49 (0) 3677 / 79 95 50, Fax. +49 (0) 3677 / 79 95 511
E-Mail: jmeder@3dinteractive.de

Zusammenfassung

Seit Anbeginn der Computergrafik verhindern Hardwarebeschränkungen interaktive Bildraten bei der Darstellung komplexer 3D-Szenen. Zwar steigt die Hardwareleistung mit der Zeit kontinuierlich, die Größe der Szenen allerdings ebenfalls. Existierende Lösungen dieses Problems, wie Visibility Guided Rendering, zeigen je nach Systemleistung und angefordertem Szenendetailgrad für die Anforderungen des Anwendungsfalls zu niedrige oder szenenabhängig schwankende Bildraten.

Diese Arbeit konzentriert sich daher auf die Erforschung einer Bilderzeugungstechnik, welche möglichst unabhängig von der Szenenkomplexität eine im Vergleich zum klassischen Rendering höhere Bildrate garantiert und frei von Bildratenschwankungen ist. Damit einhergehend ist eine geringe Reaktionszeit auf Nutzereingaben (Kamerabewegungen möglichst sofort ersichtlich). Die dazu verwendete Technik wird in der Literatur als Depth Image Based Rendering (DIBR) bezeichnet und basiert auf der Wiederverwendung des Tiefenbilds und des Farbbilds – beides Standardergebnisse heutiger Rasterisierungsverfahren – zur Erzeugung neuer Ansichtsbilder durch Projektion der Bildinformationen in neue Kameraeinstellungen. Eine Herausforderung stellen dabei in den klassisch gerenderten Bildern fehlende Szenenteile dar (Informationslücken), welche sich in den neuen Ansichtsbildern als Löcher äußern.

Im Detail betrachtet diese Arbeit einerseits Möglichkeiten der effizienten Umsetzung der DIBR Technik auf heutigen GPUs und andererseits die Behandlung von Informationslücken beziehungsweise der Reduktion dadurch entstehender Bildartefakte. Es wird gezeigt, dass diese Technik deutlich kostengünstiger als das klassische Rendering ist und damit hohe Bildraten bereitstellen kann. Löcher können bereits mit ebenfalls simplen Verfahren stark reduziert werden. Zusätzlich werden bestehende Limitationen des DIBR-Verfahrens sowie weitere Anwendungsgebiete außerhalb der Ein-Nutzer-Workstation, wie die Anwendung auf mobilen Geräten, diskutiert.

Schlüsselworte

GPU, Rendering, Depth, Image based, interaktive Bildraten

Depth Image Based Rendering for decoupling slow rendering processes from display

Abstract

Since the beginnings of computer graphics limited hardware capabilities fail to provide interactive frame rates of 3D-scenes. Hardware performance increases with time, but so does the amount of objects to display. Existing techniques to alleviate this problem, like Visibility Guided Rendering, still face problems such as for certain use cases unsuitable low frame rates or scene-dependent frame rate fluctuations.

This work therefor focuses on researching a rendering technique which is able to guarantee a non-fluctuating frame rate independent of scene complexity. In line with this requirement is a low response time to user inputs (camera movement should cause nearly instantaneous display updates). Literature calls the technique to achieve this Depth Image Based Rendering (DIBR). The basic idea constitutes of using the depth and color image – both common results of today's rasterization methods – to create new view images by projecting the picture contents to new camera views. A challenge is presented by scene parts invisible in the classically rendered images which cause holes in the new view images (information gaps).

In detail, possible efficient GPU-implementations of DIBR and methods for dealing with information gaps or rather the diminishing of resulting artifacts will be discussed. The technique presents itself as much cheaper than the classical rendering and can therefor provide high frame rates. Likewise, holes can be greatly reduced using simple methods. In addition, remaining limitations of the DIBR technique and further applications apart from the single-user-workstation, like mobile devices, are discussed.

Keywords

GPU, rendering, depth, image based, interactive frame rate

1 Einführung

Ein bestehendes Problem der interaktiven Computergrafik ist die Balancierung von dargestelltem Szenendetailgrad und der resultierenden Bildwiederholrate. Techniken zur dynamischen Reduktion jenes Detailgrads (Culling, Level-of-Detail) reichen bei entsprechender Szenengröße oder zu geringer Hardwareleistung für hohe Bildratenziele (60 Hz) nicht aus (vgl. Kapitel 5, Bild 7 Quellrenderer). Auch aufgrund szenenbedingter Detailveränderungen resultieren daraus letztendlich niedrige, schwankende Bildraten, die subjektiv als unangenehm empfunden werden können und produktivitätshindernd wirken oder Anforderungen professioneller Anwendungen unerfüllt lassen.

Ziel ist es daher, dem Anwender solch problematische Bildraten zu verbergen. Dazu ist einerseits eine höhere angezeigte Bildrate als die des eigentlichen Renderprozesses notwendig, andererseits ist eine zügige Reaktion der Anzeige auf Nutzereingaben erforderlich. Die dazu eingesetzte Kernmethode „Depth Image Based Rendering“ (DIBR) wird in Abschnitt 1.1 erläutert, während in Abschnitt 1.2 vergleichbare Arbeiten genannt werden. Kapitel 2 beschreibt geeignete Implementationen auf der Grafikkarte. Darauffolgend untersucht Kapitel 3 Probleme dieser Verfahren und erläutert die angewandte Lösung. Einen Eindruck der Verfahrensperformanz des in Kapitel 4 zusammengefassten Systems liefert Kapitel 5 und Kapitel 6 geht auf bestehende Limitationen des Verfahrens, weitere Anwendungsmöglichkeiten und künftige Arbeiten ein.

1.1 Depth Image Based Rendering

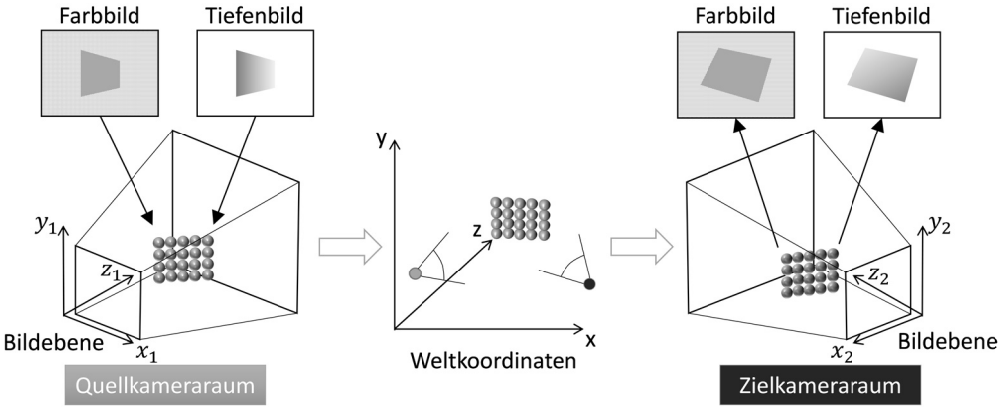


Bild 1: DIBR-Projektion von Farb- und Tiefenbild in andere Ansichten

Das DIBR Verfahren ist in der Literatur bereits um 1990 bekannt geworden, wobei es vorerst nicht unter diesem Namen, sondern 3D-Image-Warping [MM97] geführt wurde. Es geht von einem Farbbild und dem dazugehörigen Tiefenpuffer als Eingabe aus. Wie Bild 1 zeigt, definieren die Bildpixel bereits durch die jeweilige x-y-Bildposition und

die im Tiefenpuffer gespeicherte z-Koordinate eine 3D-Punktewolke im Bildraum der zugehörigen Kamera[1].

$$p_2 = P_2 R_2 T_2 (P_1 R_1 T_1)^{-1} p_1 \quad (1)$$

Gemäß Formel (1) müssen die vorhandenen Punkte lediglich durch Inversion der Bild- und Kameraraumtransformation zurück in den Weltraum und anschließend durch Anwendung der zur gewünschten neuen Ansicht gehörenden Transformationen in den neuen Bildraum transformiert werden. P_i ist dabei die Bildebenenprojektionsmatrix[2], R_i die Rotationsmatrix und T_i die Translationsmatrix des Kameraraums i sowie p_i die homogenen Koordinaten eines Punktes in Bildraum i.

1.2 Relevante Arbeiten

Eine der hauptsächlich anzutreffenden Anwendungsrichtungen des DIBR ist die Erzeugung von Pseudostereobildpaaren. Ein früher Ansatz von McMillan et al [MB95]. erzeugt diese für Head-Mounted-Displays, wobei er eine für diesen Fall effizientere Projektion verwendet. Der DIBR Begriff wurde erstmals von C. Fehn geprägt, der seinem „Shift-Sensor"-Algorithmus auf einer einfachen, lediglich in x-Richtung stattfindenden DIBR-Projektion aufbaut [Feh03]. Seine Idee und Grundtechnik wurde unter anderem von Zhang et al. [ZT05] und Ndjiki-Nya et al. [NKD+11] weiterverwendet. Letztere adaptierten Fehns Technik auf die für brillenloses 3D bei mehreren Betrachtern notwendige Erzeugung multipler Stereobildpaare aus einem einzigen gegebenen. Xi et al [XWY+13].und Oh et al [OYH09].approximieren eine inverse DIBR-Projektion von Ziel- zu Quellbild um Abtastungsfehler zu vermeiden.

Die andere Hauptanwendungsrichtung, in die sich auch diese Arbeit einordnet, liegt in der Synthese beliebiger Kameraansichten. Ansätze von Max et al. [MO95] Shade et al. [SGH+98]und Popescu et al. [PLA+98].modellieren die Szene in Form diskreter, offline mit hoher Qualität vorberechneter Quellbilder, aus denen eine über DIBR projizierte Untermenge ein gewünschtes Ansichtsbild liefert. Ebenfalls offline vorberechnet sind die Dreiecksnetzte von Darsa et al. [DCV97] und Fu et al. [FWH98] die die Szene durch eine hochauflösende Texturierung imitieren (auch als Impostoren [GFB13] bezeichnet). Ghiletiuc et al. [GFB13] berechnen solche Impostoren online auf einem Server und rendern diese auf mobilen Endgeräten. Dagegen senden Chang et al [CG02]. die Quellbilder direkt zu den Klienten, welche die komplette DIBR-Projektion ausführen.

Die Idee des DIBR ist sehr ähnlich der der Motion Compensated Frame Interpolation (MCFI) [LN10]. Beide Techniken zielen auf eine Bildsynthese aus den vorhandenen

---

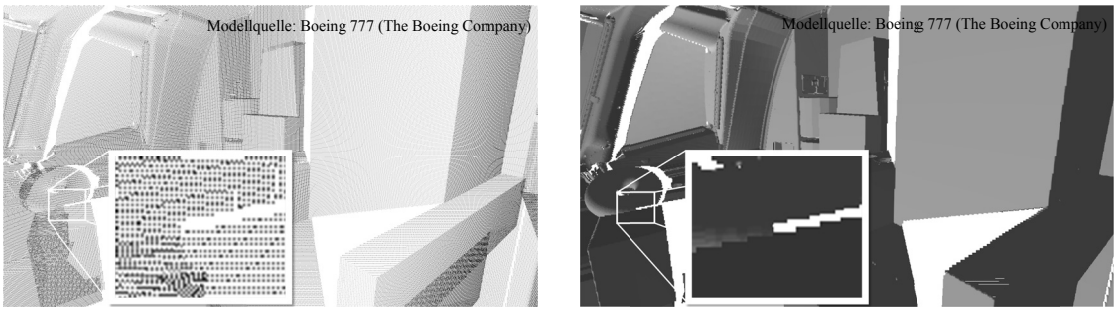
[1] Im Folgenden werden zur Projektion verwendete Daten wie Farb-, Tiefenbild und Kameraeinstellungen als Quelle bezeichnet bzw. mit dem Präfix Quell- versehen und Ergebnisse analog als Ziel.

[2] P_i ist hier eine in der Computergrafik gebräuchliche perspektivische Projektion, jedoch ist das Verfahren auch auf andere Projektionsarten anwendbar.

Echtbildern ab, allerdings ist MCFI auf zeitlich aufeinanderfolgende und korrekte Einzelbilder (bereits vorhandene Videos), ohne zusätzliches Tiefenbild, ausgerichtet. DIBR und MCFI können ineinander überführt werden [Med14].

2 DIBR auf der GPU

2.1 Punktbasiert



(a) (b)

Bild 2: Lücken (weiß) bei (a) reiner Punktprojektion, im Gegensatz zu (b) Splatting

Der einfache Ansatz aus Abschnitt 1.1, lediglich die durch die Inputbilder definierten 3D-Bildraumpunkte zu transformieren, zeigt schnell Probleme wie in Bild 2a: Die Abbildung zwischen den Bildpixeln des Quell- und Zielbilds ist oft nicht eineindeutig.

Eine Lösung dieses Problems stellt Splatting [Wes90] dar, dessen einfache Umsetzung mittels Punktrendering unter Variation der Punktgröße auf heutigen Grafikkarten möglich ist, allerdings werden die Pixel als Quadrat umgesetzt. Bessere Abschätzungen liefern nicht-uniforme Rekonstruktionskernel [Wes90], die allerdings schwieriger auf der GPU realisiert werden können. Inverses Mapping vom Ziel in den Quellbildraum (die bei Texturmapping favorisierte Lösung) kann hier aufgrund des fehlenden Zieltiefenbilds nicht ohne Weiteres angewendet werden.

2.2 Rein Bildbasiert

Das Projektionsverfahren von Morvan [Mor09] verwendet kein Primitivrendering, sondern reine Pixelabbildungen. Hier wird der Transformationsprozess aus Abschnitt1.1 in drei Schritte zerlegt.

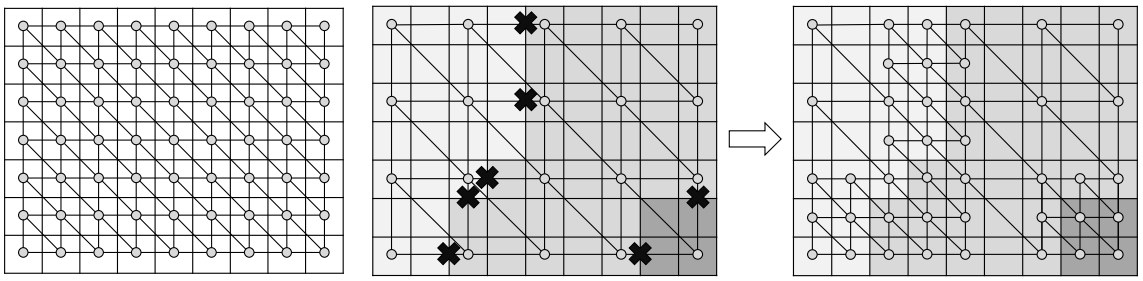
$$p_2 = P_2 R_2 R_1^{-1} P_1^{-1}\left(-z_w p_1 - P_1 R_1 (t_2 - t_1)\right) \qquad (2)$$

Abgeleitet wird der Algorithmus aus einer Umformung[3] von Formel (1) welche in Formel (2) gegeben ist. t_i bezeichnen die Translationsvektoren der Bildräume und z_w der Abstand von p_1 zu Kamera 1 im Weltkoordinatensystem[4]. Die Translationskomponente $-P_1 R_1 (t_2 - t_1)$ kann in einzelne x- und y-Verschiebungen zerlegt werden, was eine einfache zeilen- oder spaltenweise Interpolation zwischen zwei benachbarten Pixeln nach der Verschiebung ermöglicht [Mor09]. Lücken des vorigen Abschnitts werden so vermieden.

Die Umsetzung dieser beiden Translationsschritte ist nicht in der Standardgrafikpipeline möglich, da eine Projektion das Schreiben mehrerer Pixel in einem Schritt voraussetzt [Med14]. Die seit einiger Zeit existierenden General-Purpose-Computing Schnittstellen wie CUDA, DirectX Compute Shader oder OpenCL erlauben dies jedoch.

Als letzter Schritt wird das entstandene Zwischenbild mittels der verbleibenden Transformationen $P_2 R_2 R_1^{-1} P_1^{-1}$ in den Zielbildraum projiziert, was einer projektiven Texturabbildung entspricht und damit Standard-GPU-Texturfunktionen zulässt [Mor09].

2.3 Dreiecksbasiert



(a) (b)

Bild 3: (a) Vollständige Triangulation des Quellbilds, (b) Adaptive Triangulation anhand des Tiefenbilds, ausgehend von einem groben Dreiecksnetz (✱ detektierte Tiefenkante)

Die in dieser Arbeit erforschte Variante konstruiert aus den durch das Tiefenbild gegebenen 3D-Punkten ein Dreiecksnetz, welches mittels des Farbbilds texturiert wird. Bild 3a zeigt einen einfachen Fall (verwendet in [MM97]), bei dem jedes Bildpixel als Vertex modelliert wird und zwischen vier benachbarten jeweils zwei Dreiecke gebildet werden. Die Transformation aus Abschnitt 1.1 wird auf jedes Vertex angewendet und das Dreiecksnetz klassisch gerendert. Lücken werden durch die GPU-inhärente Interpolation der Vertexattribute vermieden.

---

[3] Eine vollständige Herleitung kann [Mor09] oder [Med14] entnommen werden.

[4] aus der durch das Tiefenbild gegebenen Bildraumtiefe von p_1 zurückrechenbar

Aufgrund der entstehenden $2 \cdot (m-1)(n-1)$ Dreiecke[5] wird dieses Vorgehen bei heute verwendeten Auflösungen wie FullHD (1920x1080 Pixel) zu aufwändig [Med14]. Da Quellbildbereiche mit geringen Varianzen in den Tiefenwerten nur grobe Triangulationen benötigen, wird eine adaptive Erzeugung des Dreiecksnetzes vorgenommen, unter Verwendung der Tessellation-Shader-Fähigkeiten heutiger Grafikkarten.

Dazu wird einmalig ein grobes Dreiecksnetz für alle Projektionen konstruiert. Infolge der hardwarebegrenzten maximal möglichen Unterteilung eines Primitivs[6] durch den Tessellation-Shader wird dieses Netz entsprechend der Quellbildauflösung so aufgebaut, dass bei maximaler Unterteilung aller Primitive wieder das Netz aus Bild 3a resultiert. Dadurch können Teilbereiche des Quellbilds immer noch zu stark trianguliert sein, jedoch kann, wie Bild 3b zeigt, eine große Ersparnis zum vorigen Ansatz entstehen.

Die dynamische Unterteilung eines Primitivs wird nur vorgenommen, sofern eine signifikante Tiefendifferenz durch den zugehörigen Quellbildbereich verläuft, was approximativ auf den achsenparallelen Kanten des Primitivs überprüft wird (siehe Bild 3b).

3 Lochreduktion

3.1 Detektion von Informationslücken

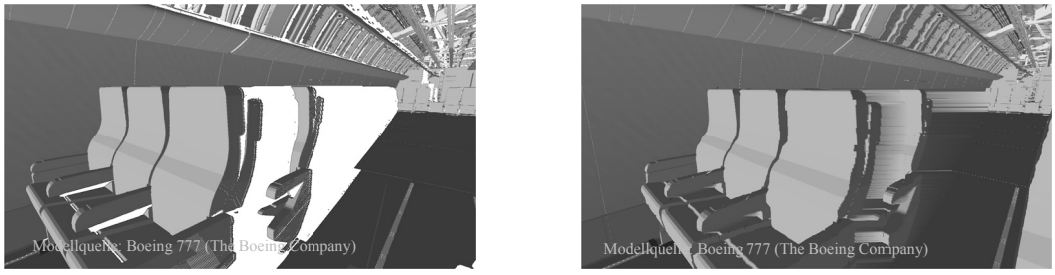


Bild 4: a) Im Quellbild fehlende Information führt zu Lücken (weiß) in Projektionen. b) Beim dreiecksbasierten Verfahren hingegen zeigen sich Artefakte.

Ein Hauptproblem des DIBR-Verfahrens ist in den verwendeten Einzelbildern fehlende Szeneninformation, was sich als sichtbare Lücken in synthetisierten Nutzeransichten äußert (siehe Bild 4a). Beim Verfahren dieser Arbeit zeigt sich das Problem jedoch in der Form wie in Bild 4b als falsche Flächen, was an der Interpretation des Bildraums als zusammenhängendes Dreiecksgitter liegt. Diese Artefakte werden unter anderem als „Rubber-Sheets“ [Mor09] oder als „Skins“ [GFB13] bezeichnet.

[5] m ist die Breite, n die Höhe des Quellbilds.

[6] Zum Zeitpunkt der Implementierung waren maximal 64^2 Unterteilungen möglich.

Solche Flächen verhindern die Bildung einer Lückenmaske[7] und müssen gesondert detektiert werden. Dies wird über eine Kantenerkennung im Quelltiefenbild, anhand dessen diskreter zweiter Ableitung (Laplace-Operator [Wan07]), erreicht. Während der DIBR-Projektion werden die erzeugten Vertices nun markiert, sofern ihre zugehörigen Quellbildpixel auf einer detektierten Kante liegen. Dreiecke, deren Vertices alle markiert sind, werden als falschen Flächen identifiziert und entfernt.

Ist der Tiefenabstand zwischen den Pixelbereichen nicht zu groß relativ zur Kamera, so bieten diese falschen Flächen jedoch eine brauchbare erste Füllung der korrespondierenden Informationslöcher. Die Kantendetektion zwischen zwei Quellpixeln wird daher vermieden, wenn ihr Tiefenabstand einen Schwellwert unterschreitet, welcher proportional zum Abstand der Pixel von der Kamera skaliert.

3.2 Multi-Kamera-Ansatz

Zum Füllen der detektierten Löcher wird hier ausschließlich auf vorhandene Szeneninformation zurückgegriffen. In anderen Arbeiten verwendete Inpaintingverfahren (bspw. in [NKD+11]) können zwar optisch plausible Rekonstruktionen liefern, weisen jedoch häufig merkliche Diskrepanzen zur korrekten Darstellung auf [Med14]. Multi-Layer Ansätze wie in [GFB13] hingegen erfordern eine Modifikation des Quellrenderers.

Daher wird für die Einzelbildsynthese ein Ansatz mit mehreren DIBR-Projektionen verschiedener zeitlich zurückliegender Einzelbilder verwendet. Löcher die so gefüllt werden können enthalten weitestgehend korrekte Szeneninformation. Neue Quellbildpaare werden vom klassischen Renderer asynchron erzeugt und in einen Puffer gelegt. Da nicht beliebig viele DIBR-Vorgänge ausgeführt werden können, werden die sinnvollsten Quellbilder in diesem Puffer nach folgendem Kriterium aufsteigend ausgewählt:

$$\frac{1 + \|c_i - c_u\|}{0.5\,(d_i \circ d_u + 1)} - 1 \tag{3}$$

$\|c_i - c_u\|$ ist der euklidische Abstand von Pufferkamera *i* zur aktuellen Nutzerkamera *u* und $d_i \circ d_u$ das Vektorskalarprodukt ihrer normierten Blickrichtungsvektoren. Es werden also Quellbilder bevorzugt, deren erzeugende Kamera eine geringe Distanz zur Nutzerkamera und ähnliche Blickrichtung wie diese hat. Hingegen werden Quellen mit den höchsten Werten bei Pufferüberlauf verworfen.

3.3 Vorausschauendes Rendern

Bereits gerenderte Quellbilder vergangener Kamerapositionen reichen in der Praxis kaum aus, da sich die Nutzerkamera stetig von diesen Positionen entfernen kann (bspw. durch eine simple Vorwärtsbewegung). Überwindet die Nutzerkamera dabei Hindernis-

[7] Stellen im Zielbild, die während der DIBR-Projektion nicht beschrieben wurden

se wie eine Wand, so können die entstehenden Lücken mit keinem der gepufferten Quellbilder gefüllt werden. Daher wird der von Mark et al. vorgeschlagene Ansatz [MM97] implementiert, bei dem durch den Quellrenderer zukünftige Kameraansichten zum Rendern verwendet werden.

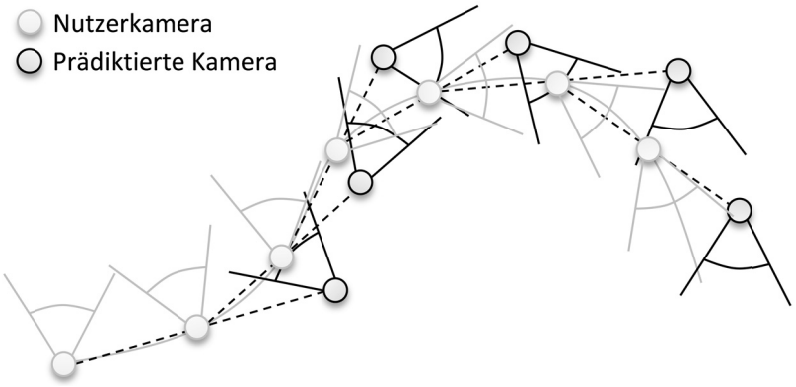


Bild 5: Lineare Kameraprädiktion (Originalbild: [Med14])

Diese Ansichten sind natürlicherweise nicht bekannt, sondern müssen geschätzt werden. Gemäß Bild 5werden Position und Rotation der angenommenen zukünftigen Nutzerkamera linear aus den zeitlich aktuellsten Kameras extrapoliert. Es fällt auf, dass diese Prädiktion Fehlern unterliegt, vor allem wenn sich der Nutzer erratisch verhält. Eine perfekte Prädiktion ist allerdings aufgrund der DIBR-Projektion nicht notwendig.

Weiterhin wird die Reaktionszeit der Prädiktion durch Berücksichtigung der vom Nutzer eingegebenen Steuerbefehle vermindert. Da sie dennoch hauptsächlich von der Renderzeit des Quellrenderers abhängt, kann es bei plötzlichen Änderungen der Kamerabewegung für kurze Zeit zu größeren Informationslücken kommen (vor allem am Bildrand bei Drehungen). Dies wird durch eine Vergrößerung des Quellkamerasichtfelds kompensiert, was allerdings auch eine Erhöhung der Quellbildauflösung erfordert [Med14].

4 Zusammenfassung des Systems

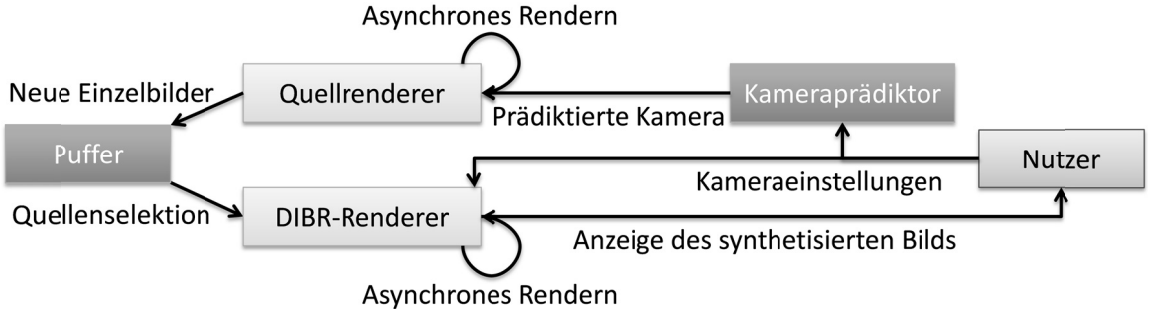


Bild 6: Implementiertes System

Eine Zusammensetzung der beschriebenen Abläufe ist auf Bild 6 abgebildet. Der Quellrenderer erzeugt fortwährend neue echte Einzelbilder, wobei die dafür verwendete Ka-

meraansicht aus dem Kameraprädiktor bezogen wird, und legt diese im Puffer ab. Asynchron dazu erzeugt der DIBR-Renderer aus den gepufferten Bildern die dem Nutzer final angezeigten Einzelbilder. Damit ist die sichtbare Kamerabewegung vom Quellrenderer entkoppelt. Der gesamte Vorgang findet auf derselben Hardware statt.

5 Experimentelle Ergebnisse

Die in Abschnitt 2 erläuterten Verfahren wurden im Rahmen dieser Arbeit auf der GPU implementiert, wobei punkt- und dreiecksbasierte vollständig in der Standardpipeline realisiert wurden, während das bildbasierte teilweise in CUDA umgesetzt wurde. Die Implementierung erfolgte in C++ und OpenGL auf einer Windows 7 Plattform, ausgestattet mit einer Intel Core i7-3820 CPU und einer NVIDIA GeForce GTX 650 GPU.

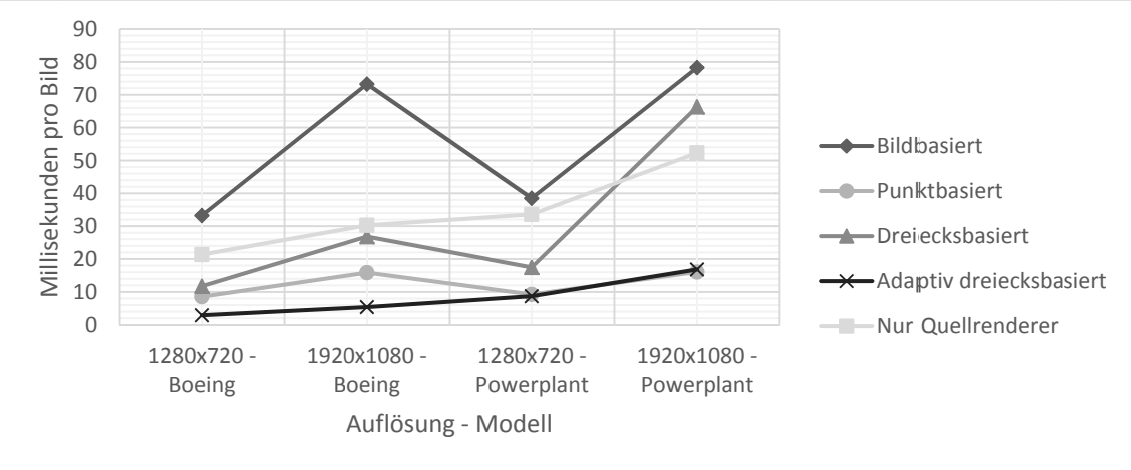


Bild 7: Renderzeiten der Verfahren im Vergleich (je niedriger desto besser)

Zum Testen wurden zwei Modelle verwendet: Das Boeing 777 Modell von The Boeing Company (ca. 300 Millionen Dreiecke) und das Powerplant Modell[8] der University of North Carolina (ca. 12,7 Millionen Dreiecke), letzteres allerdings zur Simulation hoher Tiefenvarianz in einem 8x8x8 Raster repliziert (ca. 6,5 Milliarden Dreiecke). Als Quellrenderer wurde der VGR-Renderer von 3DInteractive [BHP07] verwendet. Es wird nun die Renderzeit dieser Verfahren im Systembetrieb des vorigen Abschnitts verglichen. Für jede Bildsynthese wurden zwei Quellbilder projiziert. Das Quellkamerasichtfeld wurde auf 110° festgesetzt, im Gegensatz zu 90° bei der Zielkamera.

Wie Bild 7 zeigt, ist das rein bildbasierte Verfahren in der aktuellen Form für eine Beschleunigung des Quellrenderers zu langsam. Dies und beobachtete Unzulänglichkeiten des Verfahrens, wie störende Artefaktbildung [Med14], lassen von einer Verwendung abraten. Auch der in Abschnitt 2.3 angesprochene zu hohe Aufwand einer vollständigen Quellbildtriangulierung zeigt sich hier.

[8] http://gamma.cs.unc.edu/Powerplant/

Sowohl das punktbasierte als auch das adaptive dreiecksbasierte Verfahren erlauben auf der Testplatform eine Beschleunigung der angezeigten Bildrate auf über 60 Hz. Da das punktbasierte Verfahren durchschnittlich deutlich mehr Primitive benötigt, ist seine Renderzeit (Bild Boeing) im Vergleich zum adaptiven Verfahren oft höher. Ist die Grafikkarte allerdings bereits stark durch den Quellrenderer ausgelastet oder die Szene sehr komplex, so nähert sich das adaptive dem punktbasierten Verfahren zeitlich an oder kann langsamer sein (Bild Powerplant). Die Ursache dafür ist einerseits der höhere Aufwand pro Primitiv des adaptiven Verfahrens (bspw. Kantentests, Tessellierung, Flächenentfernung) und andererseits die mit der Varianz des Tiefenbilds steigende Dreieckszahl des adaptiv erzeugten Gitters.

Tabelle1: Mittlere Renderzeiten (μ) und Standartabweichung (σ) der Renderer im Parallelbetrieb auf derselben Hardware (Millisekunden)

	1280x720				1920x1080			
	Boeing		Powerplant		Boeing		Powerplant	
	μ	σ	μ	σ	μ	σ	μ	σ
Adaptiv Dreiecksbasiert DIBR	2,89	2,40	8,69	5,05	5,39	4,15	16,85	10,80
Quellrenderer (1.43x Auflösung)	68,45	42,28	167,58	71,25	119,13	75,78	479,04	326,24

6 Resümee und Ausblick

Im Verlauf dieser Arbeit wurde eine effiziente, vollständig in der Standard-GPU-Pipeline umgesetzte DIBR-Implementation gezeigt. Außerhalb konstruierter Härtefälle ist die Renderzeit deutlich niedriger als bei anderen Herangehensweisen. Die dadurch hohe Performanz der Anzeige und ihre systematische Trennung vom Quellrenderer ermöglichen das Verbergen hoher Quellbildlatenzen. Weiterhin werden beim DIBR-Verfahren einhergehende Informationslücken durch den Multi-Kamera-Puffer ohne großen Aufwand bereits stark reduziert.

Die Aufteilung des Systems in Abschnitt 4 ist geeignet für eine Client-Server-Implementierung wie in [GFB13]. Der Server führt dabei den Quellrenderer aus und liefert die Quellbilder an die Klienten, während diese den DIBR-Renderer benutzen. Dadurch kann auch auf hardwareschwachen Geräten mit Massive-Data-Rendering gearbeitet werden, das heißt auch auf mobilen Geräten.

Bei den meisten aktuellen Tablets empfiehlt sich allerdings eine Modifikation des Verfahrens, so dass der Server den Tessellierungsteil ausführt und lediglich das erzeugte, texturierte Dreiecksnetz sendet. Diese Modifikation ist darin begründet, dass sowohl die Hardwarebestückung mit Shadereinheiten auf diesen Geräten noch zu gering ist und auch generell Tessellation-Shader-Fähigkeiten dort wenig verbreitet sind. Dieses Konzept ist jenem in [GFB13] identisch, allerdings werden die Dreiecksnetzte hier schneller und mit höherer Auflösung erzeugt.

Die DIBR-Technik ist in ihrer Grundform nur für statische Szenen geeignet. Kameraabhängige Dynamik, wie spekulare Reflexionen oder Transparenz, und temporale Dynamik, wie Animationen, müssen gesondert behandelt werden [Med14]. Probleme mit kameraabhängiger Beleuchtung werden bereits durch Deferred Shading [HH04] vermieden. Hinsichtlich Transparenzen ist die zusätzliche Verwendung von Multi-Layered-Rendering interessant, während für Animationen eine Verbindung mit MCFI nahe liegt.

Zwar zeigte sich die Qualität der synthetisierten Bilder bei typischen Industriemodellen wie der Boeing 777 als sehr gut, jedoch konnten in Härtetests noch starke Lochbildungen beobachtet werden. Diese entstehen bei Quellbildraten von ca. 5 Hz und tiefer, da die Szene dann sehr spärlich geometrisch aktualisiert wird. Daher wird zur weiteren Lastreduktion eine bessere Strategie gesucht, bei der der Detailgrad der Primitivtessellierung nicht mehr binär gewählt wird und auch nicht immer vollständige Quellbilder projiziert werden, sondern lediglich noch zur Lochfüllung benötigte Teile.

Weiterhin erbt das Verfahren ein Problem des Originals in [MM97]: Ein Bereich von einem halben Quellbildpixel wird beim Verwerfen falscher Flächen mit entfernt (da die Vertices in den Pixelzentren liegen), was Strukturen mit nur einem Pixel Ausdehnung effektiv löschen kann. Hier ist zukünftig ein genaueres Entfernen von lediglich Primitivteilen oder ein detaillierteres Triangulieren von Rändern angedacht.

Die bisher lediglich lineare Kameraprädiktion ist durch eine nichtlineare zu ersetzen, da dies den Nutzerweg genauer modelliert [MM97]. Dahingehend bietet vor allem der Bereich des Machine Learnings interessante Ansätze robuster Prädiktoren.

Literatur

[BHP07] BRÜDERLIN, B.; HEYER, M.; PFÜTZNER, S.: Interviews3D: A Platform for Interactive Handling of Massive Data Sets. IEEE Computer Graphics and Applications, Band 27(6), S. 48-59, 2007

[CG02] CHANG, C.; GER, S.: Enhancing 3D Graphics on Mobile Devices by Image-Based Rendering. Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing, S. 1105-1111, London, 2002

[DCV97] DARSA, L.; COSTA SILVA, B.; VARSHNEY, A.: Navigating Static Environments Using Image-space Simplification and Morphing. Proceedings of the 1997 Symposium on Interactive 3D Graphics, S. 25-ff., New York, 1997

[Feh03] FEHN, C.: A 3D-TV Approach Using Depth-Image-Based Rendering (DIBR). Proceedings of Visualization, Imaging and Image Processing, S. 482-487 2003

[FWH98] FU, C. W.; WONG, T. T.; HENG, P. A.: Triangle-based View Interpolation Without Depth-Buffering. Journal of Graphics Tools 3, S. 13-31, 1998

[GFB13] GHILETIUC, J.; FÄRBER, M.; BRÜDERLIN, B.: Real-time Remote Rendering of Large 3D Models on Smartphones Using Multi-layered Impostors. Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications, S. 14:1-14:8, New York, 2013

[HH04] HARGREAVES, S.; HARRIS, M.: Deferred Shading. NVIDIA Präsentation: 6800 Leagues Under The Sea, URL

ftp://download.nvidia.com/developer/presentations/2004/6800_Leagues/6800_Leagues_Deferred_Shading.pdf , 2004

[LN10] LEE, Y.; NGUYEN, T.: High frame rate Motion Compensated Frame Interpolation in High-Definition video processing. Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference on, S. 858-861, 2010

[MB95] MCMILLAN, L.; BISHOP, G.: Head-Tracked Stereoscopic Display using Image Warping. Eurographics Rendering Workshop, S. 21-30, 1995

[MM97] MARK, W. R.; MCMILLAN, L.: Post-Rendering 3D Warping. Proceedings of the 1997 Symposium on Interactive 3D Graphics, S. 7-ff., New York, 1997

[MO95] MAX, N.; OHSAKI, K.: Rendering Trees from Precomputed Z-Buffer Views. Eurographics Rendering Workshop, S. 45-54, 1995

[Mor09] MORVAN, Y.: Acquisition, Compression and Rendering of Depth and Texture for Multi-View Video. Dissertation, Technische Universität Eindhoven, 2009

[Med14] MEDER, J.: Erzeugung extrapolierter Szenenansichten mittels Depth Image Based Rendering. Masterarbeit, Fakultät für Informatik und Automatisierung, Technische Universität Ilmenau, 2014

[NKD+11] NDJIKI-NYA, P.; KOPPEL, M.; DOSHIKOV, D.; LAKSHMAN, H.; MERKLE, P.; MULLER, K.; WIEGAND, T.: Depth Image-Based Rendering With Advanced Texture Synthesis for 3-D Video. Multimedia, IEEE Transactions on, Band 13(3), S. 453-465, 2011

[OYH09] OH, K.; YEA, S.; HO, Y.: Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video. Picture Coding Symposium, S. 1-4, 2009

[PLA+98] POPESCU, V.; LASTRA, A.; ALIAGA, D.; NETO, M.: Efficient Warping for Architectural Walkthroughs Using Layered Depth Images. IEEE Visualization, S. 211-215, 1998

[SGH+98] SHADE, J.; GORTLER, S.; HE, L.; SZELISKI, R.: Layered Depth Images. Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, S. 231-242, 1998

[Wan07] WANG, X.: Laplacian Operator-Based Edge Detectors. Pattern Analysis and Machine Intelligence, IEEE Transactions on, Band 29(5), S. 886-890, 2007

[Wes90] WESTHOVER, L.: Footprint Evaluation for Volume Rendering, Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, S. 367-376, New York, 1990

[XWY+13] XI, M.; WANG, L.; YANG, Q.; LI, D.; ZHANG, M.: Depth-image-based rendering with spatial and temporal texture synthesis for 3DTV. EURASIP Journal on Image and Video Processing 2013(1), S. 1-18, 2013

[ZT05] ZHANG, L.; TAM, W. J.: Stereoscopic image generation based on depth images for 3D TV. Broadcasting, IEEE Transactions on, Band 51(2), S. 191-199, 2005

Autoren

Julian Meder ist seit September 2013 Softwareentwickler bei 3DInteractive GmbH in Ilmenau. Er erhielt seinen Masterabschluss im Studiengang Informatik an der Technischen Universität Ilmenau im Oktober 2014. Zudem ist Herr Meder dort im Moment wissenschaftlicher Mitarbeiter im Fachgebiet Grafische Datenverarbeitung der Fakultät Automatik und Automatisierung und verfolgt im Rahmen dieser Beschäftigung eine Promotion im Bereich Visualisierung.

Automatische Ableitung geometrischer Eigenschaften von Bauteilen aus dem 3-D-Polygonmodell

Claudius Jähn, Matthias Fischer, Maria Gerges
Heinz Nixdorf Institut & Institut für Informatik
Universität Paderborn
Fürstenallee 11, 33102 Paderborn
Tel. +49 (0) 5251 / 60 64 66, Fax. +49 (0) 5251 / 60 64 82
E-Mail: {claudius|mafi|marmur}@upb.de

Jan Berssenbrügge
Heinz Nixdorf Institut
Universität Paderborn
Fürstenallee 11, 33102 Paderborn
Tel. +49 (0) 5251 / 60 62 32, Fax. +49 (0) 5251 / 60 62 68
E-Mail: Jan.Berssenbruegge@hni.upb.de

Zusammenfassung

In der CAD-unterstützten Entwicklung von technischen Systemen (Maschinen, Anlagen etc.) werden virtuelle Prototypen im Rahmen eines virtuellen Design-Reviews mit Hilfe eines VR-Systems gesamtheitlich betrachtet, um frühzeitig Fehler und Verbesserungsbedarf zu erkennen. Da die Analyse beweglicher Bauteile ein wichtiger Untersuchungsgegenstand ist, werden die beweglichen Bauteile einer Maschine (z. B. motorgetriebene Mechaniken) im VR-System animiert. Kinematikmodelle, die Rotationsachsen, Freiheitsgerade und Übersetzungsverhältnisse definieren, um die Bewegungsmöglichkeiten der Bauteile festzulegen, werden im zugrundeliegenden CAD-Modell in der Praxis oft nicht modelliert oder können bei der Konvertierung in polygonale Modelle für das VR-System verloren gehen. Unser Ziel ist die Reduzierung des notwendigen manuellen Nachbereitungsaufwand für das Design-Review durch eine automatische Berechnung der geometrischen Eigenschaften. Wir stellen Algorithmen vor, die geometrische Eigenschaften wie die Lage von im 3-D-Raum orientierten Objekten, Rotationsachsen von zylindrischen Objekten und Bohrungen, die Umkreise von Wellen, die Zahnanzahl von Zahnrädern, sowie die Zahnabstände von Zahnstangen aus polygonalen dreidimensionalen Objekten automatisch ermitteln. Die Algorithmen implementierten wir in unserem VR-System PADrend und erprobten sie anhand von Bauteilen aus virtuellen Prototypen realer Maschinen.

Schlüsselworte

Geometrische Eigenschaften, Animationen, CAD-Systeme, VR-Systeme

Automatic derivation of geometric properties of components out of the 3D polygon model

Abstract

In CAD-supported development of technical systems (machines, equipment, etc.) virtual prototypes are looked in its entirety in the context of virtual design reviews using a VR system, in order to early detect errors and need for improvements. Since the analysis of moving components is an important object of investigation, the moving parts of a machine (e.g., motor-driven mechanisms) are animated in the VR system. In practice kinematic models that define the rotation axis, degrees of freedom, and gear ratios to determine the possible movements of the components are often not modeled in the underlying CAD model or may be lost when converting to polygonal models for the VR system. Our goal is to reduce the necessary manual follow-up costs for the design review by an automatic calculation of geometrical properties. We present algorithms to automatically determine the geometric properties such as the location of in 3D space oriented objects, rotation axes of cylindrical objects and holes, the perimeters of waves, the number of tooth of gears, as well as the tooth spacing of toothed racks of polygonal three-dimensional objects. We implemented the algorithms in our VR system PADrend and tested it on basis of components of virtual prototypes of real machines.

Keywords

Geometric properties, animations, CAD systems, VR systems

1 Motivation

Ein wesentlicher Bestandteil in der CAD-unterstützten Entwicklung von technischen Systemen (Maschinen, Anlagen etc.) ist die gesamtheitliche Betrachtung des virtuellen Prototypen im Rahmen eines virtuellen Design-Reviews. Dabei wird der Entwurf von Baugruppen oder einzelner Bauteile des Produkts systematisch, ganzheitlich in einer virtuellen Szene betrachtet und beurteilt. So können Fehler und Verbesserungsbedarf frühzeitig erkannt werden. Neben der Untersuchung der Bauteilgeometrie ist die Analyse von beweglichen Bauteilen ein wichtiger Untersuchungsgegenstand am virtuellen Prototypen. Dazu müssen die beweglichen Bauteile einer Maschine (beispielsweise motorgetriebene Mechaniken) am virtuellen Prototypen animiert werden. Dies kann ein sehr aufwändiger Prozess sein, falls die erforderlichen Daten (wie z.B. Rotationsachsen, Übersetzungsverhältnisse, Bewegungsfreiheitsgrade, kinematische Ketten etc.) nicht zur Verfügung stehen.

Im parametrischen Modell des CAD-Systems sind Grundkörper mit ihren *geometrischen Eigenschaften* (Position, Länge, Radius usw.) definiert. Ebenso kann ein Kinematikmodell anhand von Rotationsachsen, Gelenken und Freiheitsgeraden definiert werden, das die Bewegungsmöglichkeiten der Bauteile festlegt. In der Praxis werden jedoch oft viele dieser Eigenschaften nicht im CAD-System definiert, da dies, im Verhältnis zum erwarteten Nutzen, einen zu großen Aufwand darstellt. Des Weiteren können diese Informationen verloren gehen, wenn das CAD-Modell in ein Austauschformat für VR-Systeme umgewandelt wird. Die parametrischen CAD-Modelle werden dabei in *Polyonmodelle* umgewandelt, welche die exakte Geometrie der CAD-Bauteile durch polygonbasierte Oberflächenmodelle approximieren. Das Kinematikmodell wird gar nicht oder nur unvollständig konvertiert. Im VR-System müssen die Bewegungsabläufe dann durch Animationspfade manuell nachgebildet werden. Schwierig und zeitaufwändig ist dabei insbesondere die manuelle Festlegung der Rotationsachsen einzelner Bauteile, z. B. von Zahnrädern oder Antrieben und die Bestimmung von exakten Übersetzungsverhältnissen.

Ziel dieser Arbeit ist, den notwendigen manuellen Nachbereitungsaufwand für das Design Review zu reduzieren, indem die Berechnung der geometrischen Eigenschaften automatisiert wird. In dieser Arbeit entwickeln wir Algorithmen, die solche *geometrische Informationen* und *Eigenschaften* aus polygonalen dreidimensionalen Objekten *automatisch* ermitteln. Wir haben verschiedene Algorithmen entwickelt, die unterschiedliche Eigenschaften allein auf Basis der Polygondaten berechnen. Dazu gehört die Erkennung der Lage von beliebig im 3-D-Raum orientierten Objekten, Rotationsachsen von zylindrischen Objekten (und zylindrischen Bohrungen), der Umkreise von Wellen, der Zahnanzahl von Zahnrädern und Zahnabstände von Zahnstangen. Die Algorithmen haben wir in unserem VR-System PADrend [EJP11] implementiert und anhand von Bauteilen aus virtuellen Prototypen realer Maschinen erprobt.

2 Stand der Technik

Erste Betrachtungen zur Erkennung geometrischer Eigenschaften wurden im der algorithmischen Geometrie in der 2-D-Ebene über einer Menge fest definierter Objekttypen (Punkte, Liniensegmente, Ellipsen, Kreise) durchgeführt. Algorithmen bestimmen in optimaler Laufzeit $\Theta(n \log n)$ alle exakten Symmetrieachsen [Ata85]. Die Symmetrie eines einzelnen Polygons im 2-D kann in Zeit $\Theta(n)$ und die Symmetrie einer Punktemenge und eines einzelnen Polyeders im 3-D-Raum kann in optimaler Zeit $\Theta(n \log n)$ berechnet werden [WWV85]. Diese Methoden berechnen exakte Symmetrieachsen bzgl. der Punktmengen der Eingabe. Sie lassen sich schwer auf unser Problem anwenden, da durch den Export der CAD-Modelle, die Körper zwar symmetrisch erscheinen, die Punkte des exportierten Modells jedoch nicht notwendigerweise symmetrisch angeordnet sind.

Die Berechnung geometrischer Eigenschaften ist ein zentrales Thema in der Computervision. Die Bestimmung von Symmetrieeigenschaften wie Achsen oder Ebenen von Objekten (Flugzeuge, Gebäude, Personen) in Fotos natürlicher Umgebungen hat eine so hohe Bedeutung, dass die entwickelten Algorithmen zu Wettbewerben antreten [RBK+11]. Herausfordernd ist die durch die perspektivische Darstellung verzerrte Darstellung der Objekte im 2-D-Bildraum. Man versucht Merkmale aus dem Foto zu extrahieren und für Gruppen solcher Merkmale Symmetrieeigenschaften zu erkennen [CPM+07]. Erschwerend ist, dass die zu entdeckenden Rotationssymmetrien (Autorad) oder Disymmetrien (Schmetterling) nur lokale Eigenschaften des Fotos vor einem ggfs. komplexen Hintergrund sind [LE06]. Daher werden auch Methoden aus dem Bereich des künstlichen Lernens zur Bestimmung von Symmetrien eingesetzt [TK12].

Einen engeren Bezug zu unseren Arbeiten ist die automatische Erkennung von Symmetrien in 3-D-Modellen: Typischerweise sind es Einzelobjekte, deren, durch das Modell gegebene Symmetrieeigenschaft (z.B. Achse oder Ebene), man bestmöglich erkennen möchte [KLA+14]. Die Symmetrie von Einzelobjekten kann auch dann erkannt werden, wenn symmetrisch aufgebaute Modelle (z.B. Menschen), in einer nichtsymmetrischen Pose verharren (verformte Modelle). Die Symmetrie wird hergestellt als Prozess von zulässigen Transformationen, z.B.: Skalierung oder starrer Verschiebung [OSG08]. Bei partieller Symmetrie fehlen zudem noch Teile des an sich symmetrischen Modells [RBB+10]. Für einzelne 3-D-Modelle kann Symmetrie in Form von Kugelflächenfunktionen so beschrieben werden, dass für ein gegebenes symmetrisches Objekt (z.B. Rundtisch) aus einer Datenbank von beliebigen 3-D-Objekten weitere Rundtische gefunden werden, die ähnlich aussehen, aber durchaus verschieden sind [KFR04]. In komplexen, aus vielen Teilobjekten zusammengesetzten Modellen, werden Teilobjekte über die Bestimmung der Symmetrieeigenschaften als vergleichbar identifiziert, um Modelle zu komprimieren und instanziieren [MSH+06]. Ähnlichkeiten zu der Vorgehensweise in unserem Ansatz sind in der Arbeit von Mitra et al. zu finden [MGP06]. Dort werden paarweise Samples gezogen, um Kandidaten für Symmetrien zu finden. Es findet jedoch eine Transformation der Paare statt. In dem transformierten Raum werden

über Clustering-Algorithmen die Teilmengen der Punkte gesucht, die Symmetrieeigenschaften besitzen. Unser Ansatz zur Bestimmung von Rotationsachsen unterscheidet sich in seiner Einfachheit und in den verwendeten Kriterien, um eine Symmetrieachse zu finden.

3 Algorithmen zur Bestimmung geometrischer Eigenschaften

Im Folgenden beschreiben wir die unterschiedlichen Algorithmen zur Bestimmung der geometrischen Eigenschaften:

3.1 Beschreibung der Eingabe

Beim Design von Maschinen durch CAD werden häufig parametrisch beschriebene Modelle verwendet. Dazu gehören einfache geometrische Figuren, wie Quader, Kugeln oder Zylinder; aber auch komplexere Oberflächen, wie NURBS-Flächen oder logische Verbindungen zwischen mehreren solcher Figuren. Um die dadurch beschriebenen Objekte in einem VR-System darstellen zu können, müssen die Oberflächen der parametrisch beschriebenen Objekte durch Polygone angenähert werden. Praktisch werden hier meistens Dreiecke verwendet, die auf der Oberfläche der Objekte mit einstellbarem Detailgrad erzeugt werden. Die Daten zu den Eckpunkten der Dreiecke (die *Vertices*) beinhalten die 3-D-Position des Punktes im Raum; können jedoch weitere Daten wie Normalenvektoren (zur Bestimmung der räumlichen Ausrichtung der Oberfläche), Farbwerte oder auch Texturkoordinaten enthalten.

Als Eingabe für die weitere Verarbeitung gehen wir von einer Menge von triangulierten Oberflächenmodellen (*Gittermodelle*) aus. (Bild 1 und Bild 2 zeigen in der unteren Reihe die Gittermodelle für verschiedene Objekte.) Die vorgestellten Verfahren benötigen nur die Information, welche Vertices zu einem Dreieck gehören und die 3-D-Positionen der Vertices. Andere Eigenschaften, wie beispielsweise die Normalenvektoren, werden nicht benötigt. Um diese Eingabe aus den zugrundeliegenden CAD-Daten zu generieren, können entsprechende Exportfunktionen gängiger CAD-Systeme verwendet werden, oder es können externe Konverter verwendet werden. Die zu animierenden Bauteile sollten dabei als einzelne Objekte repräsentiert werden. Werden ganze Baugruppen (beispielsweise mit mehreren Zahnrädern) als ein gemeinsames Objekt dargestellt, können die Einzelteile jedoch in den meisten Fällen auch im VR-System noch getrennt werden (durch eine einfach umzusetzende Aufteilung des Gittermodells in Zusammenhangskomponenten).

3.2 Bestimmung der Ausrichtung eines Objektes

Das Ziel dieses Schrittes ist es, Kandidaten für die Ausrichtung eines Gittermodells im Raum zu ermitteln, da die Objekte beliebig im orientiert sein können. Das Ergebnis sind ein oder mehrere 3-D-Richtungsvektoren, die die Ausrichtung beschreiben. Für ein

Zahnrad oder einen zylindrische Achse ist dies z. B. die Richtung der Rotationsachse, oder für ein quaderförmiges Objekt die drei Richtungen der Seiten. Zur Bestimmung der Ausrichtung nutzt unser Verfahren die Beobachtung, dass bei vielen geometrischen Objekten, eine große Anzahl von Dreieckskanten, auf den gleichen Ebenen im Raum liegen. Die Normalenvektoren der Ebenen mit einer Häufung der darauf liegenden Kanten, entsprechen dann den gesuchten Ausrichtungen der Objekte. Die naheliegendere Betrachtung einer Häufung der Normalenvektoren der Dreiecke des Objektes kann bei einigen Objekten ebenfalls zum gewünschten Ergebnis führen, jedoch nicht für Objekte, bei denen keine (oder nur kleine) gerade Flächen orthogonal zur Ausrichtung existieren.

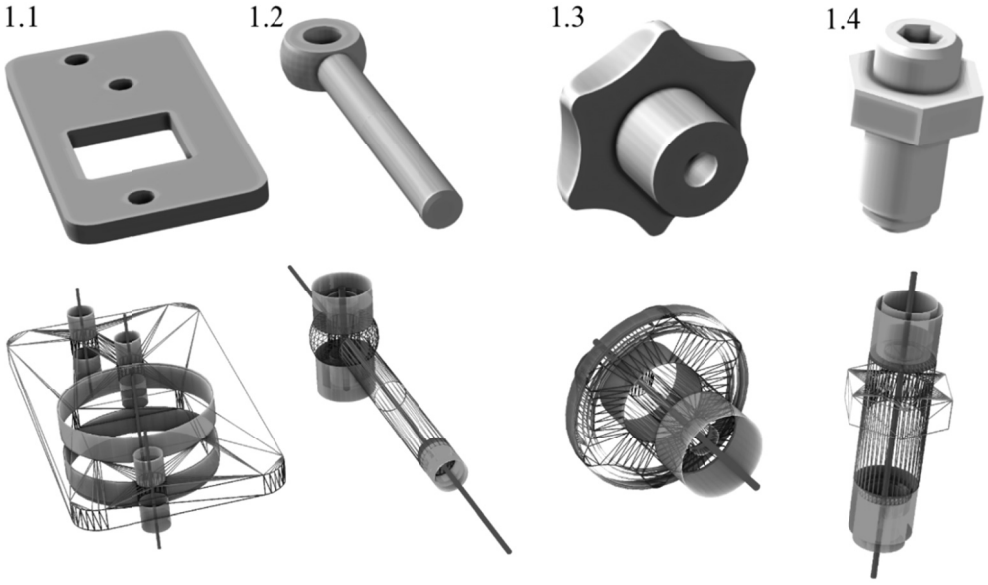


Bild 1: Verschiedene Objekte mit zylindrischen Bereichen; (unten) Gittermodell der Objekte mit Visualisierung der automatisch erkannten Rotationsachsen und der vom Algorithmus dafür verwendeten Kreise.

Der von uns entwickelte randomisierte Algorithmus arbeitet folgendermaßen: Der Algorithmus wählt wiederholt eine zufällige Kante eines zufällig gewählten Dreiecks. Aus den beiden Eckpunkten der Kante und einem weiteren, zufällig gewählten Eckpunkt eines anderen Dreiecks, wird die Ebene im Raum bestimmt, auf denen die drei Punkte liegen. Der Normalenvektor der Ebene wird als Kandidat für den Richtungsvektor registriert. Liegen die drei Punkte auf einer Geraden, wird auch die Richtung der Geraden als Kandidat für den Richtungsvektor registriert. Die Richtungsvektoren, die nach einer mehrfachen Wiederholung besonders oft registriert werden, sind die Kandidaten für die Ausrichtung des Objektes. Sobald ein Kandidat eine einstellbare Trefferquote erreicht hat, werden die besten Kandidaten zurückgegeben. Ein Richtwert für die Abbruchbedingung sind 500 Treffer, wobei dann alle Richtungen mit mehr als 250 Treffern weiter betrachtet werden.

Da die Positionsdaten technisch bedingt als Fließkommazahlen vorliegen, kommt es bei den Berechnungen zu Ungenauigkeiten. Um mehrere 3-D-Vektoren sinnvoll verglei-

chen und zusammenfassen zu können, werden die Koordinaten mit einer einstellbaren Schrittweite diskretisiert (Standardwert 0.01 mm).

3.3 Bestimmung von Rotationsachsen

Nachdem im vorherigen Schritt Kandidaten für Richtungen der Rotationsachsen identifiziert wurden, werden damit im Folgenden Kandidaten für Rotationsachsen identifiziert. Dazu werden für Achsen jeweils Kandidaten für 3-D-Punkte gesucht, die als Ursprung der Achse dienen können. Für unseren Algorithmus versuchen wir die folgende Beobachtung auszunutzen: Bei geometrischen Objekten mit erkennbaren Rotationsachsen, liegen oft viele Eckpunkte von Dreiecken im gleichen Abstand, kreisförmig um die Rotationsachsen verteilt – findet man diese Kreise, liefert ihr Zentrum Kandidaten für den gesuchten 3-D-Punkt.

Unser randomisierter Algorithmus arbeitet wie folgt: Für jeden Kandidaten der Richtungsvektoren aus dem vorherigen Schritt wird die Gesamtmenge der Vertices des Objektes zunächst in mehrere Teilmengen aufgeteilt. Jede Teilmenge enthält die Vertices, die mit dem Richtungsvektor jeweils die gleiche Ebene aufspannen (mit entsprechender Diskretisierung). Alle Vertices einer Teilmenge liegen also auf einer Ebene. Für jede Teilmenge, die genügend Vertices enthält, werden mögliche Kreise bestimmt (Richtwert für die Mindestgröße einer Teilmenge ist 30% der Größe der größten Teilmenge). Dazu wird wiederholt eine zufällige Teilmenge von drei Vertices aus der Menge gewählt und der durch die Vertices aufgespannte Kreis registriert. Sobald ein Kreiskandidat genügend Treffer aufweist (Richtwert 200 Treffer), oder nach einigen Iterationen noch kein Kandidat gefunden wurde (Richtwert 10000 Iterationen), wird mit der nächsten Teilmenge fortgeführt. Abschließend werden alle Kreiskandidaten gefiltert, so dass nur Kandidaten weiter betrachtet werden, die relativ zum besten Kandidaten ausreichend Treffer aufweisen (Richtwert: 30%).

Da Rotationsachsen meistens durch das Zentrum mehrerer Kreise verlaufen, werden die Kreise anhand ihrer Normalen und ihrer Zentren zusammengefasst. Die Zentren und Richtungen mit insgesamt den meisten Treffern ergeben die gesuchten, möglichen Rotationsachsen.

Der Algorithmus funktioniert sowohl bei zylindrischen Objekten, wie auch bei kreisförmigen Löchern (siehe Bild 1). Der Kreis kann durch Aussparungen unterbrochen sein, bis hin zu regelmäßigen Unterbrechungen wie bei Zahnrädern. Grenzen des Algorithmus zeigen sich dort, wo die Anzahl der Vertices in den gesuchten Kreisen im Verhältnis zur Gesamtanzahl der Vertices im Objekt sehr gering ist. Beispiele hierfür sind komplexere Werkstücke mit vielen Bohrlöchern, die nur sehr schwer als Achsen erkannt werden.

3.4 Bestimmung von Übersetzungsverhältnissen

Neben den Rotationsachsen benötigt man für das Modellieren einer komplexeren kinematischen Kette, wie beispielsweise in Getrieben, auch Informationen zu Übersetzungsverhältnissen. Der Umfang von Rädern lässt sich direkt aus den im letzten Schritt bestimmten Kreisen bestimmen. Für Zahnräder erlaubt die Kenntnis über die Anzahl der Zähne eine einfache und exakte Bestimmung von Übersetzungsverhältnissen. Während das manuelle Zählen von einer Handvoll Zähnen problemlos funktioniert, ist dies bei größeren Zahnrädern mit Hundert oder mehr Zähnen sehr zeitaufwändig. Die grundlegende Idee für das automatisierte Zählen von Zahnrädern ist, entlang eines erkannten Umkreises, möglichst viele, große Kugeln zu verteilen, so dass in jedem Zahn eine Kugel liegt, die keinen Vertex des Objektes enthält. Die Anzahl der Kugeln entspricht dann der Anzahl der Zähne des Zahnrades. Sind mehrere Zahnräder in einem Objekt auf einer Welle verbunden, können so auch mehre Anzahlen bestimmt werden, wenn die entsprechenden Umkreise als Ausgangsbasis verwendet werden.

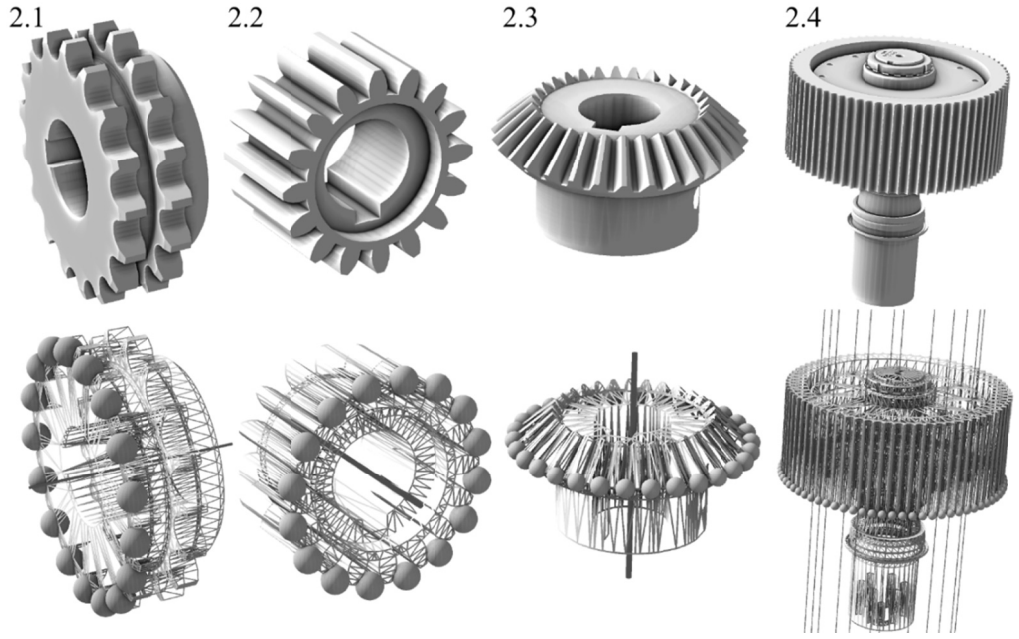


Bild 2: Zahnräder mit unterschiedler Zahnanzahlen und Formen; (unten) Gittermodell der Modelle mit Visualisierung der automatisch erkannten Rotationsachsen und Kugeln zur Bestimmung der Zahnanzahl.

Unser Algorithmus verwendet als Eingabe einen im 3-D-Raum platzierten Kreis (aus dem vorherigen Schritt) und die Menge der Vertices des Objektes. Entlang der Kreislinie werden in kleinen Schritten (Richtwert: 0.0036 Grad) Positionen gewählt. Für jede Position wird der nächstliegende Vertex ermittelt (mit Hilfe eines Octrees). Mit der Position als Zentrum und der Distanz zum nächsten Vertex als Radius, wird eine Kugel im 3-D-Raum erzeugt. Die Kugel wird gespeichert, wenn sie von keiner anderen Kugel in der Nachbarschaft mit kleinerem Radius geschnitten wird. Nachdem alle Kugeln erzeugt wurden, werden noch alle kleinen Kugeln (mit einem Radius von kleiner als 90%

des maximalen Radius) entfernt. Die Anzahl der verbleibenden Kugeln entspricht im Erfolgsfall der Anzahl der Zähne des Zahnrades.

Der Algorithmus erkennt robust Zahnräder mit unterschiedlichen Zahnanzahlen und Zahnformen. Auch Kegelräder, Zahnräder mit Schrägverzahnung und innenliegende Verzahnungen werden erkannt (siehe Bild 2). Die Erkennung beschränkt sich jedoch auf rund aufgebaute Zahnräder. Auch müssen die Außenkanten der Zähne an einer Seite möglichst frei liegen. Mit einem ähnlich gestalteten Algorithmus lassen sich auch die Zahnabstände von Zahnstangen und der Vorschub von Außengewinden bestimmen (siehe Bild 3). Entlang einer eng umschließenden Boundingbox (mit der berechneten Ausrichtung) werden an den Kanten und mittig entlang der Seiten Kugeln verteilt; äquivalent zu den Kugeln auf den Umkreisen der Zahnräder. Die Kugeln werden gefiltert, nach Größe klassifiziert und der Abstand zwischen Kugeln gleicher Größe ausgegeben. Der Benutzer wählt schließlich aus den erkannten, möglichen Abständen, den gewünschten aus.

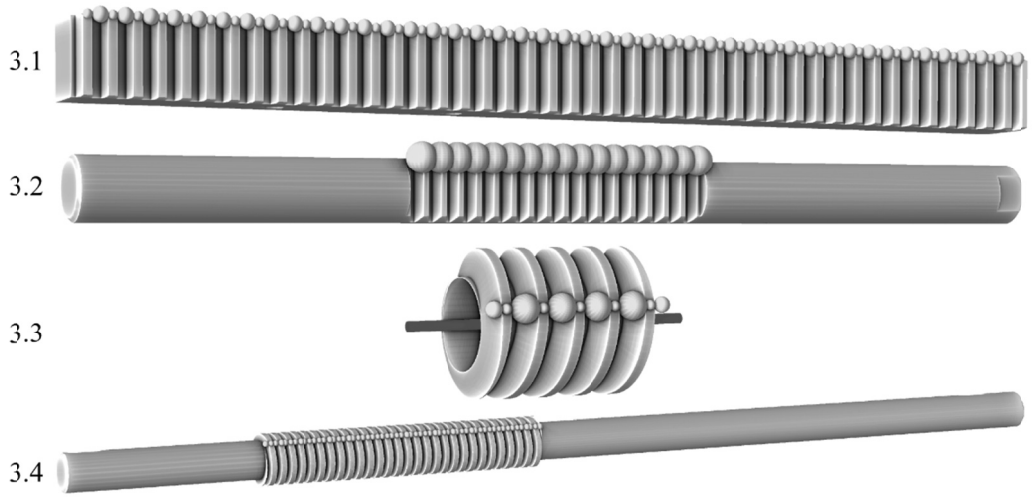


Bild 3: Verschiedene Zahnstangen und Außengewinde mit Visualisierung der Kugeln zur automatischen Bestimmung des Zahnabstandes bzw. der Steigung.

4 Gestaltung der Benutzerschnittstelle

Die vorgestellten Algorithmen wurden im PADrend-System (*Platform for Algorithm Development and rendering)* umgesetzt. Während ein Benutzer einen virtuellen Prototyp durch Animationen und Interaktionsmethoden anreichert, stehen die einzelnen Funktionen als interaktive Werkzeuge zur Verfügung. Um ein Zahnrad für eine Animation vorzubereiten, selektiert der Benutzer zunächst das Zahnradobjekt. Nun ruft der Benutzer die Funktion zur Bestimmung der Rotationsachse auf (siehe Abschnitt 3.3). Als Ergebnis werden ein oder mehrere Kandidaten für Rotationsachsen im 3-D-Raum durch Zylinder dargestellt. Der Benutzer wählt die gewünschte Achse aus und bestätigt sie damit als Rotationsachse. Durch Drehen des Objektes, lässt sich die Wahl sofort überprüfen. Wurde die gewünschte Achse nicht gefunden, können die Parameter des

Algorithmus angepasst werden. Um den Benutzer nicht zwingend mit den Details des Algorithmus zu konfrontieren, werden Werte für alle Parameter in mehreren Voreinstellungen (Schnell, Standard, Gründlich) zusammengefasst.

Um für die Bestimmung eines Übersetzungsverhältnisses, die Anzahl der Zähne des Zahnrads zu bestimmen, wählt der Benutzer zunächst die Funktion, die die in Abschnitt 3.3 bestimmten Kreise im 3-D-Raum als offene Zylinder darstellt. Der Benutzer kann nun den Zylinder auswählen, der die Zähne des Zahnrades umschließt, um hierfür die Berechnung zu starten (siehe Abschnitt 3.4) Die Anzahl der gefundenen Zähne wird ausgegeben und zur Überprüfung der Ergebnisse werden die, zwischen den Zähnen platzierten, Kugeln visualisiert.

5 Laufzeit

Die Zeit, die zum Berechnen der Eigenschaften benötigt wird, hängt neben der Anzahl der Vertices eines Objektes, auch von der geometrischen Struktur des Gittermodells ab. Die folgende Tabelle zeigt die benötigten Zeiten für die Berechnung für die in den Abbildungen gezeigten Objekte mit unserer prototypischen Implementierung[1] (dominante Schritte):

Tabelle 1: Laufzeiten für die Berechnung geometrischer Eigenschaften.

Objekt	(Abbildung)	Anzahl Dreiecke	Ausrichtung erkennen	Kreise erkennen	Zähne erkennen	Gesamtzeit
Blech mit Bohrungen	(1.1)	420	0.1 s	**9.3 s**		9.5 s
Welle	(1.2)	958	1.2 s	0.5 s		1.8 s
Handgriff	(1.3)	3566	3.5 s	1.6 s		5.1 s
Schraube	(1.4)	768	0.9 s	0.3 s		1.3 s
Zahnräder	(2.1)	4088	0.8 s	2.1 s	0.3 s	3.3 s
	(2.2)	1360	0.2 s	2.9 s	0.3 s	3.5 s
	(2.3)	1457	2.2 s	4.1 s	0.3 s	6.7 s
	(2.4)	23753	1.6 s	**14.6 s**	0.3 s	16.8 s
Zahnstange	(3.1)	1172	0.4 s		1.5 s	1.9 s
Zahnstange (rund)	(3.2)	874	4.9 s		1.7 s	6.6 s
Gewinde	(3.3)	2696	**8.9 s**		1.3 s	10.2 s
Gewindestange	(3.4)	7912	**9.0 s**		3.4 s	12.3 s

An den Messungen zu erkennen ist, dass die Zeiten für die einzelnen Schritte der Algorithmen stark variieren können. Können mehrere Kreise auf einer Ebene erkannt werden, wie bei dem Blech mit mehreren Bohrungen (Bild 1.1) oder bei dem Zahnrad mit mehreren kleineren Gewindebohrungen (Bild 2.4), wird die Laufzeit im Wesentlichen durch das Erkennen dieser Kreise bestimmt (mit mehreren Hunderttausend Iterationen).

[1] Testsystem: Intel Core i7-3770 CPU 3.4GHz; Windows 8.1 64Bit; PADrend 1.1beta

Eine andere Herausforderung für die Effizienz ist die Erkennung der Ausrichtung. Liegen nur wenige Vertices auf den gesuchten Ebenen (orthogonal zur Ausrichtung), müssen eine große Anzahl von Stichproben geprüft werden. Dies ist vor allem bei Außengewinden der Fall (Bild 3.3 und Bild 3.4).

Die prototypische Implementierung der Algorithmen wurde in einer Skriptsprache (EScript) geschrieben. Durch eine Umsetzung z. B. in C++ lässt sich die Laufzeit schätzungsweise halbieren.

6 Resümee und Ausblick

Die von uns vorgestellten Algorithmen bieten einen unkomplizierten Weg Rotationsachen und die Anzahl von Zahnradzähnen aus polygonalen Modellen zu bestimmen. Die Verfahren stellen nur wenige Anforderungen an die Eingabe und erlauben so die robuste Behandlung unterschiedlicher Modelle, ohne dass die Algorithmen gesondert angepasst werden müssen. Wenn die Möglichkeit besteht, die erforderlichen Daten direkt aus den originalen CAD-Daten zu extrahieren, kann dies genauere Werte liefern. Wenn diese Möglichkeit jedoch nicht besteht, dann bietet unsere Lösung eine pragmatische und effiziente Alternative und kann so den Aufwand zum Erstellen von komplexen Animationen für ein virtuelles Design-Review deutlich reduzieren.

Als nächste Schritte planen wir die Erweiterung der Algorithmen, um weitere Eigenschaften zu erkennen; beispielsweise um den genauen Verlauf von schienenbasierten Transportsystemen zu bestimmen. Um die Effizienz zu steigern, lassen sich die meisten Schritte der vorgestellten Algorithmen gut parallelisieren, da sie auf unabhängig durchgeführten Zufallsexperimenten beruhen. Weiterführende Anknüpfungspunkte sind die Unterstützung von Modellen, die nicht aus Polygonen, sondern aus Punktwolken aus 3-D-Laserscans bestehen oder die Kombination mit Verfahren zur Objekterkennung, so dass komplexere Strukturen, wie gesamte Getriebe, in einem stärkeren Maße automatisch erkannt und animiert werden können.

Danksagung

Dieses Forschungs- und Entwicklungsprojekt wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) im Rahmen des Spitzenclusters „Intelligente Technische Systeme OstWestfalenLippe (it´s OWL)“ gefördert und vom Projektträger Karlsruhe (PTKA) betreut. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Literatur

[Ata85] ATALLAH, M. J.: On Symmetry Detection. IEEE Transactions on Computers, Vol. 34, Nr. 7, S. 663-666, 1985.

[CPM+07] CORNELIUS, H.; PERĎOCH, M.; MATAS, J.; LOY, G.: Efficient Symmetry Detection Using Local Affine Frames. Bd. 4522, in Image Analysis, Herausgeber: BjarneKjær Ersbøll und KimSteenstrup Pedersen, Springer Berlin Heidelberg, S. 152-161, 2007.

[EJP11] EIKEL, B.; JÄHN, C.; PETRING, R.: PADrend: Platform for Algorithm Development and Rendering. Herausgeber: Jürgen Gausemeier, Michael Grafe und Friedhelm Meyer auf der Heide. Augmented & Virtual Reality in der Produktentstehung. Paderborn: Heinz Nixdorf Institut, Universität Paderborn, S. 159-170, 2011.

[KFR04] KAZHDAN, M.; FUNKHOUSER, T.; RUSINKIEWICZ, S.: Symmetry Descriptors and 3D Shape Matching. Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. New York, NY, USA: ACM, S. 115-123, 2004.

[KLA+14] KORMAN, S.; LITMAN, R.; AVIDAN, S.; BRONSTEIN, A.: Probably Approximately Symmetric: Fast Rigid Symmetry Detection With Global Guarantees. Computer Graphics Forum, 2014.

[LE06] LOY, G.; EKLUNDH, J. O.: Detecting Symmetry and Symmetric Constellations of Features. Proceedings of the 9th European Conference on Computer Vision – Volume Part II. Berlin, Heidelberg: Springer-Verlag, S. 508-521, 2006.

[MSH+06] MARTINET, A.; SOLER, C.; HOLZSCHUCH, N.; SILLION, F. X.: Accurate Detection of Symmetries in 3D Shapes. ACM Transactions on Graphics, Vol. 25, Nr. 2, S. 439-464, 2006.

[MGP06] MITRA, N. J.; GUIBAS, L. J.; PAULY, M.: Partial and Approximate Symmetry Detection for 3D Geometry. ACM Transaction on Graphics, ACM SIGGRAPH 2006, Vol. 25, Nr. 3, S. 560-568, 2006.

[OSG08] OVSJANIKOV, M.; SUN, J.; GUIBAS, L.: Global Intrinsic Symmetries of Shapes. Proceedings of the Symposium on Geometry Processing. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, S. 1341-1348, 2008.

[RBK+11] RAUSCHERT, I.; BROCKLEHURST, K.; KASHYAP, S.; LIU, J.; LIU, Y.: First Symmetry Detection Competition: Summary and Results. CSE Dept. Technical Report, Department of Computer Science and Engineering, Pennsylvania State University, 2011.

[RBB+10] RAVIV, D.; BRONSTEIN, A. M.; BRONSTEIN, M. M.; KIMMEL, R.: Full and Partial Symmetries of Non-rigid Shapes. International Journal of Computer Vision (Springer US), Vol. 89, Nr. 1, S. 18-39, 2010.

[TK12] TSOGKAS, S.; KOKKINOS, I.: Learning-Based Symmetry Detection in Natural Images. Bd. 7578, in Computer Vision – ECCV 2012, Herausgeber: Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato und Cordelia Schmid, Springer Berlin Heidelberg, S. 41-54, 2012.

[WWV85] WOLTER, J. D.; WOO, T. C.; VOLZ, R. A.: Optimal algorithms for symmetry detection in two and three dimensions. The Visual Computer, Springer-Verlag, Vol. 1, Nr. 1, S. 37-48, 1985.

Autoren

Dipl.-Inform. Claudius Jähn studierte Informatik an der Universität Paderborn und ist wissenschaftlicher Mitarbeiter in der Fachgruppe Algorithmen und Komplexität am Heinz Nixdorf Institut. Seine Forschungsinteressen liegen im Bereich Computergrafik bei der Darstellung hochkomplexer Szenen und der Analyse von Szeneneigenschaften.

Dr. rer. nat. Matthias Fischer studierte Informatik an der Universität Paderborn und ist wissenschaftlicher Mitarbeiter in der Fachgruppe Algorithmen und Komplexität am Heinz Nixdorf Institut. Er promovierte dort über verteilte virtuelle Szenen. Seine Forschungsinteressen liegen im Bereich Computergrafik, Echtzeit-Renderingalgorithmen und verteiltes Rechnen.

B. Sc. Maria Gerges studiert Informatik an der Universität Paderborn, ihre Interessen liegen im Bereich der Computergrafik.

Dr.-Ing. Jan Berssenbrügge ist wissenschaftlicher Mitarbeiter in der Fachgruppe Produktentstehung am Heinz Nixdorf Institut der Universität Paderborn. Seine Forschungsschwerpunkte liegen in den Bereichen Virtual & Augmented Reality, Virtual Engineering sowie Virtual Prototyping und Simulation. Herr Berssenbrügge studierte Informatik an der Universität Paderborn und promovierte am Heinz Nixdorf Institut mit einem Thema zum Virtual Prototyping von Scheinwerfersystemen.

.

Punktebasiertes hierarchisch-probabilistisches Level of Detail

Dipl.-Inf. Thomas Preidel,
Dipl.-Inf. Steffen Cersowsky,
Dipl.-Inf. Ronny Krüger,
Dr.-Ing. Markus Färber

Softvise GmbH
Ehrenbergstraße 11, 98693 Ilmenau
Tel. +49 (0) 3677 / 67 97 700, Fax. +49 (0) 3677 / 67 97 799
E-Mail: TP@softvise.de

Zusammenfassung

CAD-Modelle wachsen stetig in ihrer Größe und Komplexität. Detailreiche Teile überfordern bei der Visualisierung die Grafikkarte bzw. werden nicht korrekt wiedergegeben. Zur Milderung dieses Problems wird ein Level-of-Detail-Verfahren benötigt.

Die Eignung eines LoD-Verfahrens im produktiven Umfeld wird nicht nur von der erzielbaren Darstellungsqualität bestimmt, sondern auch von der Integrierbarkeit in bestehende Systeme und Prozesse. Praxistaugliche Verfahren erfordern geringe Vorverarbeitungszeiten und sparsame Speicherverwendung. In dieser Arbeit wird ein punktebasiertes hierarchisch probabilistisches Level-of-Detail-Verfahren vorgestellt, das diese Anforderungen erfüllt.

Ausgehend von weitestgehend geschlossenen CAD-Modellen wird für jede Oberfläche eine fixe Anzahl homogen verteilter Punkte per Zufallsexperiment generiert.

Das Gesamtmodell wird in einer hierarchischen räumlichen Datenstruktur (R*-Baum-Variante) organisiert, wodurch sich räumlich lokale Objektgruppen ergeben, für die Punktewolken generiert werden. Teilmengen dieser Punktewolken werden schrittweise auf höhere Ebenen des R*-Baums propagiert, wo sie mit den Teil-Punktewolken benachbarter Objektgruppen zusammengefasst werden, bis eine Punkt-Repräsentation des Gesamtmodells im Wurzelknoten des Baums entsteht.

Für die Visualisierung ist zu entscheiden, ab wann die Darstellung der originalen Objekte unnötig und stattdessen die passende Punktewolke gerendert wird. Aufgrund der Arbeitsweise der GPU-Rendering-Algorithmen fügen sich dreiecks- und punktbasierte Visualisierung nahtlos im Bild zusammen.

Schlüsselworte

Level of Detail, Point-Rendering, R*-Baum, Visualisierung, große Datenmengen

Point-based Hierarchical Probabilistic Level-of-Detail Method

Abstract

The size and complexity of CAD models grow steadily. Even with modern graphics cards it is hard or even impossible to visualize parts with many details. In order to solve this problem level-of-detail methods are needed.

For a level-of-detail technique to be useful in an production environment it is not only required to achieve good visual quality, it must also be integrable with existing systems and processes. Practical solutions need to have low preprocessing costs and a small memory footprint when used in a rendering system. In this paper we present a point-based hierarchical probabilistic level-of-detail method that satisfies these requirements.

Based on the assumption that most objects have a closed surface we generate a fixed number of evenly distributed points by randomly sampling the surface.

The complete model is stored in a spatial tree data structure (variation of an R*-tree). The tree organizes the objects into groups for which we build small point clouds. Those point clouds are then combined and reduced while propagating them upwards in the tree. The propagation stops at the root giving a point representation of the complete model.

In the renderer we need to decide when the original geometry is not needed and can therefore be replaced with its less detailed point cloud. The rendering algorithm makes sure that points and triangles can be mixed seamlessly.

Keywords

Level-of-detail, point rendering, R*-tree, visualization, big data

1 Einführung und Motivation

Viele Jahre nach den ersten Arbeiten über punktebasierte Visualisierung ist die Motivation für ihren Einsatz noch immer dieselbe: Dreiecksbasierte 3D-Visualisierung ist nicht effizient genug, um ein detailliertes 3D-Modell auf Desktop-Hardware mit Echtzeitgeschwindigkeit darzustellen.

Auch moderne Grafikkarten müssen jedes Dreieck verarbeiten, um zu bestimmen, welche Fragmente eines Dreiecks für eine bestimmte Ansicht sichtbar sind und welchen Beitrag sie zum Bild leisten.

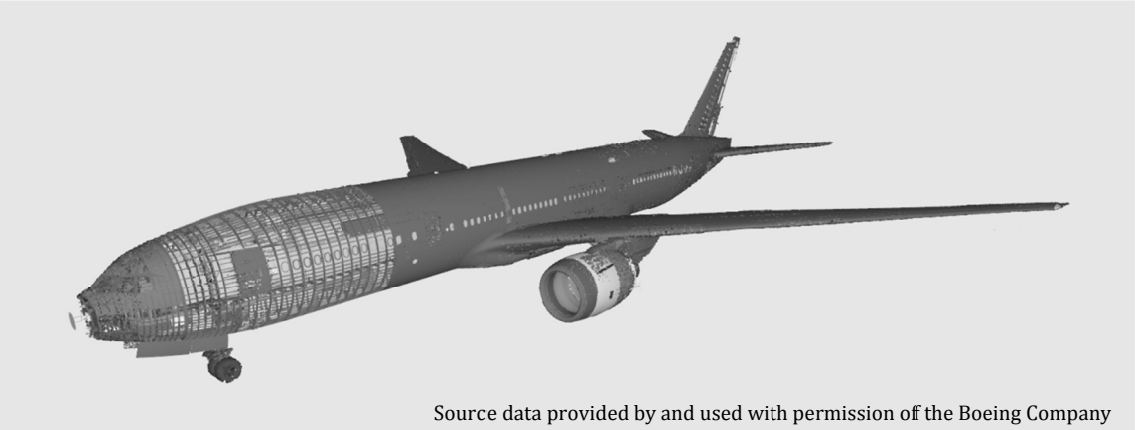


Bild 1: 3D-Modell einer Boeing 777 (ca. 350 Millionen Dreiecke)

Das im Bild 1 dargestellte Modell eines Boeing-777-Verkehrsflugzeugs besteht aus knapp 350 Millionen Dreiecken. Für eine Full-HD-Ansicht mit ca. 2 Millionen Pixeln sind zu jedem Zeitpunkt nur ungefähr ein Hundertstel der Dreiecke relevant. Mit Hilfe von drei Techniken, die teils auf der CPU, teils auf der GPU implementiert werden können, lässt sich dieser Anteil abschätzen:

- Frustum Culling:
  Welche Dreiecke liegen im aktuellen Blickfeld der Kamera?
- Occlusion Culling:
  Welche Dreiecke sind durch andere Dreiecke verdeckt?
- Level of Detail:
  Welche Dreiecke sind zu klein und sollten durch eine zusammenfassende approximative Darstellung ersetzt werden?

Vor allem das Level-of-Detail-Problem ist schwierig zu lösen. Während Polygonmodelle, die in Computerspielen, Simulatoren u. ä. verwendet werden, so entworfen werden können, dass Zusammenfassungen oder Ersatzgeometrien von vornherein zur Verfügung stehen, ist dies bei Modellen, die aus CAD-Systemen exportiert werden, sehr oft nicht der Fall.

Visualisierungssysteme, die diese Art von Modellen visualisieren können, generieren deshalb explizite Level-of-Detail-Darstellungen [GM04, PC12] oder behelfen sich mit dem Weglassen kleiner Objekte [Aut14-ol] bzw. kleiner Objektteile [BHP07].

In einer Ansicht klein erscheinende Objekte oder Dreiecke zu vernachlässigen, ist nur für die Modelle ein gangbarer Weg, in denen die Dreiecksgröße sehr stark schwankt. Der Ansatz scheitert jedoch regelmäßig für Modelle, in denen größere Flächen aus vielen ähnlich großen Dreiecken zusammengesetzt sind, zum Beispiel im Falle tesselierter Freiformflächen.

Es ist für diese Objekte zwingend notwendig, auf eine explizite detailreduzierte Repräsentation zurückzugreifen.

Sie muss in einem Vorverarbeitungsschritt generiert werden. Da in der Praxis immer die neuesten Daten zur Verfügung stehen sollen und häufig sehr kurzfristig mit den Daten gearbeitet wird, können nur Vorverarbeitungszeiten im Bereich von Minuten akzeptiert werden.

Dank der in den letzten Jahren zur Praxisreife gebrachten Visualisierungstechniken für sehr große Punktewolken, zum Beispiel für Laser-Scans mit Milliarden von Punkten [Euc14-ol], ist Point Rendering auch für Level of Detail von Polygonmodellen wieder interessant geworden.

Mit Point Rendering lässt sich gute Darstellungsqualität mit geringem Zeit- und Speicheraufwand für Vorverarbeitung und Rendering verbinden.

2 Vorbetrachtungen

2.1 Existierende Verfahren

Für die Darstellung von dreiecksbasierten 3D-Modellen mit Hilfe von Punkten wurden zahlreiche Verfahren entwickelt.

Bereits in der Pionierarbeit von Levoy und Whitted [LW85] wurde eine Methode vorgestellt, um Punkte eines Polygonmodells abzutasten und zu rendern. Wenngleich die in dieser Arbeit genutzte Render-Technologie inzwischen überholt ist, werden einige grundlegende Ideen, darunter die zufällige Anordnung der abgetasteten Punkte in einer sequenziell zu rendernden Liste, im hier vorgestellten Verfahren noch immer genutzt.

Das Verfahren von Pfister et al. [PZB+00] beschreibt erstmals, wie Punktrepräsentationen von komplexen dreiecksbasierten Szenen zur Detailverminderung bei der Visualisierung eingesetzt werden können. Das System berechnet hierzu Schnittpunkte von achsenparallelen Strahlen mit den Dreiecken. Da die Strahlen im Gitter angeordnet sind, lassen sich die gefundenen Schnittpunkte nach einer Diskretisierung sehr einfach in einem Octree [Sam95] ablegen. Mit einer Bottom-Up-Strategie werden die Punkte in

den Blattknoten zusammengefasst und nach oben propagiert. Zur Visualisierung werden in Abhängigkeit von der Entfernung eines Baumknotens zur Kamera die Knoten ausgewählt, die den entsprechenden Bildschirmbereich mit dem besten Verhältnis von Renderaufwand zu Renderqualität füllen.

Rusinkiewicz und Levoy [RL00] beschreiben zur gleichen Zeit ein etwas anders arbeitendes Verfahren, das sie für die Darstellung von detailreich gescannten realen Objekten einsetzen. Anstelle eines Octrees verwenden sie eine Bounding-Sphere-Hierarchie, und anstelle von einfachen Punkten rendern sie für jedes Sample eine kompliziertere Fläche, die als Splat bezeichnet wird.

In jüngerer Zeit wurde die Technik des Point Rendering weiter ausgebaut. Es kann jetzt als etablierter, in der Praxis bewährter Stand der Technik gelten [GEM+13, GP07]. Insbesondere ist hervorzuheben, dass die Visualisierung auch für Punktewolken möglich ist, die nicht in Hauptspeicher bzw. Grafikkartenspeicher passen und daher mit Out-of-Core-Algorithmen behandelt werden müssen [Euc14-ol, GM04, RD10, Sch14].

Für das vorgestellte Level-of-Detail-Verfahren kann somit auf eine ganze Reihe etablierter Methoden zurückgegriffen werden. Ein sehr großer Unterschied besteht jedoch in den Datenstrukturen, die zum Einsatz kommen, um das Gesamtmodell zu verwalten.

2.2 Hierarchische Speicherung des Modells

Viele existierende Systeme [BHP07, BZB+00, Sch14] verwenden eine hierarchische Speicherung, bei der eine Zerlegung des Raumes stattfindet, in den die Teile des Modells fallen. Dies geschieht mit Hilfe von kD-Bäumen, Octrees oder ähnlichen In-Memory-Datenstrukturen. Allen ist gemeinsam, dass sie Raumzellen schaffen, in die dann die Modellteile einsortiert werden. Weiterhin sind die Datenstrukturen auf maximale Effizienz bei ihrer Erstellung ausgelegt. Es entstehen dadurch unbalancierte Bäume. Dies bedeutet, dass leere Bereiche schnell gefunden werden können, während die Suche in eng mit Geometrie gefüllten Bereichen höhere Komplexität hat.

Im System von Rusinkiewicz und Levoy [RL00] wird eine Bounding-Sphere-Hierarchie verwendet. Diese unterteilt nicht primär den Raum, sondern baut eine adaptive Datenstruktur um die zu verwaltenden Punkte. Die Punktemenge wird beginnend mit allen Punkten rekursiv in zwei Hälften geteilt, wobei immer entlang der längsten Kante der Bounding-Box geteilt wird. So entsteht im Kern ein balancierter kD-Baum, in dessen Blattknoten einzelne Punkte gespeichert sind. In den inneren Knoten wird die umschließende Kugel aller Kindknoten gespeichert. Diese Unterteilungsstrategie erzeugt stark überlappende Raumzellen, was sich negativ auf die Komplexität der Suche in der Datenstruktur auswirkt. Diese ist auch gar nicht sinnvoll möglich, da die Information, welcher Punkt zu welchem Objekt gehört, verworfen wurde. Level of Detail ist im Ansatz von Rusinkiewicz und Levoy ebenfalls nicht vorgesehen.

2.2.1 R*-Baum

Das in dieser Arbeit vorgestellte System verwendet eine Variante des R*-Baums [BKS+90] zur Speicherung des Modells. Zu Beginn werden die Objekte des Modells in so genannte Objektfragmente zerlegt. Enthält ein Objekt weniger als eine festgelegte Anzahl von Dreiecken, wird es direkt in ein Objektfragment umgewandelt, anderenfalls wird es in mehrere Fragmente räumlich beieinander liegender Dreiecke geteilt. Die Fragmente werden Stück für Stück in den R*-Baum eingefügt entsprechend den in [BKS+90] beschriebenen Regeln. Im Ergebnis entsteht ein balancierter Baum.

In den Blattknoten des R*-Baumes werden Verweise auf die Geometrien gespeichert sowie die zugehörigen achsenparallelen Bounding-Boxen. In den inneren Knoten werden Verweise auf die Kindknoten abgelegt sowie die Bounding-Boxen der in den Kindern beginnenden Äste. Somit enthält der Wurzelknoten die Bounding-Box des Gesamtmodells.

Der R*-Baum erzeugt eine Organisationsstruktur für den vom 3D-Modell belegten Raum, die die unterschiedliche Geometriedichte automatisch berücksichtigt. Mit Hilfe geeigneter Heuristiken [BKS+90] werden außerdem Raumzellen erzeugt, die sich nur gering überlappen, wodurch die Komplexität der Suche im R*-Baum gering gehalten werden kann. Bereiche, in denen keine Geometrie ist (eng anliegende Bounding-Boxen vorausgesetzt), werden vom R*-Baum gar nicht erst erfasst.

Der R*-Baum ist als Out-of-Core-Datenstruktur implementiert, das heißt, dass die Größe des R*-Baums nur vom Festplattenplatz begrenzt wird, nicht aber von der Größe des Hauptspeichers.

Auf der Ebene 0, der Ebene der Blattknoten, werden die Objektfragmente mit ihrer Dreiecksgeometrie gespeichert. Auf den darüber liegenden Ebenen wird pro Knoten eine Punktewolke gespeichert, die als Level-of-Detail-Repräsentation für den unterhalb des Knoten liegenden Teilbaum dient. Die Anzahl an Punkten in der Punktewolke wird auf die maximale Anzahl von Dreiecken eines Objektfragments begrenzt. Siehe hierzu Abschnitt 3.

2.3 Visualisierung

Für das Rendern des Modells wird auf den R*-Baum zugegriffen, der für die jeweilige Ansicht relevante Teil bestimmt und die darin gespeicherte Geometrie mit Out-of-Core-Techniken in Hauptspeicher bzw. Grafikkartenspeicher geladen. Im Anschluss erfolgt die Visualisierung mit Hilfe von Geometry-, Vertex- und Pixelshadern.

2.3.1 Frustum und Occlusion Culling

Von der aktuellen Ansicht können zunächst die Objektfragmente ausgeschlossen werden, deren Bounding-Boxen vollständig außerhalb des View Frustums liegen. Diese Entscheidung kann sehr effizient durch Traversierung des R*-Baums getroffen werden.

Im Laufe der nächsten Frames werden mit Hilfe von Hardware-Occlusion-Tests auch noch Äste des R*-Baums ausgeschlossen, deren Bounding-Boxen garantiert von bereits gerenderter Geometrie verdeckt werden.

2.3.2 Level of Detail

Bild 2 zeigt einen symbolischen Ausschnitt des R*-Baums nach dem Beschnitt durch das Frustum Culling. Man kann nun davon ausgehen, dass nahezu jeder Blattknoten einen Beitrag zur aktuellen Ansicht leistet. Es ist aber immer noch möglich, dass die sich ergebende Anzahl von zu rendernden Dreiecken Größenordnungen über der Pixel-Auflösung des Bildschirms liegt. Dies ist dann der Fall, wenn die Dreiecke aufgrund großer Entfernung zur Kamera auf sehr kleine Bildschirmbereiche projiziert werden.

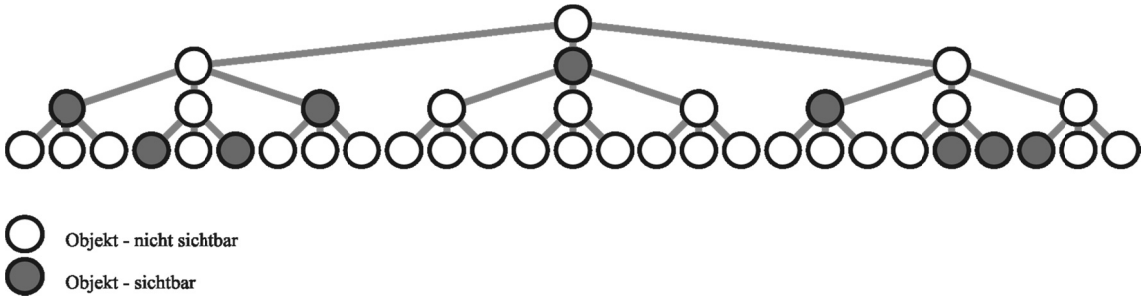


Bild 2: Ausschnitt eines R-Baums mit Markierung der sichtbaren Knoten*

Der R*-Baum erlaubt es, auf einfache Weise zu bestimmen, welche Objektfragmente so klein werden, dass an ihrer Stelle eine detailreduzierte Repräsentation gerendert werden sollte.

Hierzu wird die Bounding-Box jedes Knotens in den Bildraum projiziert und die Anzahl der belegten Pixel berechnet [ST99]. Da in jedem Knoten maximal n Punkte einer Level-of-Detail-Repräsentation oder n Dreiecke der originalen Geometrie gespeichert sind, kann immer dann die Punktrepräsentation eines Astes gerendert werden, wenn die projizierte Bounding-Box weniger als $\frac{1}{2}n$ Pixel belegt. Auf diese Weise wird das Abtasttheorem näherungsweise berücksichtigt.

Die Berechnung der projizierten Bounding-Box-Größe geschieht bereits während der oben beschriebenen Traversierung des R*-Baums. Entsprechend sind im Bild 2 die Knoten markiert, deren Detailstufe genügt, um eine repräsentative Wiedergabe des 3D-Gesamtmodells aus der gegebenen Kamera-Perspektive zu erreichen. Äste ohne mar-

kierte Knoten können sich nach einigen Frames ergeben, wenn erfolgreiches Occlusion Culling die Knoten als verdeckt erkennt.

Bild 3 zeigt, wie das Rendering-System im Laufe mehrerer Frames zur optimalen Visualisierung voranschreitet. Im ersten Frame wird nur die Level-of-Detail-Repräsentation des Gesamtmodells gerendert. Nach erfolgreichem Nachladen der Level-of-Detail-Geometrie für die nächste R*-Baum-Ebene werden diese gerendert, bis schließlich nach einiger Zeit die gesamte relevante Geometrie auf dem Bildschirm dargestellt werden kann.

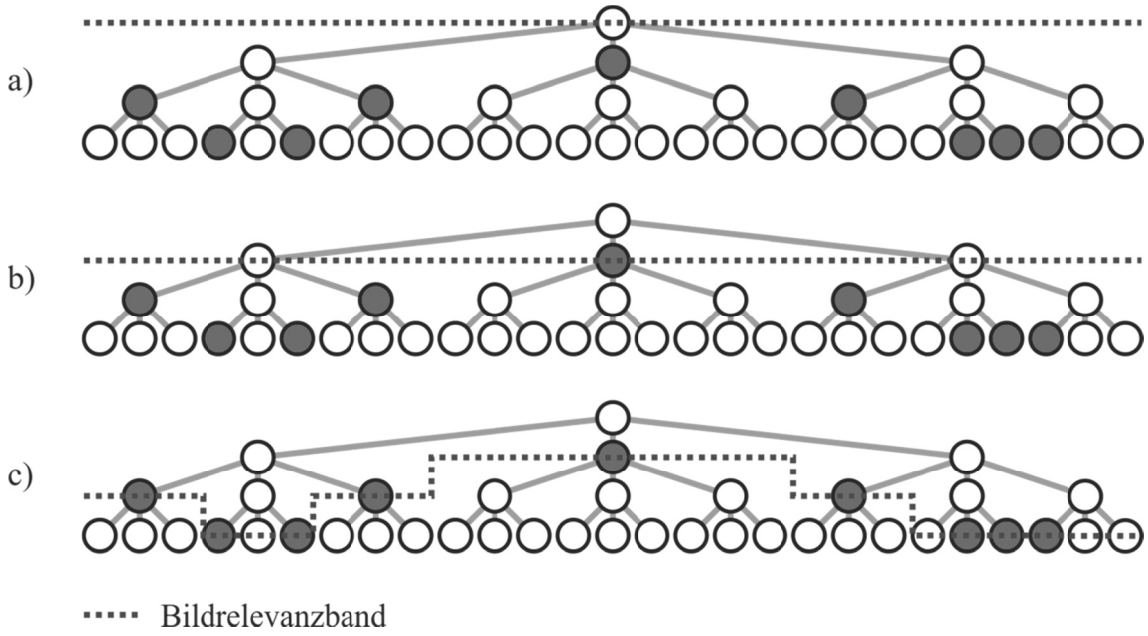


Bild 3: Anpassung des Bildrelevanzbandes für eine bestimmte Ansicht a) Ausgangspunk b) nach erster Anpassung c) Endlage

Der zu einem Zeitpunkt relevante Teil des Gesamtmodells bildet das in Bild 3 rot dargestellte Bildrelevanzband. Während der Interaktion mit dem 3D-Modell und der damit verbundenen Veränderung von Position und Orientierung der Kamera wird die beschriebene Filterung des R*-Baums kontinuierlich wiederholt. Dadurch verändert sich das Bildrelevanzband, und der Renderer folgt mit der entsprechenden Darstellung.

Abhängig von der Modellgröße, Leistungskraft der Grafikkarte, Speicherausbau des Systems und Geschwindigkeit der Festplatte wählt der Renderer gegebenenfalls eine oberhalb des optimalen Bildrelevanzbandes liegende Näherung.

3 Probabilistische Punkt-Repräsentation

Im vorangegangenen Abschnitt wurde deutlich gemacht, wie wichtig die detailreduzierte Darstellung von Teilen des Modells für das relevanzorientierte Rendering ist und dass

diese mit Hilfe der Speicherung des Modells in einem R*-Baum und geeigneten out-of-core-fähigen Algorithmen erreicht werden kann.

Die Analyse der existierenden Arbeiten und die Durchführung verschiedener Tests zeigte, dass die Detailreduzierung durch Rendern von Punkten bzw. Splats die Anforderungen des gewählten Rendering-Ansatzes sehr gut erfüllt. Insbesondere kann die Punkte-Repräsentation eines sehr großen 3D-Gesamtmodells wie das der Boeing 777 in ungefähr 30 Minuten erstellt werden.

3.1 Repräsentation eines Objektfragments durch Punkte

Die Objekte bzw. Objektfragmente eines CAD-Modells besitzen charakteristische Eigenschaften, die den Einsatz von Punkten als Level-of-Detail-Approximationen erleichtern:

1. Objekte haben oft geschlossene Oberflächen und enthalten kaum innere Details.
2. Objekte haben in der Regel niederfrequente Materialeigenschaften, oft sogar nur eine einzige Farbe.

Es kann daher angenommen werden, dass ein gleichmäßiges Abtasten der gesamten Oberfläche eines Objektfragments die relevanten Informationen erfasst, sodass deren Betrachtung aus der Entfernung ein ähnliches oder gar identisches Bild ergibt, als würden die Originaldreiecke gerendert.

Für jedes Objektfragment wird eine Punktewolke in folgender Form erstellt: Zunächst werden zufällig Dreiecke entsprechend ihrem Anteil an der Oberfläche des Fragments gewählt. Für jedes Dreieck wird anschließend ein zufälliger Punkt im Innern des Dreiecks bestimmt.

3.1.1 Bestimmung der abzutastenden Dreiecke

Zunächst wird der Anteil der Fläche jedes Dreiecks an der Gesamtfläche aller Dreiecke bestimmt. Anschließend werden Zufallsexperimente durchgeführt, um jeweils ein Dreieck auszuwählen, für das später ein Punkt abzutasten sein wird. Es wird dazu eine gleichverteilte Zufallszahl p bestimmt und es wird das Dreieck mit dem Index i ausgewählt, für das gilt: $p_{i-1} \leq p \leq p_i$.

Ein Dreieck kann mehrmals gewählt werden. Entsprechend werden in diesem Dreieck mehrere Punkte abgetastet. Auf diese Weise wird sichergestellt, dass alle Punkte gleichmäßig über die gesamte Oberfläche des Objekts verteilt sind.

Die Anzahl der Zufallsexperimente zur Dreiecksauswahl und damit die Anzahl der Punkte wird im Hinblick auf die spätere Zusammenfassung von Level-of-Detail-Repräsentationen mehrerer Objektfragmente gewählt (siehe Bild 4). Siehe dazu auch Abschnitt 3.2.

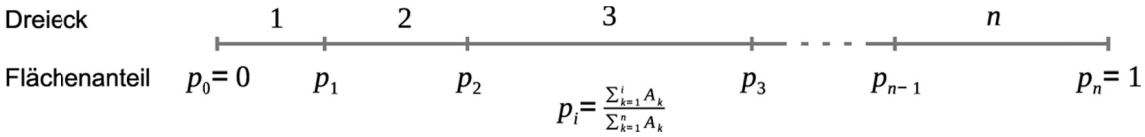


Bild 4: Zufällige Auswahl eines Dreiecks gewichtet nach der Dreiecksfläche

3.1.2 Bestimmung eines abzutastenden Punktes im Dreieck

Für jedes ausgewählte Dreieck ABC wird mit dem in [OFC+02] beschriebenen Verfahren ein zufälliger Punkt P im Innern des Dreiecks bestimmt:

$$P = \left(1 - \sqrt{r_1}\right)A + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C$$

r_1 und r_2 sind dabei gleichverteilte Zufallszahlen zwischen 0 und 1.

Alle abgetasteten Punkte für ein Fragment werden zunächst gespeichert. Die Punkte liegen aufgrund der zufälligen Dreiecksauswahl gleichverteilt vor. Somit ist auch jede Teilmenge dieser Punkte gleichverteilt über der Oberfläche des Fragments. Dies wird für das Zusammenfassen von Fragmenten genutzt.

Durch die zufällige Abtastung der Objektfragmente werden Abtast-Artefakte wie zum Beispiel der Moiré-Effekt vermieden. Auch sorgt der Zufall automatisch für eine gleichmäßige Erfassung beliebig geformter Objektfragmente, ohne dass eine aufwendige Analyse notwendig ist.

3.2 Mischen von Punktfragmenten

Die beschriebenen Punktrepräsentationen für Objektfragmente werden im Visualisierungsverfahren nicht direkt genutzt. Bei der Abarbeitung des Bildrelevanzbandes werden für Knoten, die auf Objektfragmentebene liegen, immer die Dreiecke gerendert. Anderenfalls werden Punktfragmente gerendert.

Für ihre Generierung werden die in den Kindern des Knotens gespeicherten Punktfragmente herangezogen. Zunächst wird für jedes Kind-Punktfragment der Anteil f_i seiner Dreiecksoberfläche an der Gesamt-Dreiecksoberfläche aller Kindfragmente bestimmt. Anschließend werden die ersten $n_i = f_i \cdot n$ Punkte des Kindfragments ausgewählt und in das neu erzeugte Punktfragment kopiert.

Das neue Fragment enthält nun – analog zu den Kind-Punktfragmenten – n Punkte, die alle Objektfragmente repräsentieren, die in den Blattknoten des im Knoten beginnenden R*-Baum-Astes gespeichert sind.

Um die Generierung der oberhalb des Knotens liegenden Punktfragmente auf gleiche Weise ausführen zu können, werden zum Abschluss noch alle Punkte des neu erzeugten Fragments zufällig permutiert.

3.3 Visualisierung von Punktfragmenten

Der Visualisierungsalgorithmus erhält in jedem Frame ein Bildrelevanzband (siehe Bild 5) mit den zugehörigen Daten für Objektfragmente (Dreiecksgeometrie) bzw. Punktfragmente.

Die Punktfragmente enthalten für jeden Punkt zusätzlich zur Position im Objektraum eine Normale, materialspezifische Daten (Farben u. ä.) sowie den Flächeninhalt, den der Punkt belegt. Letzterer wird wie folgt geschätzt: Wird ein Objektfragment mit n Punkten abgetastet, so repräsentiert ein Punkt $\frac{1}{n}$ der Summe der Flächeninhalte der Dreiecke in diesem Fragment.

Position, Normale und Flächeninhalt werden zur Visualisierung einem Geometry-Shader übergeben, der im Objektraum ein Splat in Quadratform erzeugt.

Die weiteren Rendering-Schritte sind für Dreiecke und Splats identisch, sodass die gemischte Darstellung von Dreiecken und Splats ein visuell ansprechendes Bild erzeugt.

Im Bild 5 ist die Boeing 777 aus Bild 1 nochmals wiedergegeben. Anstelle von knapp 350 Millionen Dreiecken des originalen Modells werden Punktrepräsentationen mit links 8.327.455 Punkten und rechts 262.252 Punkten verwendet. Die Bild-Auflösung beträgt jeweils 914 x 914 Pixel.

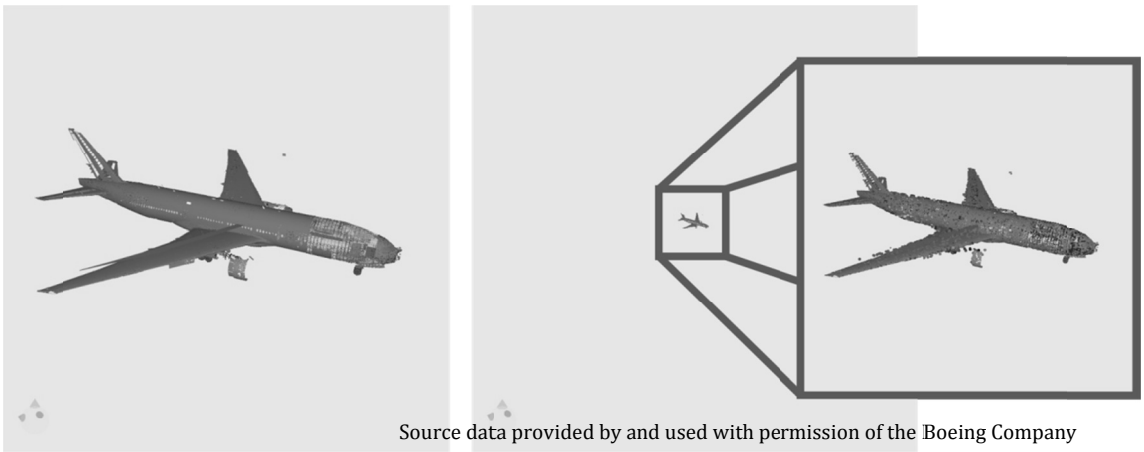


Bild 5: Visualisierung der Boeing 777 mit Hilfe von Punktfragmenten



Bild 6: Stanford Lucy[20] in verschiedenen LoD-Repräsentationen, Orginal: 14.027.872 Punkte, 28.055.742 Dreiecke, Auflösung 914x914 a) Darstellung mit 11.564.988 Dreiecken, b) 68.173 Punkte (vergrößerte LoD Darstellung links oben im Bild im Größenvergleich zu a, und c) gröbste LoD-Stufe mit 4096 Punkten

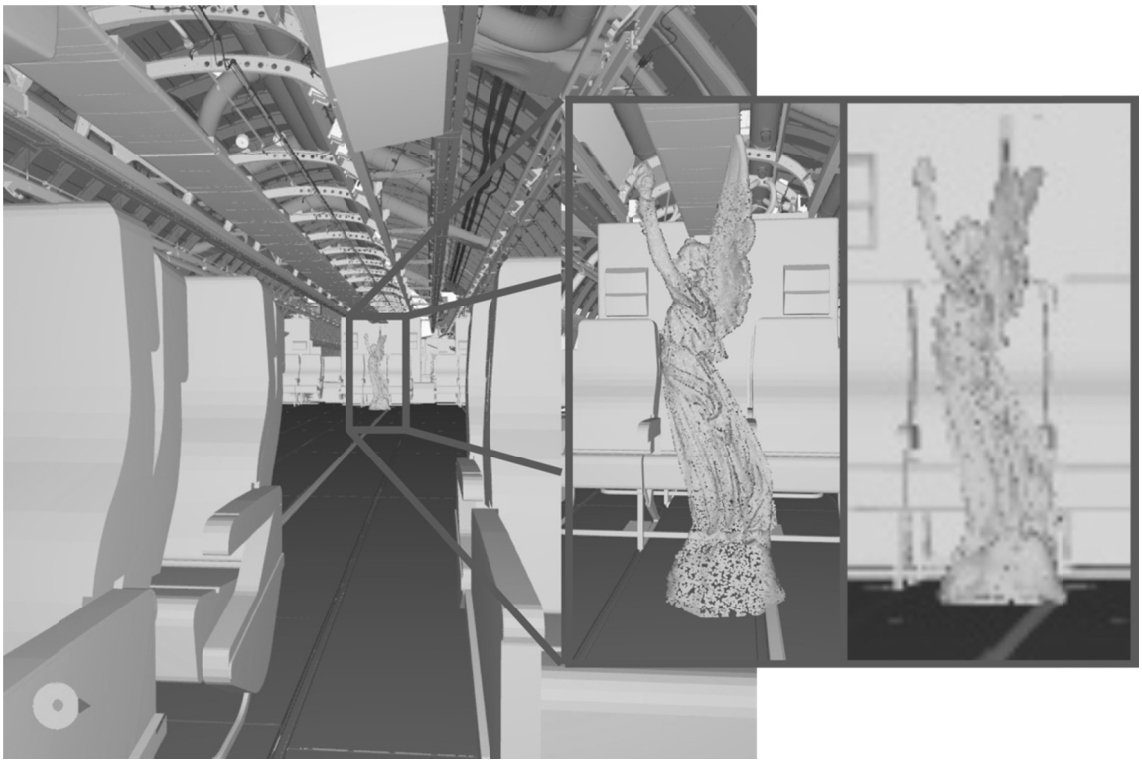


Bild 7: Kombination von Boeing und Lucy; vergrößerte Darstellung durch Heranzoomen an die LoD-Repräsentation bzw. durch Vergrößerung des Orginalbildes

4 Resümee und Ausblick

Es wurde ein System zur Visualisierung sehr großer dreiecksbasierter 3D-CAD-Modelle vorgestellt, das zur Detailreduzierung approximative Punkt-Repräsentationen einsetzt. Das Modell wird in Objektfragmente zerlegt und in einer Variante des R*-Baums ge-

speichert. Neben den Dreiecksgeometrien der Objektfragmente in den Blattknoten werden in den inneren Knoten Punktfragmente abgelegt.

Die Erzeugung der Punktfragmente erfolgt schnell (Komplexität ist linear) und speichereffizient durch Monte-Carlo-Sampling der Dreiecksgeometrien und anschließendem Ausdünnen und Mischen der entstehenden Punktewolken.

Punkte und Dreiecke können durch weitgehend identische Visualisierungstechniken mit sehr guter Darstellungsqualität und in Echtzeitgeschwindigkeit gerendert werden.

Alle Algorithmen und Datenstrukturen sind out-of-core-fähig, sodass die Visualisierung nicht durch die Größe von Haupt- und Grafikkartenspeicher begrenzt ist.

In Zukunft soll der Visualisierungsansatz auf sich dynamisch ändernde Objekte ausgedehnt werden. Eine einfache Erweiterung zur Visualisierung von Laserscans erscheint ebenfalls naheliegend wie auch die weitere Beschleunigung der Datenerzeugung durch Parallelisierung.

Literatur

[Aut14-ol] AUTODESK, INC.: Projektprüfungssoftware Navisworks®. http://www.autodesk.de/products/navisworks, abgerufen am 27.11.2014.

[BKS+90] BECKMANN, NORBERT; KRIEGEL, HANS-PETER; SCHNEIDER, RALF; SEEGER, BERNHARD: The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. Proceedings of ACM SIGMOD 1990, pp. 322-331.

[BHP07] BRÜDERLIN, BEAT; HEYER, MATHIAS; PFÜTZNER, SEBASTIAN: Interviews3D: A Platform for Interactive Handling of Massive Data Sets. IEEE Computer Graphics and Applications, 27(6):48-59, 2007.

[Euc14-ol] EUCLIDEON PTY LTD: Unlimited Detail Engine. http://en.wikipedia.org/wiki/Euclideon, abgerufen am 27.11.2014.

[GM04] GOBBETTI, ENRICO; MARTON, FABIO: Layered Point Clouds: A Simple and Efficient Multiresolution Structure for Distributing and Rendering Gigantic Point-sampled Models. Computers & Graphics, 28(6):815-826, 2004.

[GM08] GOBBETTI, ENRICO; MARTON, FABIO: Far Voxels: A Multiresolution Framework for Interactive Rendering of Huge Complex 3D Models on Commodity Graphics Platforms. Proceedings of ACM SIGGRAPH Asia 2008 courses, pp. 32:1-32:8.

[GEM+13] GOSWAMI, PRASHANT; EROL, FATIH; MUKHI, RAHUL; PAJAROLA, RENATO; GOBBETTI, ENRICO: An Efficient Multi-resolution Framework for High Quality Interactive Rendering of Massive Point Clouds using Multi-way kd-trees. The Visual Computer, 29:69-83, 2013.

[GP07] GROSS, MARKUS; PFISTER, HANSPETER: Point-Based Graphics. Morgan Kaufmann, 2007.

[GD98] GROSSMAN , J. P.; DALLY, WILLIAM J.: Point Sample Rendering. Proceedings of EUROGRAPHICS 1998, pp 181-192.

[LW85] LEVOY, MARK; WHITTED, TURNER: The Use of Points as a Display Primitive. Technical Report 85-022, University of North Carolina at Chapel Hill, 1985.

[OFC+02] OSADA, ROBERT; FUNKHOUSER, THOMAS; CHAZELLE, BERNARD; DOBKIN, DAVID: Shape Distributions. ACM Transactions on Graphics 21(4):807-832, 2002.

[PSL05] PAJAROLA, RENATO; SAINZ, MIGUEL; LARIO, ROBERTO: Xsplat: External Memory Multiresolution Point Visualization. Proceedings of IASTED International Conference on Visualization, Imaging and Image Processing, 2005, pp. 628-633.

[PC12] PENG, CHAO; CAO, YONG: A GPU-based Approach for Massive Model Rendering with Frame-to-frame Coherence. Computer Graphics Forum 31(2pt2):393-402, 2012.

[PZB+00] PFISTER, HANSPETER; ZWICKER, MATTHIAS; BAAR, JEROEN VAN; GROSS, MARKUS: Surfels: Surface Elements as Rendering Primitives. Proceedings of ACM SIGGRAPH 2000, pp. 335-342.

[RD10] RICHTER, RICO; DÖLLNER, JÜRGEN: Out-of-Core Real-Time Visualization of Massive 3D Point Clouds. Proceedings of AFRIGRAPH 2010, pp. 121-128.

[RL00] RUSINKIEWICZ, SZYMON; LEVOY, MARC: QSplat: A Multiresolution Point Rendering System for Large Meshes. Proceedings of ACM SIGGRAPH 2000, pp. 343-352.

[Sam95] SAMET, HANAN: Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS. Addison-Wesley, 1995.

[Sch14] SCHEIBLAUER, CLAUS: Interactions with Gigantic Point Clouds. Dissertation, Technische Universität Wien, 2014.

[ST99] SCHMALSTIEG, DIETER; TOBLER, ROBERT F.: Fast Projected Area Computation for 3D Bounding Boxes. Journal of Graphics Tools 4(2):37-43, 1999.

[Sta15] STANFORD UNIVERSITY COMPUTER GRAPHICS LABORATORY: The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep, abgerufen am 10.03.2015

[TRB+04] ZWICKER, MATTHIAS; RÄSÄNEN, JUSSI; BOTSCH, MARIO; DACHSBACHER, CARSTEN; PAULY, MARK: Perspective Accurate Splatting. Proceedings of Graphics Interface Conference, London, ON, Canada, 2004, pp. 247-254.

Autoren

Thomas Preidel ist Diplom-Informatiker und als Geschäftsführer der Softvise GmbH in Ilmenau tätig. Nach seinem Diplom an der Technischen Universität Ilmenau war er zunächst wissenschaftlicher Mitarbeiter am Fachgebiet Grafische Datenverarbeitung der TU Ilmenau, später Software-Entwickler bei der 3DInteractive GmbH in Ilmenau, bis er im Jahr 2014 mit seinen Kollegen die Softvise GmbH gründete. Während seiner Zeit bei 3DInteractive war er Projektleiter in den BMBF-Forschungsprojekten AVILUS und ARVIDA. Zu seinen speziellen Kompetenzen gehört die Entwicklung von Verfahren zur Bildsynthese sowie die Handhabung und Verarbeitung von CAD-Daten.

Steffen Cersowsky ist ebenfalls Diplom-Informatiker. Nach dem Diplom an der TU Ilmenau hat er als wissenschaftlicher Mitarbeiter am Fachgebiet Grafische Datenverarbeitung gearbeitet. Nach einer mehrjährigen Tätigkeit als Software-Entwickler bei 3DInteractive ist er seit 2014 Mitarbeiter und Gesellschafter der Softvise GmbH. Er entwickelt unter anderem räumliche Datenstrukturen und effiziente Visualisierungstechnologien.

Ronny Krüger ist wie seine Kollegen Diplom-Informatiker. Er war nach dem Diplom an der TU Ilmenau als wissenschaftlicher Mitarbeiter am Fachgebiet Grafische Datenverarbeitung und anschließend als Software-Entwickler bei der 3DInteractive GmbH in

Ilmenau angestellt. Er ist Mitgründer der Softvise GmbH und arbeitet seit 2014 in diesem Unternehmen als Geschäftsführer und Software-Entwickler. Ronny Krüger ist Fachmann für Architekturen interaktiver 3D-Grafiksysteme mit den Schwerpunkten Out-of-Core-Algorithmen sowie Interaktionstechniken für 3D-Modellierung.

Markus Färber Markus Färber hat an der Technischen Universität Dresden Informatik studiert. Nach dem Abschluss als Diplom-Informatiker arbeitete er als wissenschaftlicher Mitarbeiter am Fachgebiet Grafische Datenverarbeitung der TU Ilmenau. Dort war er in verschiedenen Projekten beschäftigt, zuletzt als wissenschaftlicher und geschäftsführender Koordinator der Graduiertenschule für Bildverarbeitung und Bildinterpretation in Ilmenau und Jena. 2014 wurde er an der TU Ilmenau promoviert. Er ist ebenfalls Gesellschafter der Softvise GmbH und als Software-Entwickler tätig. Seine Kompetenzschwerpunkte sind unter anderem räumliche Datenstrukturen, approximative Visualisierungstechniken sowie Algorithmen zum geometrischen Modellieren.

VR für Fabrikplanungsprozesse

Ein Konzept zur Integration von Virtual Reality Anwendungen zur Verbesserung des Informationsaustauschs im Fabrikplanungsprozess

Dipl.-Inform. Sebastian Pick
Dipl.-Medieninform. Sascha Gebhardt
Dr. rer. nat. Bernd Hentschel
Prof. Dr. rer. nat. Torsten W. Kuhlen
IT Center, RWTH Aachen University
Seffenter Weg 23, 52074 Aachen
Tel. +49 (0) 241 / 80 24 867
E-Mail: pick@vr.rwth-aachen.de

Dipl.-Math. Rudolf Reinhard
Dipl.-Wirt.-Ing. Christian Büscher
Lehrstuhl Informationsmanagement im Maschinenbau (IMA) der RWTH Aachen
IMA der RWTH Aachen, Dennewartstr. 27, 52068 Aachen
Tel. +49 (0) 241 / 80 91 131
E-Mail: rudolf.reinhard@ima-zlw-ifu.rwth-aachen.de

Toufik Al Khawli, M.Sc.
Dipl.-Phys. Urs Eppelt
RWTH Aachen University
Lehr- und Forschungsgebiet für Nichtlineare Dynamik der Laser-Fertigungsverfahren NLD
Steinbachstr. 15, 52074 Aachen
Tel. +49 (0) 241 / 89 06 80 60
E-Mail: toufik.al.khawli@ilt.fraunhofer.de

Hanno Voet, M.Sc.
Dipl.-Wirt.-Ing. Julian Utsch, M.Sc.
Werkzeugmaschinenlabor WZL der RWTH Aachen
Abteilung Fabrikplanung
Manfred-Weck Haus, Steinbachstraße 19, 52074 Aachen
Tel. +49 (0) 241 / 80 20 531
E-Mail: h.voet@wzl.rwth-aachen.de

Zusammenfassung

Die Planung von Fabriken stellt einen äußerst heterogenen Prozess dar, an dem verschiedene Expertengruppen gleichzeitig beteiligt sind. Ein häufig auftretendes Problem stellt in diesem Zusammenhang die Kommunikation zwischen den einzelnen Expertengruppen dar, was unter anderem auf unterschiedliches Domänenwissen zurückzuführen ist. Da Entscheidungen innerhalb einer Domäne jedoch häufig Auswirkungen auf die anderen haben, ist es wichtig derartige Wechselwirkungen für alle Experten erfassbar zu machen, um den Planungsprozess zu verbessern.

In dieser Arbeit stellen wir vor dem Hintergrund der Virtual Production Intelligence ein Konzept vor, welches die Integration zweier separater Virtual-Reality- und Visualisierungsbasierter Werkzeuge für unterschiedliche Anwendungsbereiche des Planungsprozesses beschreibt. Ziel ist es einen Ansatz zu schaffen, welcher Wechselwirkungen zwischen unterschiedlichen Domänen sichtbar zu machen vermag, um dem oben genannten Problem entgegenzuwirken.

Schlüsselworte

Informationsvisualisierung, Virtuelle Realität, Digitale Fabrik, Fabrikplanung, Prozessoptimierung

A Concept for the Integration of Virtual Reality Applications to Improve the Information Exchange within the Factory Planning Process

Abstract

Factory planning is a highly heterogeneous process that involves various expert groups at the same time. In this context, the communication between different expert groups poses a major challenge. One reason for this lies in the differing domain knowledge of individual groups. However, since decisions made within one domain usually have an effect on others, it is essential to make these domain interactions visible to all involved experts in order to improve the overall planning process.

In this paper, we present a concept that facilitates the integration of two separate virtual-reality- and visualization analysis tools for different application domains of the planning process. The concept was developed in context of the Virtual Production Intelligence and aims at creating an approach to making domain interactions visible, such that the aforementioned challenges can be mitigated.

Keywords

Information visualization, virtual reality, virtual factory, factory planning, process optimization

1 Einleitung

Heutzutage sind Virtuelle Realität (VR) und Visualisierung etablierte Werkzeuge zur Unterstützung diverser komplexer Aufgaben, wie auch dem Fabrikplanungsprozess. Eine der größten Herausforderungen dieses Prozesses liegt im Informationsaustausch zwischen Expertengruppen, welche sich mit unterschiedlichen Aufgabenbereichen des Planungsprozesses befassen. Ein solcher Austausch ist notwendig, damit Wechselwirkungen verschiedener Teilaspekte sichtbar gemacht und somit bei der Planung berücksichtigt werden können. Um dieses Ziel zu erreichen, ist es jedoch notwendig Lösungen für Teilaspekte in ein Gesamtsystem zu integrieren. Neben dem initialen Fabrikplanungsprozess kann ein solches System auch im Kontext Kontinuierlicher Verbesserungsprozesse (KVPs) eingesetzt werden. Vor diesem Hintergrund präsentieren wir ein Konzept, um existierende Werkzeuge für Teilprobleme unterschiedlicher Aufgabenbereiche in eine integrative Lösung zu überführen.

Konkret wird die Integration zweier Werkzeuge aus den Aufgabenbereichen Fabriklayoutplanung und Maschinenkonfiguration beschrieben. Die Anwendung *flapAssist* wurde zur Unterstützung der Fabriklayoutplanung entwickelt. Mit ihr ist es möglich erstellte Fabriklayouts bereits während des Planungsprozesses am Desktop oder in immersiven VR-Systemen, wie etwa CAVEs, zu betrachten. Weiter werden zusätzliche planungsrelevante Informationen, wie etwa Materialflüsse, innerhalb der virtuellen Umgebung angezeigt. Für den Aufgabenbereich Maschinenkonfiguration wurde die Anwendung *memoSlice* entwickelt. Sie erlaubt es, multidimensionale Metamodelle zu explorieren und somit den Parameterraum von Maschinen nach idealen Konfigurationen mit Hinblick auf gewünschte Fertigungskriterien zu durchsuchen und so eine optimale Maschinenkonfiguration zu erzielen. Beide Werkzeuge wurden vor dem Hintergrund der Virtual Production Intelligence entwickelt, deren Ziel es u.a. ist, allen beteiligten Experten planungsrelevante Daten in leicht zugänglicher und verständlicher Form bereitzustellen.

Für beide Teillösungen wurde ein Integrationskonzept entwickelt, welches zur Sichtbarmachung der Wechselwirkungen zwischen den betrachteten Aufgabenbereich eingesetzt werden kann. Zusammenfassend stellen wir zunächst die beiden Teillösungen vor dem Hintergrund der Virtual Production Intelligence vor. Anschließend besprechen wir das Integrationskonzept, wobei auf den Anwendungshintergrund und den resultierenden Mehrwert eingegangen wird.

2 Werkzeuge

Im Folgenden stellen wir zunächst die Virtual Production Intelligence vor, welche den Rahmen für die Entwicklung der beiden VR-Anwendungen *flapAssist* und *memoSlice* bildet. Anschließend widmen wir uns den einzelnen Anwendungen im Detail.

2.1 Virtual Production Intelligence

Die Virtual Production Intelligence (VPI) beschreibt das grundlegende Konzept, auf dem das in dieser Arbeit entwickelte Informationssystem für die Fabrikplanung basiert. Bei der VPI handelt es sich um ein ganzheitliches, integratives Konzept zur kollaborativen Planung von Kernprozessen der Technologie- (Material/Maschine), Produkt-, Fabrik- und Produktionsplanung und der Überwachung und Steuerung der Produktion und Produktentwicklung [RMB+12]. In dem Zusammenhang wird folgendes Verständnis der verwendeten Begriffe zugrunde gelegt:

- **Ganzheitlich**: Adressierung aller Teilprozesse der Produktentwicklung.
- **Integrativ**: Vermeidung der Schaffung neuer, weiterer Standards; stattdessen wird die Nutzung und Zusammenführung vorhandener Lösungsansätze und Potentiale unterstützt.
- **Kollaborativ**: Betrachtung aller im Planungsprozess involvierten Rollen und ihrer Kommunikations- und Übergabeprozesse.

Das Konzept umfasst Methoden und Modelle zur Konsolidierung und Propagation von Daten, die entlang der betrachteten Domäne und des Anwendungsfalls generiert werden. Dabei werden insbesondere Planungsprozesse betrachtet, die in den Bereich der Virtuellen Produktion fallen. Darüber hinaus umfasst die VPI zur Analyse und Bereitstellung der generierten Informationen moderne Techniken der Visualisierung und Anwenderinteraktion. Der Begriff Virtual Production Intelligence lehnt sich an verschiedene Konzepte an und vereint diese. Dabei verfolgt der Ansatz die ursprüngliche, von Luhn [Luh58] definierte Idee der Business Intelligence und überträgt diese auf den Bereich der Virtuellen Produktion. VPI bezieht sich dabei auf die integrierte Verarbeitung und Analyse von Informationen aus der Virtuellen Produktion. Diese wird verstanden als „die simulativ durchgeführte vernetzte Planung und Steuerung von Produktionsprozessen mit Hilfe digitaler Modelle. Sie dient der Optimierung von Produktionssystemen und erlaubt eine flexible Anpassung der Prozessgestaltung vor einer prototypischen Realisierung“[1].

Der grundlegende Informationsverarbeitungsprozess der VPI stützt sich auf den in Bild 1 skizzierten Informationsmanagementzyklus. Ziel dabei ist die Umsetzung eines Informationssystems in einer neuen oder bereits existierenden Domäne, das die Anforderungen derselben und der darin involvierten Akteure erfüllt. Ausgangspunkt ist die Identifikation der relevanten Akteure sowie die explizite Formulierung der offenen Bedürfnisse in der vorherrschenden Informationsinfrastruktur. Anschließend werden die zu integrierenden Informationsquellen identifiziert. Dabei werden sämtliche Daten durch Operationen wie Konsolidierung, Transformation, Bereinigung und Fusion in Informationsressourcen innerhalb des Systems und so in ein integriertes Modell überführt. Da-

[1] VDI Richtlinie 4499 – Blatt 1, Februar 2008: Digitale Fabrik – Grundlagen

rauf aufbauend werden domänenspezifische Analysemethoden eingesetzt, um die existierenden Informationen weiter anzureichern und schließlich mittels des Informationsproduktes dem Anwender zugänglich zu machen. Dieser kann verschiedene Analysen und eine anschließende Visualisierung der Ergebnisse anstoßen. Das Informationsprodukt der VPI ist die sogenannte VPI-Plattform, mit einer Webanwendung als Benutzeroberfläche, deren Zugang system- und geräteunabhängig ist. Diese Plattform dient Planungs- und Unterstützungsprozessen durch die Bereitstellung integrativer und explorativer Analysen für die Unterstützung von Fabrik- und Produktplanungsprozessen und hat zum Ziel, einen Beitrag zur Verringerung von Planungsaufwendungen sowie der Steigerung der Planungseffizienz bei Forschungs- und Entwicklungsaufgaben zu leisten.

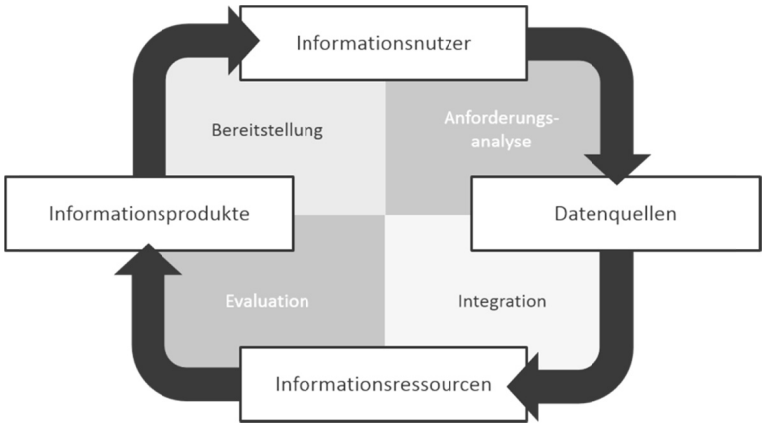


Bild 1: Informationsmanagementzyklus (nach [Krc11])

Zur Informationsmodellierung der betrachteten Domäne wird auf das Konzept der Ontologien aus dem Bereich der semantischen Technologien zurückgegriffen, um Entitäten sowie deren Beziehungen untereinander formal und explizit zu spezifizieren [Gru93]. Die Herausforderung hierbei erwächst aus der Komplexität der betrachteten Anwendungsdomänen und der Heterogenität der Datenstrukturen der Informationsquellen sowie aus der Implementierung semantischer Technologien in einem Informationssystem. Erste Vorarbeiten zur Lösung des dargestellten Problems und zur Einbindung von ontologie-basierten Konzepten liefern [Sch10] und [Mei12] mit einem Framework zur Kopplung numerischer Simulationen. Auf Basis dieser Arbeiten ist das Konzept der Virtual Production Intelligence (VPI) mit der sogenannten adaptiven Informationsintegration als zentralem Bestandteil und der VPI-Plattform als technische Implementierung entstanden [MMS+11], [RMB+12]. Die VPI überträgt grundlegende Konzepte von Intelligence-Ansätzen mittels Integration und Analyse von entlang der virtuellen Prozesse generierten Daten auf den Bereich der Virtuellen Produktion. Somit bietet die VPI die Möglichkeit eines integrativen Monitorings der Eigenschaften von Planungsprozessen sowie deren Entscheidungsunterstützung mittels eines Informationssystems, das sich aus einer konsistenten Informationsbasis und Kennzahlen als Grundlage für eine explorative und interaktive Informationsanalyse durch den Anwender zusammensetzt.

Der Beitrag der VPI im Gesamtkontext der Virtuellen Produktion fokussiert auf eine Realisierung der Digitalen Fabrik zum einen im Bereich der Integration unterschiedlicher Informationsquellen, zum anderen im Bereich der explorativen und interaktiven Analyse. Diese wird momentan über die Benutzerschnittstelle der Webanwendung (VPI-Plattform) realisiert. Die Implementierung einer Schnittstelle für die Interoperabilität mit Anwendungen aus der Virtual Reality wird derzeit im Rahmen der Entwicklung von *flapAssist* und *memoSlice* durchgeführt.

2.2 Fabriklayoutplanung mit *flapAssist*

Die Aufgabe von *flapAssist* ist die Unterstützung des Planers bei der Erarbeitung von Fabriklayouts mithilfe von Ansätzen der VR, der Informationsvisualisierung sowie der Datenannotation. Die Anwendung stellt dabei eine Analysemethode im Sinne der VPI dar, wobei sowohl Daten aus der VPI-Plattform abgerufen als auch neue, während der Nutzung generierte Informationen in diese zurückintegriert werden.

Zum einen werden bereits etablierte Funktionen bereitgestellt, wie etwa die realistische Begehung virtueller Fabrikmodelle innerhalb einer immersiven virtuellen Umgebung. Zum anderen dient *flapAssist* als Forschungsplattform für die Entwicklung und Erprobung neuer Konzepte zur weitergehenden Unterstützung des Planungsprozesses. Ein wichtiger Punkt bei der Entwicklung von *flapAssist* ist dabei immer die Integration in etablierte Arbeitsabläufe. Während beispielsweise die virtuelle Begehung von Fabrikmodellen spezielle VR-Hardware, wie etwa CAVEs oder Head-Mounted Displays (HMDs) voraussetzt, so muss gleichzeitig berücksichtigt werden, dass derartige Hardware im alltäglichen Planungsprozess entweder nicht zur Verfügung steht oder nicht durchgehend benötigt wird. Eine Entwicklungsrichtlinie ist daher die Einsetzbarkeit von *flapAssist* auf unterschiedlichen Hardware-Plattformen.



Bild 2: flapAssist in der aixCAVE mit davor aufgebautem Fabrikplanungstisch auf welchem visTABLE®touch läuft.

Virtuelle Begehungen werden durch die realitätsnahe Darstellung des virtuellen Fabrikmodells ermöglicht. Die darzustellenden Modelldaten bezieht *flapAssist* dabei vom

kommerziellen Fabrikplanungswerkzeug plavis visTABLE®touch[2] (siehe Bild 2). Hierzu wurde für visTABLE®touch eine Erweiterung entwickelt, welche Lade- und Modifikationskommandos über eine Netzwerkschnittstelle an *flapAssist* weiterleitet. Dort kann das Fabrikmodell folglich synchron zu den Planungsvorgängen in visTABLE®touch angezeigt werden. Im Gegensatz zu ähnlichen, existierenden Ansätzen [NPS+11] ist *flapAssist* explizit darauf ausgelegt, auf unterschiedlichen VR Plattformen eingesetzt werden zu können. Während CAVEs besonders zur maßstabsgetreuen virtuellen Begehung mit Gruppen geeignet sind, so lassen sich Desktop Systeme besser in den alltäglichen Planungsprozess integrieren. Um diese unterschiedlichen Plattformen auf technischer Ebene ansprechen zu können, kommt das VR Toolkit ViSTA zum Einsatz [AK08], welches es erlaubt durch angepasste Konfigurationsdateien, eine Vielzahl an VR Anzeigesystem, sowie Ein- und Ausgabegeräte einzubinden ohne Quellcode-Anpassungen vornehmen zu müssen. Daneben ist aber vor allem die Umsetzung eines Interaktionskonzepts notwendig, um *flapAssist* auf unterschiedlichen Plattformen sinnvoll einsetzbar zu machen. Zur Navigation durch das Modell stellt *flapAssist* für jede Plattform adäquate Techniken zur Verfügung. In immersiven Umgebungen kommen etwa Point-and-Fly Techniken in Kombination mit 6-Degree-of-Freedom Eingabegeräten zum Einsatz. Die Navigation auf nicht-immersiven Geräten erfolgt dagegen entweder über eine Maus- oder eine – für die 3D Navigation gut geeignete – Gamepad Steuerung. Für Selektions- und Steuerungsaufgaben werden Point-and-Click Metaphern in Kombination mit radialen Kontext Menüs, sogenannten Pie Menus, genutzt.

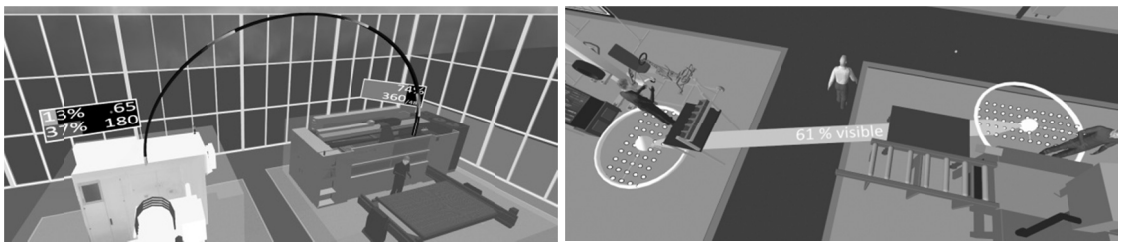


Bild 3: Links – Visualisierung von Materialflüssen. Rechts – Bestimmung der Einsehbarkeit von Arbeitsplätzen.

Durch die Visualisierung planungsrelevanter Kennzahlen wird der Planungsprozess weiter unterstützt. Zum einen erlaubt *flapAssist* es dem Nutzer Materialflusskosten anzuzeigen (siehe Bild 3, links). Zum anderen können auch Sichtbarkeitsinformationen (siehe Bild 3, rechts), zur Überprüfung der Sichtbarkeit zwischen beliebigen Arbeitsplätzen nachvollzogen werden.

Materialflusskosten beschreiben für ein beliebiges Paar von Maschinen oder Arbeitsplätzen, die Höhe des Aufwandes um Arbeitsstücke zwischen den betrachteten Endpunkten zu bewegen. In *flapAssist* wird diese Information in Form einer Adjazenzmatrix verarbeitet, welche aus der VPI-Plattform bezogen wird. Der Zugriff auf die Da-

[2] http://www.vistable.de/

ten geschieht dabei mittels SOAP Web-Services[3], welche eine standardisierte Schnittstelle für den Zugriff auf die Daten darstellen. Die Visualisierung erfolgt in Form von farbkodierten Bögen (Transportkosten zwischen einem Paar von Endpunkten) und Info-Karten (akkumulierte Transportkosten über einen Arbeitsplatz/eine Maschine). Die Farbe informiert den Planer dabei über die genauen Kosten, während eine Animation die Flussrichtung veranschaulicht.

Die Einsehbarkeit zwischen Arbeitsbereichen ist wichtig, um etwa die Kommunikation mittels Handzeichen sicherstellen zu können. Mit *flapAssist* kann der Benutzer sie überprüfen, indem er entsprechende Sichtbarkeitsinformationen semi-automatisch generiert. Dazu werden Arbeitsbereiche zunächst interaktiv durch geometrische Primitive approximiert und miteinander verbunden (siehe Bild 3). Anschließend wird automatisch eine Schätzung der Einsehbarkeit berechnet. Hierzu werden die approximierten Arbeitsbereiche zunächst mit Punkten abgetastet und so diskretisiert. Anschließend wird für jede auf diesen Punkten basierende „Sichtlinie“ algorithmisch detektiert, ob diese durch Objekte in der Fabrikhalle gestört wird. Aus dem Verhältnis zwischen der Zahl ungestörter Sichtlinien und der Gesamtzahl an Sichtlinien kann so die Einsehbarkeit zwischen den beiden Arbeitsbereichen abgeschätzt werden. Da diese Berechnungen mitunter sehr rechenintensiv sein können, werden sie asynchron ausgeführt. Die Visualisierung der approximierten Sichtbarkeit erfolgt durch farbkodierte Pfeile, welche mit dem numerischen Sichtbarkeitswert beschriftet sind (siehe Bild 3).

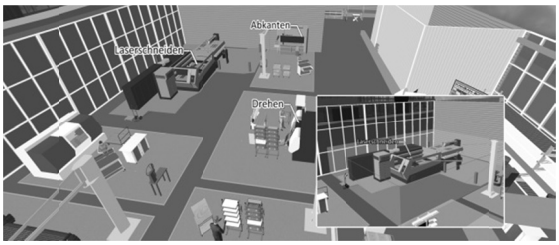


Bild 4: Erstellung von Annotationen zum Persistieren von Erkenntnissen und Kommentaren (hier: Beschriftungs- und Blickpunktannotationen).

Damit Erkenntnisse und Anmerkungen, welche während einer virtuellen Begehung gewonnen werden, nicht verloren gehen, erlaubt *flapAssist* die Erstellung von Annotationen. Der Benutzer kann dazu an beliebigen Stellen Beschriftungs- oder Blickpunkt-Annotationen erstellen und diese mit weiteren Metadaten, wie etwa Texten, Bildern, Zeichnungen oder Audiokommentaren versehen (siehe Bild 4). Damit die Erstellung von Annotationen vor allem innerhalb von CAVEs einfach möglich ist, wurde ein Annotations-Framework umgesetzt, welches geeignete Interaktionsmetaphern bereitstellt [PK15]. Dabei wurde eine Persistenzschicht realisiert, welche mit der VPI-Plattform kompatibel ist und so Annotationen zentral dort ablegen kann. Der Vorteil dabei ist, dass Benutzer beliebig zwischen verschiedenen VR Systemen oder auch Arbeitsplätzen

[3] http://www.w3.org/TR/soap/

wechseln können und dennoch Zugriff auf alle bereits erstellten Annotationen erhalten. Speziell wird hierdurch im Sinne der VPI der kollaborative Planungsprozess unterstützt.

2.3 Prozessanalyse mit *memoSlice*

Eine weitere Anwendungsdomäne die im Zuge der VPI betrachtet wird, ist die Maschinenkonfiguration. Hierbei anfallende Aufgaben werden üblicherweise von Domänenexperten für den Prozess der entsprechenden Maschinen durchgeführt. Heutzutage werden hierfür meist Technologietabellen herangezogen, die es ermöglichen unter einer Vielzahl möglicher Parameterkombinationen diejenige auszuwählen, die den zu erreichenden Qualitätskriterien am nächsten kommt. Da jedoch nur eine endliche Zahl möglicher Parameterkombinationen hinterlegt ist, ist es nicht möglich jede Kombination für einen bestimmten Prozess auszuwählen. So können Vorgaben an Kriterien in der Regel nur näherungsweise erreicht werden.



Bild 5: Analyse einer Maschinenkonfiguration mit dem memoSlice-*Widget innerhalb von* flapAssist *in der aixCAVE der RWTH-Aachen.*

Um jedoch Zugriff auf alle verfügbaren Kombinationen zu erhalten, muss eine kontinuierliche Abbildung von Parametern auf Kriterien vorliegen und ausgewertet werden. Dies kann mithilfe so genannter Metamodelle umgesetzt werden, welche eine Abstraktion über einem bekannten Modell eines Produktionsprozesses darstellen und Parameter mittels einer Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ auf Kriterien abbilden. Da solche multidimensionalen Funktionen allerdings inhärent schwer zu verstehen sind, sind Analysewerkzeuge nötig, die es dem Anwender erleichtern sich im multidimensionalen Parameterraum zurechtzufinden und so Parameterkombinationen auswählen zu können, die die geforderten Qualitätskriterien erfüllen. *memoSlice* stellt ein ebensolches Analysewerkzeug dar. Es erlaubt Anwendern mittels explorativer Visualisierung ein Verständnis für den Parameterraum von Prozessen zu entwickeln und ideale Parameterkombinationen zu bestimmen.

Es vereint mehrere Visualisierungstechniken, die für die Analyse von Metamodellen angepasst wurden und bietet Hilfestellungen für die Navigation im multidimensionalen Parameterraum. Da eine flüssige, interaktive Bedienung wesentlich für die Exploration des Parameterraums ist, wurde bei der Entwicklung großer Wert auf kurze Antwortzeiten und schnelle Berechnungen gelegt.

Wie in Bild 5 zu sehen ist, besteht die Anwendung aus drei Hauptkomponenten. Eine Scatterplot-Matrix [BC87] unterstützt den Anwender bei der Analyse von Korrelationen zwischen einzelnen Parametern und zeigt zugleich auf einen Blick, welcher Wertebereich an Kriterien für jeden einzelnen Parameter verfügbar ist. Die einzelnen Punkte in der Matrix stellen keine vorberechneten Werte dar, sondern können dadurch, dass ein Metamodell zugrunde liegt in Echtzeit neu berechnet werden. So ergibt sich auch die Möglichkeit für den Benutzer, Berechnungsvorschriften anzugeben, aufgrund derer auch in stark eingeschränkten Parametersubräumen Scatterplots mit hoher Dichte erstellt werden können.

Eine HyperSlice-Visualisierung [WL93] bildet die zweite Hauptkomponente von *memoSlice*. Sie stellt alle orthogonalen Schnitte von paarweisen Parameterkombinationen durch einen gemeinsamen Fokuspunkt dar. Zusätzlich werden auf der Diagonale Diagramme angezeigt, die alle eindimensionalen Schnitte durch den Parameterraum entlang jeweils einer Achse repräsentieren. Beides ermöglicht es dem Anwender, auf einen Blick die Auswirkungen von Änderungen der Parameter zu erfassen.

Die dritte Hauptkomponente bildet eine 3D-Visualisierung, die auf verschiedene Arten einen dreidimensionalen Schnitt durch den Fokuspunkt darstellt. Zum einen werden drei zueinander orthogonale 2D-Schnitte angezeigt, die präzise Parameteränderungen entlang der drei dargestellten Achsen repräsentieren. Zum anderen werden diese von einem 3D-Schnitt überlagert, der als direktes Volumenrendering dargestellt ist und einen weiträumigen Überblick über die Umgebung des Fokuspunkts gibt.

In der HyperSlice- und 3D-Visualisierung werden ergänzend noch weitere Navigationshilfen angezeigt. Eine Trajektorie, die mittels Particle-Tracings entlang des Gradientenfeldes des Metamodells berechnet wird, dient als Navigationshilfe. Durch sie kann der Anwender schnell zum nächstgelegenen lokalen Minimum oder Maximum des Parameterraums navigieren, was ihn weiter bei der Identifizierung idealer Prozesskonfigurationen unterstützt. Eine weitere Navigationshilfe, die besonders für Metamodellentwickler von Bedeutung ist, stellt die Anzeige von Trainingspunkten des Metamodells dar.

3 Integrationskonzept

Sowohl *flapAssist* als auch *memoSlice* sind auf spezifische Anwendungsfälle zugeschnittene Lösungen, die von unterschiedlichen Domänenexperten genutzt werden. Allerdings beeinflussen sich die betrachteten Problemstellungen teils gegenseitig. So beeinflusst beispielsweise die Konfiguration einzelner Maschinen teilweise die gesamte

Prozessplanung, was besonders im Kontext der KVP von Relevanz ist. Zwar können beide Anwendungen auch getrennt voneinander genutzt werden um solche Auswirkungen zu untersuchen, allerdings liegen so sämtliche relevanten Informationen getrennt voneinander vor. Zudem schließt die getrennte Nutzung auch einen teils ungünstigen Arbeitsablauf mit ein, da nicht nur eine Anwendung, sondern zwei Anwendungen separat voneinander bedient werden müssen. Insgesamt wird durch eine derartige Separierung die Synchronisierung der Informationen in beiden Anwendung erschwert.

Diesen Schwierigkeiten wirkt eine Kombination der beiden Einzelanwendungen in eine integrierte Gesamtlösung entgegen. Die größten Vorteile hiervon sind sicherlich, dass alle relevanten Informationen an einer Stelle vorliegen, aber auch, dass Experten aus verschiedenen Domänen in der Lage sind, dieselbe Anwendung zu bedienen, was weiterführend Diskussionen in einem kollaborativen Kontext positiv beeinflusst. Durch die Zusammenführung der Informationen aus zwei unterschiedlichen Domänen, kann das integrierte Planungswerkzeug somit zu einer zentralen Plattform für Diskussionen zwischen den unterschiedlichen Anwendungsbereichen werden. Darüber hinaus ergeben sich unmittelbar Synergien durch die kombinierten Funktionalitäten beider Einzelwerkzeuge. Die bei der beschriebenen kollaborativen Planung gewonnenen Erkenntnisse können unmittelbar über das in *flapAssist* zur Verfügung gestellte Annotationssystem festgehalten werden.

Vor dem Hintergrund dieser Überlegungen wurde *memoSlice* als ein Teilwerkzeug in *flapAssist* integriert. Dort kann es für zuvor definierte Maschinen direkt innerhalb von *flapAssist* aufgerufen werden. Bild 5 illustriert die Nutzung von *memoSlice* innerhalb von *flapAssist* bei einer virtuellen Fabrikbegehung in einer CAVE, wo es für eine Laserschneidemaschine aufgerufen wurde. Die beteiligten Experten können sofort im Kontext der gesamten Fabrik die Konfiguration der Maschine diskutieren und gegebenenfalls gewonnene Resultate sofort mit dem Annotationssystem festhalten. Der Vorteil dieser Lösung besteht nun darin, dass die so persistierten Ergebnisse nicht nur während des laufenden oder zukünftigen kollaborativen Planungstreffen zur Verfügung stehen (zeitlich und räumlich synchrone Kollaboration), sondern auch zu jedem beliebigen anderen Zeitpunkt von allen Experten eingesehen und ggf. weiter modifiziert werden können (zeitlich und räumlich asynchrone Kollaboration). Bei späteren Planungstreffen kann dann wieder gemeinsam auf die Informationen zugegriffen und mit diesen weitergearbeitet werden. Darüber hinaus bleibt die Nutzung von *memoSlice* und *flapAssist* als eigenständige und unabhängige Komponenten von deren Integration unberührt. Mithilfe der VPI-Plattform können Ergebnisse, welche mit einer Einzelkomponente erzielt wurden, wieder in die integrierte Gesamtlösung überführt werden. Insgesamt definiert die Kombination aus *memoSlice* und *flapAssist* somit eine zentrale Anlaufstelle für Experten verschiedener Anwendungsdomänen, ganz im Sinne des VPI-Grundgedanken.

Natürlich gilt es bei der Umsetzung des genannten Systems sicherzustellen, dass die integrierte Lösung weiterhin ein nutzerfreundliches Bedienkonzept zur Verfügung stellt

und auch die technischen Anforderungen der Einzelkomponenten berücksichtigt. Derartigen Fragestellungen widmen wir uns genauer in der Arbeit [GPH+15].

Insgesamt kann festgehalten werden, dass die Integration von *memoSlice* und *flapAssist* Vorteile für die Kommunikation und den Informationsaustausch zwischen unterschiedlichen Anwendungsbereichen mit sich bringt. Dabei stellt die vorgestellte Umsetzung jedoch nur einen ersten Schritt dar, welcher durch die Integration weiterer geeigneter Analysewerkzeuge weiter ausgebaut werden kann.

4 Resümee und Ausblick

In dieser Arbeit haben wir ein Konzept zur Integration zweier separater Werkzeuge zur Unterstützung von Planungsaufgaben im Fabrikkontext vor dem Hintergrund der Virtual Production Intelligence beschrieben. So bietet die integrierte Gesamtlösung Vorteile für die Kommunikation zwischen Experten verschiedener Domänen, wodurch der ganzheitliche Planungsprozess verbessert werden kann. In Zukunft planen wir die vorgestellte Arbeit zu evaluieren und weitergehende integrierte Visualisierungslösungen zu entwickeln.

Danksagungen

Die Autoren bedanken sich bei der Deutschen Forschungsgemeinschaft DFG für die freundliche Unterstützung der Forschungsarbeiten im Rahmen des Exzellenzclusters „Integrative Produktionstechnik für Hochlohnländer“. Die Autoren danken ferner der Demonstrationsfabrik Aachen für die Bereitstellung von Fabrikmodelldaten und der plavis GmbH für die Unterstützung bei der Entwicklung der visTABLE®touch-Integration.

Literatur

[AK08] ASSENMACHER, I.; KUHLEN, T.: The ViSTA Virtual Reality Toolkit. IEEE Conference on Virtual Reality, SEARIS Workshop, S. 23-28, 2008

[BC87] BECKER, R. A.; CLEVELAND, W. S.: Brushing Scatterplots. Technometrics, S. 127-142, 1987

[GPH+15] GEBHARDT, S.; PICK, S.; HENTSCHEL, B.; KUHLEN, T. W.: Softwaretechnische Integration von Virtual Reality Anwendungen am Beispiel des Fabrikplanungsprozesses. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, 2015

[Gru93] GRUBER, T. R.: A translation approach to portable ontology specifications. Knowledge Acquisition, Vol. 5, No. 2, S. 199-220

[Krc11] KRCMAR, H.: Einführung in das Informationsmanagement. Berlin: Springer, 2011

[Luh58] LUHN, H. P.: A business intelligence system. IBM Journal of Research and Development, Vol. 2, No. 4, S. 314-319, 1958

[Mei12] MEISEN, T.: Framework zur Kopplung numerischer Simulationen für die Fertigung von Stahlerzeugnissen. Dissertation, RWTH Aachen University

[MMS+11] MEISEN, T.; MEISEN, P.; SCHILBERG, D.; JESCHKE, S.: Adaptive Information Integration: Bridging the Semantic Gap between Numerical Simulations. Proceedings of the 13th International Conference on Enterprise Information Systems, Vol. 1, S. 51-65, 2011

[NPS+11] NEUGEBAUER, R.; PURZEL, F.; SCHREIBER, A.; RIEDEL, T.: Virtual reality-aided planning for energy-autonomous factories. Proceedings of the 9th IEEE International Conference on Industrial Informatics, S. 250-254, 2011

[PK15] PICK, S.; KUHLEN, T. W.: A Framework for Developing Flexible Virtual-Reality-centered Annotation System. IEEE Conference on Virtual Reality, SEARIS Workshop, 2015

[RMB+12] REINHARD, R.; MEISEN, T.; BÜSCHER, C.; SCHILBERG, D.; JESCHKE, S.: Virtual Production Intelligence – Ein Beitrag zur digitalen Fabrik. Verein Deutscher Ingenieure (Hg.): VDI-Berichte/VDI-Tagungsbände - Automation 2012. 13. Branchentreff der Mess- und Automatisierungstechnik. VDI Verlag (2171), 2012

[Sch10] SCHILBERG, D.: Architektur eines Datenintegrators zur durchgängigen Kopplung von verteilten numerischen Simulationen. Dissertation, RWTH Aachen University, 2010

[WL93] VAN WIJK, J. J.; VAN LIERE, R.: HyperSlice: Visualization of Scalar Functions of Many Variables. Proceedings of the IEEE Conference on Visualization, S. 119-125

Autoren

Dipl.-Inform. Sebastian Pick erhielt Ende 2009 sein Diplom in Informatik von der RWTH Aachen. Seit Anfang 2010 ist er Doktorand in der Virtual Reality Group des Lehrstuhls für Hochleistungsrechnen am IT Center der RWTH Aachen.

Dipl.-Medieninform. Sascha Gebhardt erhielt 2011 sein Diplom in Medieninformatik von der Ludwig-Maximilians-Universität München. Seitdem ist er Doktorand in der Virtual Reality Group des Lehrstuhls für Hochleistungsrechnen am IT Center der RWTH Aachen.

Dr. Bernd Hentschel ist leitender Mitarbeiter der Virtual Reality Group an der RWTH Aachen. Seine Forschungsinteressen umfassen die wissenschaftliche Visualisierung großer Daten, immersive Visualisierungsverfahren und die Entwicklung paralleler Visualisierungsalgorithmen. Er legt dabei besonderen Wert auf die enge, interdisziplinäre Zusammenarbeit mit Wissenschaftlern anderer Fachgebiete mit dem Ziel, anwendungsrelevante Visualisierungsverfahren zu entwickeln. Dr. Hentschel studierte Informatik an der RWTH Aachen (Dipl.-Inform., 2003) und promovierte dort in der Fakultät für Mathematik, Informatik und Naturwissenschaften (2009).

Prof. Dr. Torsten W. Kuhlen studierte an der RWTH Aachen Informatik und promovierte dort an der Fakultät für Elektrotechnik und Informationstechnik. Von 1998 bis 2014 leitete er die Virtual Reality Group am IT Center der RWTH Aachen. Als Wissenschaftler am Jülich Supercomputing Centre arbeitet er aktuell in interdisziplinären Kooperationen mit Ingenieuren und Neurowissenschaftlern an der Entwicklung von VR-Techniken für technisch-wissenschaftliche Anwendungen. Seit 2008 ist er Professor in der Fachgruppe Informatik der RWTH Aachen. Er ist Mitverfasser von über 200 begut-

achteten Beiträgen und Mitglied im Programm- und Organisationskomitee mehrerer internationaler Konferenzen zur Virtuellen Realität, Computergraphik und Visualisierung.

Dipl.-Math. Rudolf Reinhard arbeitet seit März 2010 als wissenschaftlicher Mitarbeiter des IMA/ZLW & IfU der RWTH Aachen in der Forschungsgruppe Produktionstechnik. Seit Oktober 2012 ist er als Projektleiter innerhalb des Exzellenzclusters "Exzellenzcluster „Integrative Produktionstechnik für Hochlohnländer" für die Research Area "Virtual Production Intelligence" verantwortlich.

Dipl.-Wirt.-Ing. Christian Büscher hat an der RWTH Aachen Wirtschaftsingenieurwesen mit Fachrichtung Maschinenbau und Studienrichtung Produktionstechnik studiert. Seit Juli 2010 arbeitet Herr Büscher als wissenschaftlicher Mitarbeiter am IMA/ZLW & IfU in der Forschungsgruppe Produktionstechnik, die er seit Oktober 2014 leitet. Seit November 2013 ist Herr Büscher zudem Geschäftsführer des Vereins Deutscher Ingenieure (VDI) Aachener Bezirksvereins.

M. Sc. Toufik Al Khawli absolvierte das Bachelorstudium "Mechatronics Engineering" an der Hariri Canadian University im Libanon. Er erhielt 2009 seinen Master von der RWTH Aachen. Seit 2009 ist er wissenschaftlicher Mitarbeiter des Lehr- und Forschungsgebietes Nichtlineare Dynamik der Laser-Fertigungsverfahren (NLD) an der RWTH Aachen, wo er momentan promoviert.

Dipl.-Phys. Urs Eppelt studierte Physik an der RWTH Aachen und schloss 2004 sein Studium am Lehrstuhl für Lasertechnik ab. Er erhielt den Schöneborn-Preis als Auszeichnung für sein Diplom. Seit 2004 ist er wissenschaftlicher Mitarbeiter am Lehrstuhl für Lasertechnik der RWTH-Aachen und wechselte 2006 zum Lehr- und Forschungsgebiet Nichtlineare Dynamik der Laser-Fertigungsverfahren, wo er momentan promoviert.

Hanno Voet, M.Sc. studierte Wirtschaftsingenieurwesen mit Fachrichtung Maschinenbau an der RWTH Aachen. Seit 2012 ist er wissenschaftlicher Mitarbeiter am Werkzeugmaschinenlabor WZL der RWTH Aachen in der Abteilung Fabrikplanung des Lehrstuhls für Produktionssystematik.

Dipl.-Wirt.-Ing. Julian Utsch, M.Sc. studierte Wirtschaftsingenieurwesen mit Fachrichtung Maschinenbau an der RWTH Aachen und Automotive Engineering an der Tsinghua University in Beijing. Seit Anfang 2014 ist er wissenschaftlicher Mitarbeiter am Werkzeugmaschinenlabor WZL der RWTH Aachen in der Abteilung Fabrikplanung des Lehrstuhls für Produktionssystematik.

Ein Ansatz zur Softwaretechnischen Integration von Virtual Reality Anwendungen am Beispiel des Fabrikplanungsprozesses

Dipl.-Medieninform. Sascha Gebhardt
Dipl.-Inform. Sebastian Pick
Dr. rer. nat. Bernd Hentschel
Prof. Dr. rer. nat. Torsten W. Kuhlen
IT Center, RWTH Aachen University
Seffenter Weg 23, 52074 Aachen
Tel. +49 (0) 241 / 80 29 737
E-Mail: gebhardt@vr.rwth-aachen.de

Dipl.-Math. Rudolf Reinhard
Dipl.-Wirt.-Ing. Christian Büscher
Lehrstuhl Informationsmanagement im Maschinenbau (IMA) der RWTH Aachen
IMA der RWTH Aachen, Dennewartstr. 27, 52068 Aachen
Tel. +49 (0) 241 / 80 91 131
E-Mail: rudolf.reinhard@ima-zlw-ifu.rwth-aachen.de

Zusammenfassung

Die Integration verschiedener eigenständiger Anwendungen stellt eine schwierige Aufgabe aus Sicht der Softwareentwicklung dar. Sie kann jedoch gerade im Bereich der durch virtuelle Realität (VR) gestützten Fabrikplanung deutliche Vorteile mit sich bringen, zum Beispiel um Wechselwirkungen zwischen verschiedenen Aufgabenbereichen zu kommunizieren. Um dies zu verdeutlichen wurden in dieser Arbeit zwei unabhängig voneinander entwickelte VR- und Visualisierungsanwendungen zu einer umfassenderen Planungslösung zusammengefasst. Bei der Integration wurden zusätzlich zu allgemeinen Anforderungen die Aspekte Parallelisierung und Interaktion besonders berücksichtigt. Zusammenfassend werden in dieser Arbeit anhand der Integration zweier Anwendungen aus dem Fabrikplanungsumfeld technische Lösungen präsentiert, mit denen mehrere VR-Anwendungen effektiv zu einer effizienten Gesamtlösung integriert werden können, wobei insbesondere die Aspekte Parallelität und Interaktion berücksichtigt werden. Die Effektivität des Ansatzes wird anhand von Leistungsmessungen belegt.

Schlüsselworte

Anwendungsintegration, Virtuelle Realität, Parallelisierung, Interaktion, Informationsvisualisierung

An Approach for the Softwaretechnical Integration of Virtual Reality Applications by the Example of the Factory Planning Process

Abstract

The integration of independent applications is a complex task from a software engineering perspective. Nonetheless, it entails significant benefits, especially in the context of Virtual Reality (VR) supported factory planning, e.g., to communicate interdependencies between different domains. To emphasize this aspect, we integrated two independent VR and visualization applications into a holistic planning solution. Special focus was put on parallelization and interaction aspects, while also considering more general requirements of such an integration process. In summary, we present technical solutions for the effective integration of several VR applications into a holistic solution with the integration of two applications from the context of factory planning with special focus on parallelism and interaction aspects. The effectiveness of the approach is demonstrated by performance measurements.

Keywords

Application Integration, Virtual Reality, Parallelization, Interaction, Information Visualization

1 Einleitung

Die Integration verschiedener eigenständiger Anwendungen stellt generell eine schwierige Aufgabe aus Sicht der Softwareentwicklung dar. Sie kann jedoch gerade im Bereich der Fabrikplanung deutliche Vorteile mit sich bringen, zum Beispiel um Wechselwirkungen zwischen verschiedenen Aufgabenbereichen der Planung zu kommunizieren. Da hierbei Virtual Reality (VR) immer mehr an Bedeutung gewinnt müssen bei der Integration entsprechender Anwendungen jedoch noch zusätzliche Anforderungen erfüllt werden, wie die Verwendung verteilter Systeme, niedrige Latenz und Bildwiederholrate oder konsistente Interaktionskonzepte. Deshalb präsentieren wir in dieser Arbeit eine Lösung, mit der unabhängige VR Anwendungen effektiv zu einer Gesamtlösung integriert werden können.

Um dies zu verdeutlichen wurden zwei unabhängig voneinander entwickelte VR und Visualisierungsanwendungen aus dem Bereich der Fabrikplanung und der Maschinenkonfiguration zu einer umfassenderen Planungslösung zusammengefasst. Hierbei handelt es sich zunächst um *flapAssist*, eine Anwendung, die das kommerzielle Fabrikplanungswerkzeug visTABLE®touch um die Möglichkeit immersiver virtueller Fabrikbegehungen und zusätzliche Visualisierungstechniken erweitert. Die zweite Anwendung, *memoSlice*, dient dazu, basierend auf multi-dimensionalen Metamodellen optimale Maschinenkonfigurationen zu ermitteln. Zudem wird *flapAssist* durch die Web-basierte Virtual Production Intelligence (VPI) Plattform unterstützt, welche Datenintegration und Aufbereitung bereitstellt.

Bei der Integration der Anwendungen wurden zusätzlich zu den bereits genannten Anforderungen auch die Aspekte Parallelisierung und gemeinsamer Datenaustausch mit der VPI umgesetzt. Um alle Anforderungen umzusetzen, wurde für beide Anwendungen ein gemeinsames Parallelisierungskonzept entwickelt, das eine feingranulare und faire Aufgabenverteilung garantiert. Zudem wurde ein Interaktionskonzept erarbeitet, das eine konsistente Bedienung beider Anwendungen erlaubt. Der Erfolg der Integration wird anhand von Messungen demonstriert, die verdeutlichen, dass die integrierte Lösung alle gestellten Anforderungen, besonders in Bezug auf Systemleistung, erfüllt.

Zusammenfassend werden in dieser Arbeit anhand der Integration von *flapAssist* und *memoSlice* bei gleichzeitiger Kopplung an die VPI-Plattform technische Lösungen präsentiert, mit denen mehrere VR-Anwendungen effektiv zu einer effizienten Gesamtlösung integriert werden können, wobei insbesondere die Aspekte Parallelität und Interaktion berücksichtigt werden.

2 Anwendungen

Bei den Werkzeugen, die zu einer Gesamtanwendung integriert wurden handelt es sich um *flapAssist*, das den Anwendungsbereich Fabrikplanung adressiert und *memoSlice*, das den Anwendungsbereich Maschinenkonfiguration adressiert. Besonders *flapAssist*

setzt zusätzlich auf die VPI-Plattform [RBM+12] als Informationssystem. Bevor auf die technischen Details der Integration der Anwendungen eingegangen wird, werden diese Werkzeuge kurz vorgestellt. Für eine detailliertere Beschreibung der Anwendungen verweisen wir auf [PGH+15].

Beim Entwurf von Fabriklayouts wird in der Regel auf Erfahrungswerte aus früheren Planungsvorhaben zurückgegriffen. Zusätzlich wird eine Reihe von Kennzahlen, wie etwa die Optimierung von Materialflüssen oder die Gruppierung von Maschinen ähnlichen Typs herangezogen, auf Basis derer der Planer das Layout optimiert. Dabei existieren Zusammenhänge zwischen den verschiedenen Kennzahlen, welche es für den Planer zu berücksichtigen gilt, um Konflikte während der Optimierung zu vermeiden. Beispielsweise kann eine zur Optimierung von Materialflüssen angestoßene Reorganisation von Maschinen dazu führen, dass die Sichtbarkeit zwischen Arbeitsplätzen beeinträchtigt wird.

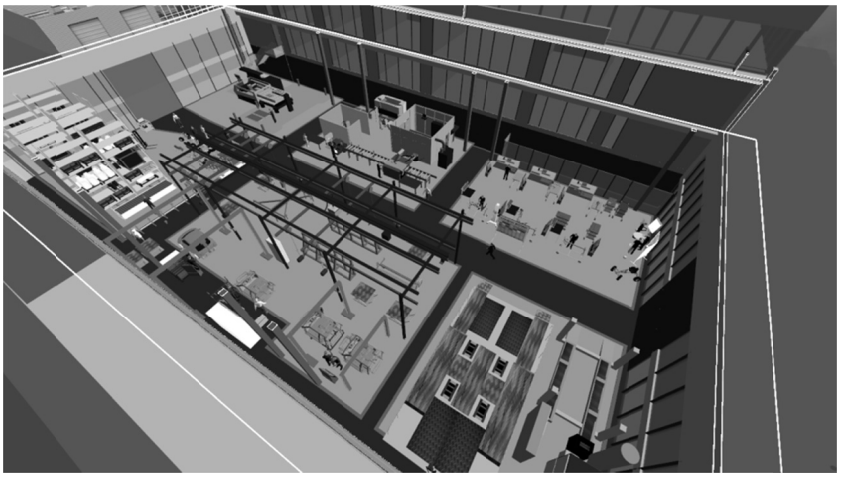


Bild 1: flapAssist, welches ein über visTABLE®touch bezogenes Fabrikmodell anzeigt.

Um diesen komplexen Planungsprozess zu unterstützen, werden dem Planer oft verschiedene Softwarelösungen an die Hand gegeben. Im Falle von *flapAssist* [PGK+14] ist es das Ziel, Techniken der virtuellen Realität (VR), der Visualisierung, sowie der Datenannotation in einer plattformübergreifenden Anwendung zu vereinen (s. Bild 1). VR erlaubt dabei die realitätsnahe virtuelle Begehung von Fabriken. Als optionale Datenquelle für Fabrikmodelle kann hierbei das kommerzielle Fabrikplanungswerkzeug visTABLE®touch genutzt werden. Zusätzlich stellt Visualisierung dem Planer weitere planungsrelevante Informationen, wie etwa die zuvor genannten Materialflüsse, innerhalb der virtuellen Umgebung zur Verfügung. Techniken der Datenannotation werden schließlich genutzt, um Ergebnisse des mittels *flapAssist* durchgeführten Analyseprozesses festzuhalten, wie beispielsweise Kommentare oder weiterführende Planungsaufgaben. Plattformübergreifend bedeutet in diesem Zusammenhang, dass nicht nur eine bestimmte Zielplattform unterstützt wird, wie etwa eine CAVE Automatic Virtual Environment (CAVE), sondern auch andere Plattformen berücksichtigt werden, wie beispielsweise Head-Mounted Displays (HMDs) oder Desktop-basierte Systeme.

Die Konfiguration von Maschinen wird in der Regel von Domänenexperten für den der Maschine zugrundeliegenden Prozess durchgeführt. Um die Experten dabei zu unterstützen existieren Technologietabellen, die eine Vielzahl von Parameterkombinationen auf zu erreichende Qualitätskriterien abbilden. Ein Nachteil hiervon ist, dass durch die begrenzte Anzahl hinterlegter Parameterkombinationen die Vorgaben an bestimmte Kriterien meist nur näherungsweise erreicht werden können.

Dieser Nachteil kann jedoch durch eine kontinuierliche Abbildung von Parametern auf Kriterien eliminiert werden. Eine solche Abbildung wird in dem hier besprochenen Beispiel über sogenannte Metamodelle realisiert. Diese stellen eine Abstraktion über dem physikalischen Modell eines Produktionsprozesses dar, indem sie die Prozessparameter mittels einer Funktion $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ auf Fertigungskriterien abbilden. Im vorliegenden Fall sind solche Funktionen technisch durch Netzwerke radialer Basisfunktionen (RBFs) realisiert. Allerdings sind derartige Funktionen inhärent schwer zu verstehen, weshalb Werkzeuge nötig sind, die es dem Anwender erleichtern, sich im multidimensionalen Parameterraum zurechtzufinden. *memoSlice* stellt ein ebensolches Analysewerkzeug dar, das es erlaubt, mittels explorativer Visualisierung ein Verständnis für den Parameterraum von Prozessen zu entwickeln und ideale Parameterkombinationen zu bestimmen. Bild 2 zeigt eine schematische Darstellung von *memoSlice*.

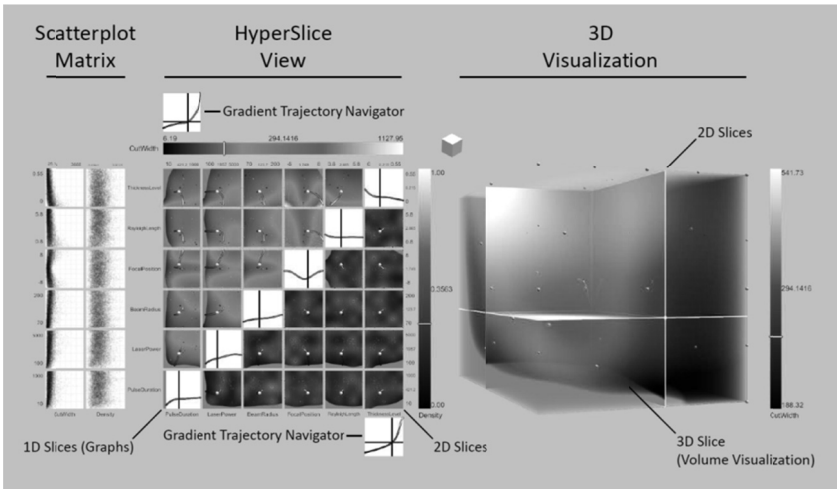


Bild 2: Schematische Darstellung von memoSlice *mit seinen Einzelkomponenten*

Die VPI beschreibt das grundlegende Konzept, auf dem das in dieser Arbeit verwendete Informationssystem für die Fabrikplanung basiert. Bei der VPI handelt es sich um ein ganzheitliches, integratives Konzept zur kollaborativen Planung von Kernprozessen der Technologie- (Material/Maschine), Produkt-, Fabrik- und Produktionsplanung und der Überwachung und Steuerung der Produktion und Produktentwicklung.

Das Konzept umfasst Methoden und Modelle zur Konsolidierung und Propagation von Daten, die entlang der betrachteten Domäne und des Anwendungsfalls generiert werden. Dabei werden insbesondere Planungsprozesse betrachtet, die in den Bereich der Virtuellen Produktion fallen. Darüber hinaus umfasst die VPI zur Analyse und Bereitstellung

der generierten Informationen moderne Techniken der Visualisierung und Anwenderinteraktion. Das Informationsprodukt der VPI ist die sogenannte VPI-Plattform, mit einer Webanwendung als Benutzeroberfläche, deren Zugang system- und geräteunabhängig ist. Diese Plattform dient Planungs- und Unterstützungsprozessen durch die Bereitstellung integrativer und explorativer Analysen für die Unterstützung von Fabrik- und Produktplanungsprozessen und hat zum Ziel, einen Beitrag zur Verringerung von Planungsaufwendungen sowie der Steigerung der Planungseffizienz bei Forschungs- und Entwicklungsaufgaben zu leisten. In der VPI Plattform vorhandene Daten und Funktionen werden zudem über Web Services bereitgestellt und können so von Anwendungen wie *flapAssist* genutzt werden.

3 Integration

Damit eine Integration der Anwendungen zu einem späteren Zeitpunkt generell möglich war, wurden beide Anwendungen im Hinblick darauf von vorn herein mit dem plattformunabhängigen VR-Toolkit ViSTA [AK08] entwickelt. Dieses stellt grundlegende Funktionalität, v.a. bezüglich der Abstraktion von Ein- und Ausgabegeräten, sowie zum Nutzen von Rechenclustern für den Betrieb größerer VR-Systeme bereit. Um die Anwendungen zu einer Gesamtlösung zu integrieren, mussten jedoch noch weitere Voraussetzungen geschaffen werden, um zu garantieren, dass Interaktivität und Bildwiederholrate durch die Integration so wenig wie möglich beeinflusst werden. Dies schließt die Umstellung, sowohl auf ein gemeinsames Parallelisierungskonzept, als auch ein gemeinsames Interaktionskonzept ein. Diese Aspekte werden im Folgenden genauer beleuchtet.

3.1 Parallelisierung

Sowohl in *flapAssist*, als auch in *memoSlice* müssen teils viele Berechnungen nebenläufig getätigt werden, da sie zu langwierig für eine Berechnung in den Hauptschleifen der Anwendungen sind. Da die Anzahl der gleichzeitig abzuarbeitenden Aufgaben in beiden Anwendungen stark von der Nutzerinteraktion abhängt, ist eine naive Lösung, in der für jede Aufgabe ein einzelner Thread gestartet wird nicht zielführend, denn dies könnte dazu führen, dass nebenläufig abzuarbeitende Threads auf dem Kern der Hauptschleife ausgeführt werden, was die Antwortzeiten und Bildwiederholraten der Anwendungen deutlich beeinträchtigen könnte.

So wurden beide Anwendungen aufgrund der Komplexität der nebenläufigen Aufgaben, aber auch mit Ausblick auf die Integration, auf ein Task-basiertes Parallelisierungskonzept umgestellt. Hierfür bot sich Intel® Threading Building Blocks (TBB)[1] an, da es mit

[1] www.threadingbuildingblocks.org

moderatem Aufwand erlaubt performante, hierarchische, Task-basierte Parallelisierung zu realisieren.

Nachdem die Parallelisierung sowohl in *flapAssist*, als auch in *memoSlice* auf Intel® TBB Tasks basiert, können beide Anwendungen problemlos und ohne große Leistungseinbußen im selben Prozess betrieben werden. Um jedoch zu jeder Zeit eine maximale Prozessorauslastung bei fairer Ressourcenverteilung zu gewährleisten, wurden die Tasks in beiden Anwendungen hierarchisch so verfeinert, dass einzelne Aufgaben bei geringer Prozessorauslastung leicht auf alle verfügbaren Kerne verteilt werden können, bei hoher Auslastung allerdings die Tasks aus beiden Anwendungen möglichst gleichmäßig auf die Kerne verteilt bleiben. So erstellen beide Anwendungen bei Bedarf Tasks, die an den Intel® TBB Task Scheduler übergeben werden. Bei der Ausführung werden die Berechnungen jeweils in weitere Subtasks aufgeteilt um dem Intel® TBB Task Scheduler die Möglichkeit einzuräumen, die Taskhierarchie je nach aktueller Ressourcenverfügbarkeit mit Tiefen- oder Breitenpriorität abzuarbeiten. Dies ist in Bild 3 illustriert.

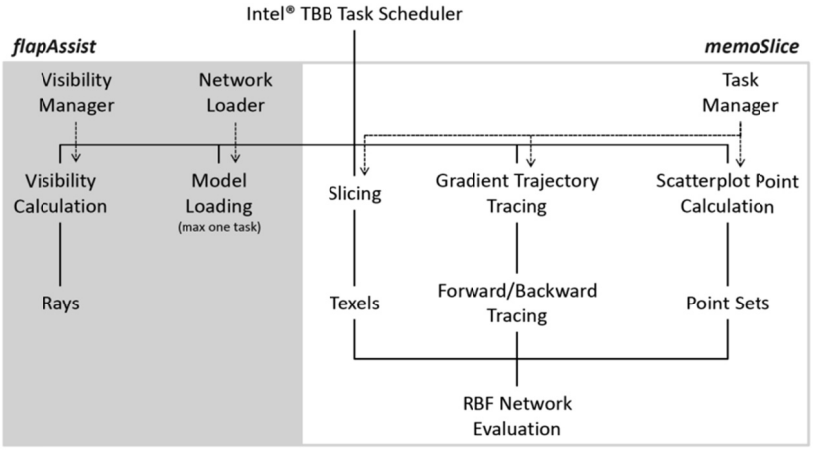


Bild 3: Die verschiedenen Tasks der integrierten Lösung mit ihren Verfeinerungsstufen.

In *flapAssist* müssen Sichtbarkeitsberechnungen und Ladevorgänge von Geometrieobjekten nebenläufig abgearbeitet werden, da diese mitunter einige Sekunden bis zu ihrer Beendigung benötigen. Für die Sichtbarkeitsberechnungen wird für jeweils ein Paar von Arbeitsbereichen ein Intel® TBB Task generiert in welchem alle Linientests durchgeführt werden. Innerhalb dieser Tasks werden wiederum Subtasks für die Berechnung von Strahlenbündeln erzeugt.

Zum Laden von Geometrieobjekten wird ebenfalls für jeden Ladevorgang, d.h. für jedes zu ladende Modell, ein Intel® TBB Task generiert. Lade-Tasks unterscheiden sich von den anderen Tasks dahingehend, dass sichergestellt wird, dass zu jedem Zeitpunkt nur maximal ein Ladevorgang durchgeführt wird. Diese Einschränkung wurde getroffen, um zu vermeiden, dass Dateisystemzugriffe durch mehrere gleichzeitige Lade-Tasks

zum Flaschenhals werden und sich dadurch ihre Laufzeit verlängert. Aus diesem Grund wurde auch darauf verzichtet diese Tasks weiter zu verfeinern.

In *memoSlice* werden alle rechenintensiven Visualisierungsaufgaben nebenläufig abgearbeitet, um Latenz in der Nutzerschnittstelle zu vermeiden. Um Bildraten nicht negativ zu beeinflussen, wurde darauf verzichtet, zeitintensive Berechnungen auf der Grafikkarte durchzuführen. Stattdessen werden sämtliche Berechnungen als Intel® TBB Tasks auf der CPU durchgeführt. Es existieren drei verschiedene Arten von Tasks in *memoSlice*: Berechnung von Schnitten durch den Parameterraum, Berechnung von Gradienten-Trajektorien und Berechnung von Punkten für die Scatterplot Matrix. Die Berechnungen von Schnitten und Punkten erfolgen iterativ. Es werden also zunächst niedrige Auflösungen, bzw. Punktzahlen berechnet um ein schnelles Feedback für den Anwender zu ermöglichen. Sobald niedrige Auflösungen verfügbar sind, werden die Darstellungen weiter verfeinert. Da jedoch durch Intel® TBB die Abarbeitungsreihenfolge einzelner Tasks nicht garantiert wird, nutzt *memoSlice* einen eigenen Task Manager, der Intel® TBB Tasks so erstellt, dass eine gleichmäßige Verfeinerung der einzelnen Visualisierungskomponenten gewährleistet wird.

Alle Task-Kategorien in *memoSlice* werden mit jeweils zwei Stufen weiter verfeinert. So werden Tasks für das Erstellen von Schnitten zunächst in Subtasks auf Texel-Ebene für die Schnitttexturen aufgeteilt. Tasks für die Berechnung von Gradienten-Trajektorien werden in Vorwärts- und Rückwärtsberechnung geteilt und die Berechnung von Punkten wird in Punkte-Submengen aufgeteilt. Die Auswertung des zugrunde liegenden RBF-Netzwerks erfolgt in *memoSlice* immer Task-parallel. Da alle zuvor erwähnten Subtasks eine mehrfache Auswertung des RBF-Netzwerks beinhalten erfolgt hier inhärent die zweite Stufe der Task-Verfeinerung.

3.2 Interaktion

Um eine alltagstaugliche Integration beider Anwendungen zu ermöglichen, ist auch ein durchgängiges Interaktionskonzept nötig. In Vorbereitung auf die Integration wurden deshalb beide Anwendungen auf ein neues Widgetsystem umgestellt, das derzeit als Erweiterung für ViSTA entwickelt wird. Dieses ermöglicht beispielsweise die kombinierte Verwendung verschiedener Fokusstrategien und stellt Abstraktionsmöglichkeiten für verschiedene Eingabegeräte, sowie eine Auswahl an Basiswidgets bereit. Für Aufgaben aus dem Bereich der Systemkontrolle wurden in beiden Anwendungen Extended Pie Menus [GPL+13] verwendet, die mittlerweile auch als Erweiterung des neuen Widgetsystems für ViSTA vorliegen.

Nachdem *flapAssist* durch die Integration des Widgetsystems in der Lage ist, beliebige ViSTA Widgets zu verwenden, wurde *memoSlice* in ein solches Widget umgewandelt, welches frei im Raum positioniert werden kann. So ist es möglich für jede zu konfigurierende Maschine der virtuellen Fabrikhalle eine Instanz von *memoSlice* zu erstellen und bei Bedarf vor der Maschine anzuzeigen. Durch die gemeinsame Verwendung des

Widgetsystems existiert automatisch ein durchgängiges Bedienkonzept, das es erlaubt, im freien Wechsel mit *memoSlice* oder beliebigen anderen Bedienelementen in *flapAssist* zu interagieren.

3.3 Kopplung mit der VPI Plattform

Bereits vor der Integration von *flapAssist* und *memoSlice* konnte *flapAssist* die VPI-Plattform zum einen als Datenquelle und zum anderen zum Persistieren von gewonnenen Erkenntnissen nutzen. Hierfür bietet die VPI-Plattform WebServices über eine SOAP-Schnittstelle an. Durch die Integration ist es nun möglich, die bereits bestehenden Kommunikationskanäle zwischen VPI-Plattform und *flapAssist* auch zum Persistieren von mittels *memoSlice* in der integrierten Lösung gewonnenen Erkenntnissen zu nutzen.

4 Ergebnisse

Um die Effektivität der im letzten Kapitel aufgezeigten Lösung zu untersuchen, wurde der Einsatz des Gesamtsystems in der aixCAVE der RWTH Aachen analysiert. Die Ergebnisse dieser Analyse werden im vorliegenden Kapitel besprochen.

Für die Leistungsanalyse wurde die Leistung der Einzelkomponenten *flapAssist* und *memoSlice* mit der des Gesamtsystems verglichen. Da die Leistung der einzelnen Komponenten von der Zahl der vorhandenen CPU-Kerne abhängt, wurden die Messungen für verschiedene Systemkonfigurationen durchgeführt. Als Testsystem diente dabei die aixCAVE der RWTH Aachen, welche von einem Hochleistungs-Rechencluster angetrieben wird. Da für eine interaktive Anwendung die Ansprechbarkeit des Systems von hoher Bedeutung ist, wurde als Metrik die Bildwiederholrate der Anwendung zugrunde gelegt.

Bei der aixCAVE handelt es sich um eine 5-seitige CAVE bei der insgesamt 24 Aktiv-Stereo Projektoren zum Einsatz kommen. Jeder Projektor wird von einem dedizierten Clusterknoten mit Bilddaten versorgt, wobei jedem Knoten zwei 6-Kern Intel® Xeon X5650 CPUs, zwei NVIDIA Quadro 6000 Grafikkarten und insgesamt 24 GB Arbeitsspeicher zur Verfügung stehen. Die Bilder für das linke und das rechte Auge werden auf jedem Clusterknoten mit zwei entsprechend konfigurierten Anwendungsinstanzen erzeugt, so dass diese sich die Rechenressourcen teilen. Die Leistungsmessungen wurden für *flapAssist* und *memoSlice* separat sowie für das integrierte Gesamtsystem durchgeführt. Alle Messungen wurden sowohl für den Einzelplatz- als auch für den Clusterbetrieb durchgeführt. Für die Messungen im Einzelplatzbetrieb wurde jeweils nur ein Knoten des Rechenclusters verwendet, wohingegen Messungen im Clusterbetrieb mit allen Knoten erfolgten. Dabei wurde den Applikationen in jedem Messdurchgang eine bestimmte Anzahl an CPU-Kernen für das Abarbeiten nebenläufiger Aufgaben zugeteilt. Für *flapAssist* und das Gesamtsystem wurden ferner zwei verschiedene Fabriksze-

narien unterschiedlicher Komplexität miteinander verglichen. Alle Messungen wurden jeweils unter Volllast und ohne zusätzliche Last durch nebenläufige Aufgaben durchgeführt. Tabelle 1 zeigt die Ergebnisse dieser Messungen. Gemessen wurde die Systemleistung jeweils im Einzelplatz-, bzw. im Clusterbetrieb sowie bei Verwendung verschiedener Zahlen von Kernen für nebenläufige Aufgaben unter Volllast und ohne Last. Für flapAssist und das Gesamtsystem wurden zwei verschiedene Fabrikmodelle verwendet, ein geometrisch einfaches (Fabrik #1) sowie ein komplexes (Fabrik #2).

Tabelle 1: Analyseergebnisse für die Teilsysteme sowie das integrierte Gesamtsystem in Bildern pro Sekunde inkl. Standardabweichung (σ).

			flapAssist Fabrik #1	*flapAssist* Fabrik #2	*memoSlice*	Gesamtsys. Fabrik #1	Gesamtsys. Fabrik #2
Einzelplatzbetrieb	1 Kern	ohne Last	122,7 (σ=1,5)	31,8 (σ=1,0)	61,6 (σ=1,4)	46,9 (σ=0,64)	20,7 (σ=1,1)
		Voll-last	120,8 (σ=3,5)	31,1 (σ=1,2)	66,4 (σ=10,8)	50,4 (σ=2,0)	21,5 (σ=1,0)
	5 Kerne	ohne Last	122,8 (σ=0,8)	30,9 (σ=0,4)	61,2 (σ=0,3)	46,2 (σ=0,6)	21,0 (σ=0,4)
		Voll-last	121,5 (σ=3,2)	32,4 (σ=1,1)	63,8 (σ=10,3)	50,0 (σ=2,2)	21,9 (σ=0,8)
Clusterbetrieb	1 Kern	ohne Last	34,8 (σ=1,4)	14,4 (σ=1,2)	29,0 (σ=2,2)	17,2 (σ=0,7)	11,2 (σ=1,0)
		Voll-last	35,7 (σ=2,0)	14,3 (σ=1,3)	34,0 (σ=2,8)	17,0 (σ=0,8)	11,4 (σ=0,6)
	5 Kerne	ohne Last	34,9 (σ=1,9)	14,6 (σ=1,0)	29,7 (σ=2,6)	16,9 (σ=1,1)	10,9 (σ=0,8)
		Voll-last	32,3 (σ=3,0)	13,1 (σ=1,0)	29,9 (σ=3,6)	16,3 (σ=1,4)	10,5 (σ=0,6)

Die Ergebnisse zeigen, dass die Bildwiederholraten für *flapAssist* und das Gesamtsystem wie erwartet stark von der Komplexität des angezeigten Fabrikmodells abhängen. Gleichzeitig wird aber auch ersichtlich, dass die Bildwiederholrate in allen Fällen nur marginal abfällt, sobald die Kerne für nebenläufige Aufgaben ausgelastet sind. Für *memoSlice* und das Gesamtsystem ist teils sogar ein leichter Anstieg der Bildwiederholrate zu beobachten. Dieser lässt sich dadurch erklären, dass *memoSlice* während Berechnungen reduzierte Darstellungen ausgibt, die die Bildwiederholrate positiv beeinflussen. Insgesamt kann anhand der Ergebnisse empirisch die Effektivität des vorgestellten In-

tegrationskonzepts bestätigt werden, da im Rahmen der Messgenauigkeit die Bildwiederholraten des Gesamtsystems den kombinierten Bildwiederholraten der Einzelsysteme entspricht.

5 Resümee und Ausblick

In dieser Arbeit wurden die Anwendungen flapAssist und memoSlice vorgestellt, die unterstützende Werkzeuge auf verschiedenen Ebenen des Fabrikplanungsprozesses darstellen. Als Kernpunkt wurde dann die Integration der beiden Anwendungen zu einer umfassenderen Fabrikplanungsanwendung beschrieben, bevor im Ergebniskapitel gezeigt werden konnte, dass die Integration ohne größere Leistungseinbußen durchgeführt wurde.

Als nächste Schritte auf dem Weg zu einer umfassenden Fabrikplanungslösung werden künftig weitere Visualisierungen von Kennzahlen umgesetzt, die sich derzeit noch in Planung befinden. Ferner wird zum Zweck eines besseren Informationsaustausches die Anbindung an die VPI-Plattform ausgeweitet.

Danksagungen

Die Autoren bedanken sich bei der Deutschen Forschungsgemeinschaft DFG für die freundliche Unterstützung der Forschungsarbeiten im Rahmen des Exzellenzclusters „Integrative Produktionstechnik für Hochlohnländer. Die Autoren danken ferner der Demonstrationsfabrik Aachen für die Bereitstellung von Fabrikmodelldaten und der plavis GmbH für die Unterstützung bei der Entwicklung der visTABLE®touch-Integration.

Literatur

[AK08] ASSENMACHER, I.; KUHLEN, T. (2008): The ViSTA Virtual Reality Toolkit. Proceedings of the IEEE VR Workshop Software Engineering, S. 23-28.

[GPL+13] GEBHARDT, S., PICK, S., LEITHOLD, F., HENTSCHEL, B., & KUHLEN, T. (2013): Extended Pie Menus for Immersive Virtual Environments. Visualization and Computer Graphics, IEEE Transactions on (Vol. 19, No. 4), S. 644-651.

[PGH+15] PICK, S., GEBHARDT, S., HENTSCHEL, B., KUHLEN, T. W., REINHARD, R., BÜSCHER, C., ET AL. (2015). Ein Konzept zur Verdeutlichung der Wechselwirkungen zwischen verschiedenen Aufgabenbereichen des Virtual Reality gestützten Fabrikplanungsprozesses. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung.

[PGS+14] PICK, S., GEBHARDT, S., KREISKÖTHER, K., REINHARD, R., VOET, H., BÜSCHER, C., ET AL. (2014). Advanced Virtual Reality and Visualization Support. Entwerfen Entwickeln Erleben, S. 187-198.

[RBM+12] REINHARD, R., BÜSCHER, C., MEISEN, T., SCHILBERG, D., & JESCHKE, S. (2012): Virtual Production Intelligence – A Contribution to the Digital Factory. In C.-Y. Sum, S. Rakheja, & H. Liu, Lecture Notes in Computer Science; Intelligent Robotics and Applications, S. 706-715.

Autoren

Dipl.-Medieninform. Sascha Gebhardt erhielt 2011 sein Diplom in Medieninformatik von der Ludwig-Maximilians-Universität München. Seitdem ist er Doktorand in der Virtual Reality Group des Lehrstuhls für Hochleistungsrechnen am IT Center der RWTH Aachen.

Dipl.-Inform. Sebastian Pick erhielt Ende 2009 sein Diplom in Informatik von der RWTH Aachen. Seit Anfang 2010 ist er Doktorand in der Virtual Reality Group des Lehrstuhls für Hochleistungsrechnen am IT Center der RWTH Aachen.

Dr. Bernd Hentschel ist leitender Mitarbeiter der Virtual Reality Group an der RWTH Aachen. Seine Forschungsinteressen umfassen die wissenschaftliche Visualisierung großer Daten, immersive Visualisierungsverfahren und die Entwicklung paralleler Visualisierungsalgorithmen. Er legt dabei besonderen Wert auf die enge, interdisziplinäre Zusammenarbeit mit Wissenschaftlern anderer Fachgebiete mit dem Ziel, anwendungsrelevante Visualisierungsverfahren zu entwickeln. Dr. Hentschel studierte Informatik an der RWTH Aachen (Dipl.-Inform., 2003) und promovierte dort in der Fakultät für Mathematik, Informatik und Naturwissenschaften (2009).

Prof. Dr. Torsten W. Kuhlen studierte an der RWTH Aachen Informatik und promovierte dort an der Fakultät für Elektrotechnik und Informationstechnik. Von 1998 bis 2014 leitete er die Virtual Reality Group am IT Center der RWTH Aachen. Als Wissenschaftler am Jülich Supercomputing Centre arbeitet er aktuell in interdisziplinären Kooperationen mit Ingenieuren und Neurowissenschaftlern an der Entwicklung von VR-Techniken für technisch-wissenschaftliche Anwendungen. Seit 2008 ist er Professor in der Fachgruppe Informatik der RWTH Aachen. Er ist Mitverfasser von über 200 begutachteten Beiträgen und Mitglied im Programm- und Organisationskomitee mehrerer

internationaler Konferenzen zur Virtuellen Realität, Computergraphik und Visualisierung.

Dipl.-Math. Rudolf Reinhard arbeitet seit März 2010 als wissenschaftlicher Mitarbeiter des IMA/ZLW & IfU der RWTH Aachen University in der Forschungsgruppe Produktionstechnik. Seit Oktober 2012 ist er als Projektleiter innerhalb des Exzellenzclusters "Exzellenzcluster „Integrative Produktionstechnik für Hochlohnländer" für die Research Area "Virtual Production Intelligence" verantwortlich.

Dipl.-Wirt.-Ing. Christian Büscher hat an der RWTH Aachen Wirtschaftsingenieurwesen mit Fachrichtung Maschinenbau und Studienrichtung Produktionstechnik studiert. Seit Juli 2010 arbeitet Herr Büscher als wissenschaftlicher Mitarbeiter am IMA/ZLW & IfU in der Forschungsgruppe Produktionstechnik, die er seit Oktober 2014 leitet. Seit November 2013 ist Herr Büscher zudem Geschäftsführer des Vereins Deutscher Ingenieure (VDI) Aachener Bezirksvereins.

Web-basierte Integrationsplattform für dynamische 3D-Anlagenvisualisierung

Reinhard Langmann, Christian Geiger
Fachhochschule Düsseldorf
Josef-Gockeln-Str. 9, 40474 Düsseldorf
Tel. +49 (0) 211 / 43 51 308, Fax. +49 (0) 211 / 43 51 303
E-Mail: langmann@ccad.eu

Zusammenfassung

Der Beitrag stellt eine Integrations- und verteilte Steuerungsplattform für die 3D-Anlagenvisualisierung vor, die ausschließlich auf Webtechnologien basiert. Die Plattform ermöglicht es per Webbrowser eine 3D-Anlagenvisualisierung zu projektieren, die einzelnen 3D-Modelle mit realen Prozessdaten aus einer automatisierten Anlage zu verbinden und in einem RUN-Mode die so erstellt Anlagenvisualisierung auch auszuführen. Zur flexiblen Nutzung der Plattform ist diese mehrnutzerfähig. Nutzer können sich mit unterschiedlichen Rollen (Gast, User, Admin) in die Plattform einloggen.

Schlüsselworte

3D-Anlagenvisualisierung, Integrationsplattform, CPS-basierte Automation, Industrie 4.0

Web-based integration platform for dynamic 3D plant visualization

Abstract

The contribution presents a distributed integration and control platform for 3D plant visualization based completely on Web technology. The platform allows the projecting of a 3D plant visualization by a Web browser, the connection of separate 3D objects with real process data from an automated station and the executing of the projected plant visualization in a RUN mode in the Web browser. The platform is multi-client capable for flexible using. Users can log in into the platform by different roles (guest, user, admin).

Keywords

3D plant visualization, integration platform, CPS-based automation, Industry 4.0

1 Motivation

Die Forderung nach immer kürzeren Innovationszyklen, die höhere Individualisierung von Produkten und die damit einhergehende Flexibilisierung von Produktionsabläufen sind mit klassischen Produktionssystemen meist nicht mehr realisierbar [KAG14]. Aus diesen Forderungen ergeben sich eine Umstrukturierung von Produktionsprozessen und eine Neugestaltung vorhandener Infrastrukturen. Grundlegend bei diesem Transformationsprozess, welcher aktuell auch mit dem Begriff „Industrie 4.0“ beschrieben wird, ist der Einsatz von Cyber Physical Systems (CPS). CPS verstehen sich als ein Verbund von softwaretechnischen Komponenten mit mechanischen und elektronischen Teilen, welche über eine Dateninfrastruktur, wie das Internet, miteinander verbunden sind [GMA14]. CPS im Kontext der produzierenden Industrie, werden häufig auch als Cyber Physical Production Systems (CPPS) bezeichnet.

Ein grundlegendes Ziel von CPPS ist die Konstruktion neuartiger Industrieproduktionsanlagen, welche sich hochflexibel an die jeweiligen Produktionserfordernisse anpassen können. Um dies zu realisieren wandelt sich die klassische Automatisierungshierarchie in ein verteiltes Netz von Funktionen und Diensten, auch als „Automatisierungswolke“ oder „CPS-basierte Automation“ bezeichnet (Bild 1).

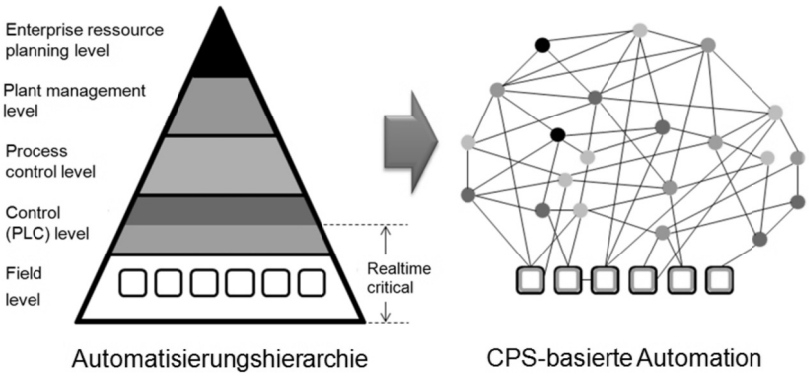


Bild 1: CPS-basierte Automation nach [GMA14]

Neben den Vorteilen für das eigentliche Produktionssystem liefert diese Strategie auch die Basis für eine neue Form von Überwachungs- und Steuerungssystemen. Diese werden bei einer direkten Interaktion mit einem Gerat üblicherweise als Human Machine Interface (HMI) und bei einer geräteübergreifenden Interaktion als Supervisory Control and Data Acquisition (SCADA) System bezeichnet. Durch den Einsatz von CPPS lassen sich SCADA Systeme dezentraler und somit verteilt und kollaborativ gestalten.

Die Verwendung von Webtechnologien und die damit verbundene Auslagerung des SCADA Systems auf Webbrowser und -server können völlig neue Produktions- und Überwachungsmöglichkeiten zulassen, speziell in der Darstellung von Prozessdaten. So lässt sich auch der aufkommende Standard *Web Graphics Library* (WebGL), welcher zur Darstellung von 2D- und 3D-Objekten im Webbrowser mit Unterstützung der Gra-

fikkarte genutzt wird, in das System integrieren. Mit Hilfe von WebGL und einem webbasierten SCADA-System, können Prozessdaten nicht nur klassisch in Form von 2D-Bedienelementen, sondern auch als dreidimensionale virtuelle Repräsentation der Produktionsanlage unter Berücksichtigung der Echtzeitdaten dargestellt werden. Durch eine 3D-Darstellung lassen sich Prozessdaten übersichtlicher und zugleich realitätsnah präsentieren. Der Ansatz dieser Darstellung ist nicht neu, jedoch wurde er bisher nur proprietär, betriebssystemabhängig und in zentralisierten SCADA-Systemen teilweise umgesetzt. Beispiele dazu sind die SCADA-Systeme Movicon NEXT [MOV14] und Iconics Genesis64 [GEN14].

2 Stand der Technik und Aufgabenstellung

Die Recherchen im Stand der Technik haben gezeigt, dass es zwar zunehmend mehr SCADA/Visualisierungssysteme gibt, die sich auf Webtechnologien konzentrieren, jedoch gibt es bisher kaum Systeme, welche ohne Browser PlugIn‘s auskommen. Die Möglichkeit, ein kundenspezifisches SCADA-System vollständig über den Browser zu konfigurieren, editieren und zu bearbeiten, also eine Projektierungsumgebung im Browser (und Internet), bietet keines der Systeme. Weiter hat die Recherche gezeigt, dass es zwar einige wenige SCADA/HMI-Systeme gibt, die auch eine Verwendung von 3D-Darstellungen zulassen, jedoch nutzt keines dieser Systeme aktuelle standardisierte Webtechnologien zur Darstellung und Kommunikation und die Visualisierung von Echtzeitdaten ist in der Regel nicht möglich.

Auch in der Gebäudeautomatisierung, in der es zwar deutlich mehr Systeme gibt die auf reinen Webtechnologien aufbauen und die sich teilweise auch dezentralisieren lassen, fehlt es jedoch an standardisierten Schnittstellen, speziell für den professionellen Einsatz. Die erzeugbaren Benutzeroberflächen sind in der Regel nur sehr einfach und somit an die Bedienung eines privaten Anwenders angepasst. 3D-Darstellungen werden nicht verwendet und damit ist auch die Nutzung für große Gebäudekomplexe stark eingeschränkt.

Zusammengefasst ergibt sich deshalb aus den Nachteilen des Standes der Technik der folgende Forschungsbedarf:

- Es fehlt eine durchgängige und konsistente Modellierung eines SCADA-Systems auf Basis von webbasierten Standardtechnologien.
- Webtechnologien werden zwar eingesetzt, die flexible Verteilung von Struktur und Funktion eines Automatisierungssystems wird aber nicht genutzt.
- Neue Web-Technologien wie Websockets und WebGL werden für SCADA-Systeme bisher kaum eingesetzt.

Bisherige Produkte nutzten das Web nur, um Systeme mit einem eingeschränkten Funktionsumfang zu erzeugen, eine vollständige Projektierung im Browser ist nicht möglich.

3D-Modelle und -Darstellungen werden bisher überwiegend zur Simulation und nicht zur Prozessvisualisierung eingesetzt, obwohl dies für die informationsorientierte Bedienung&Beobachtung von Anlagen/Prozessen sehr effizient wäre.

Aufgrund des Stands der Technik und des ermittelten Forschungsbedarf waren folgende Aufgabenstellungen für die zu entwickelnde Integrationsplattform zu lösen:

1. Schaffung einer mandantenfähigen und rollenbasierten Rahmen-Plattform unter ausschließlicher Verwendung von standardisierten Webtechnologien.
2. Festlegung offener Schnittstellen für eine Erweiterbarkeit durch Dritte.
3. Reduzierung der erforderlichen Konfigurationen auf der Serverseite und möglichst vollständige Systemkontrolle über die Client-Weboberfläche.
4. Anwendung und Untersuchung von WebGL zur Darstellung komplexer Anlagenteile in 3D.

Da sich die Plattform mit zukünftig geplanten Erweiterungen nicht nur für die 3D-Visualisierung eignet, sondern sich auch andere Automatisierungsfunktionen (bzw. -dienste) wie z.B. Alarmmanagement integrieren lassen, wurde die Integrationsplattform mit dem Arbeitstitel ***P**lattform **I**ndependent **A**utomation **C**ontrol **S**ystem*, kurz PIACS, bezeichnet.

3 Lösungskonzept

3.1 Architektur der Integrationsplattform

Die Architektur von PIACS nutzt Prinzipien aus dem IGF[1]-Projekt WOAS [WOAS14] und basiert auf einem zentralen Backend-Server in Java, der mittels Websockets mit einer SQL-Datenbank sowie verschiedenen Frontends kommuniziert. Bild 2 veranschaulicht diese Architektur.

Die Plattform besteht aus den folgenden fünf Hauptkomponenten:

- **PIACS-Server Backend:** Zentraler Kommunikationsknoten zwischen allen Komponenten.
- **PIACS-Client Frontend:** Benutzeroberfläche, verteilt auf Desktop und mobile Geräte.
- **HTTP-Server:** Stellt der Benutzeroberfläche die benötigten Daten und Bilder zur Verfügung.
- **SQL-Datenbank:** Dient zur Sicherung der Konfigurationen/Einstellungen der nutzerspezifischen Visualisierungsprojekte.

[1] IGF – Industrielle Gemeinschaftsforschung

- **1..n Geräte-Gateways:** Stellen Kommunikationskanäle zu Sensoren und Aktoren bzw. zu Automatisierungsgeräten bereit

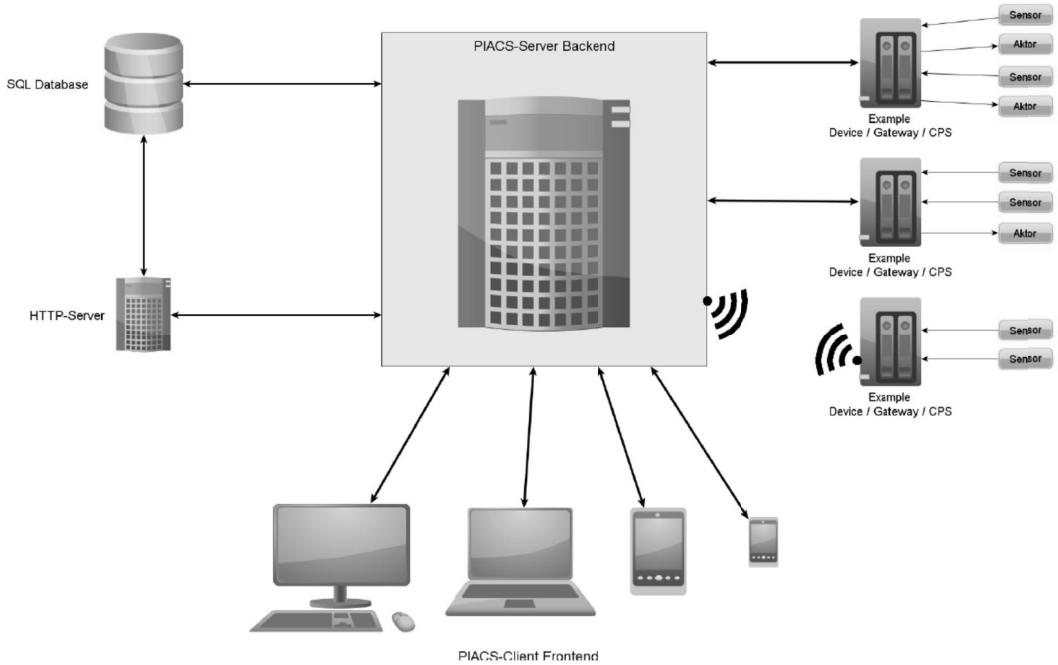


Bild 2: Architektur der Integrationsplattform PIACS

Der Ansatz einer vollständigen Realisierung auf Basis von Webtechnologien besitzt u.a. folgende Vorteile:

- Stand-Alone-Programme (Lizenzen) sind bei den Nutzern nicht mehr erforderlich.
- Keine Updates bei den Nutzern erforderlich.
- Offene und wohldefinierte Systemstruktur mit Erweiterung der Funktionalität durch Dritte.
- Es lassen sich neue Geschäftsmodelle auf Basis des Dienst-Paradigmas realisieren („Rent an Automation Service“).

3.2 Einbindung von Automatisierungsgeräten

Für die dynamische Anbindung von Automatisierungsgeräten (Sensoren, Aktoren, Steuerungen) lässt sich der Kommunikationsserver durch PlugIn‘s erweitern. Diese PlugIn‘s kommunizieren mit dem Kern des Backend Servers durch eigens hierfür entwickelte Schnittstellen. Für die Kommunikation zwischen den PlugIn‘s und den realen Geräten stehen diverse Kommunikationskanale wie Ethernet, WiFi, Serial Port, USB, Bluetooth, ZigBee, etc. zur Verfügung.

PIACS soll ein hohes Maß an Erweiterbarkeit und offene Schnittstellen bieten, welche es ermöglichen, jede Art von Geraten und Sensoren in das System einzubinden. Um

Gerate über eine verteilte Struktur und verschiedene Plattformen hinweg mit einander kommunizieren zu lassen, müssen allerdings einige Aspekte berücksichtigt werden:

Verbindung zwischen Sensoren und Aktoren

Die Verbindung verschiedener Sensoren und Aktoren unterschiedlicher Hersteller und Systeme, also eine Verbindung über verschiedene Adapter hinweg, erfolgt in PIACS über eine Verschaltungsmatrix (*Connection Group)*. Damit lassen sich sehr flexibel sowohl einfache Sensor/Aktor-Verbindungen wie auch komplexere Verbindungen mit Faktorierung bzw. Umwandlung realisieren. Auch die Verbindung mehrerer Aktoren mit einem Sensor ist dabei möglich. Bild 3 zeigt das Prinzip der Connection Groups.

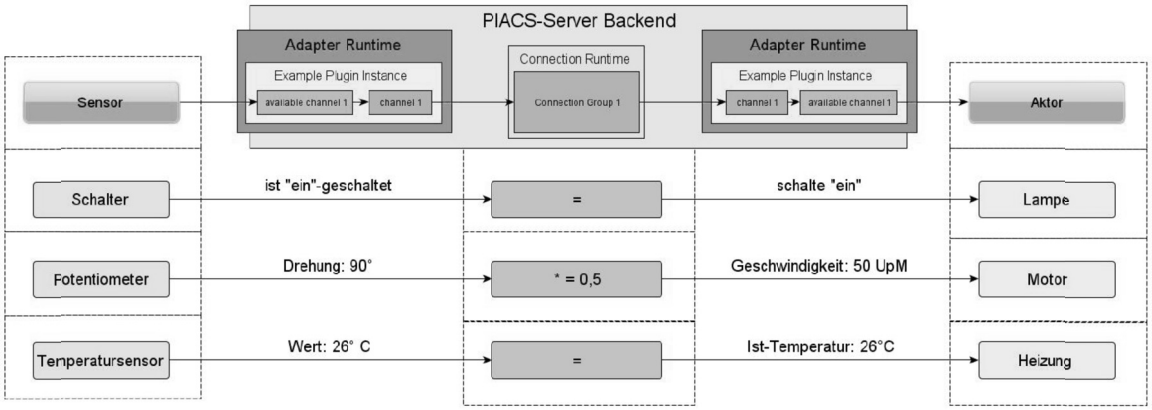


Bild 3: Beispiele für Connection Groups in PIACS

Synchronisation

Im Beispiel aus Bild 3 zeigen die Bedienelemente Potentiometer bzw. Schalter auch immer den aktuell eingestellten Wert an, da die zugehörigen Aktoren nur über diese Bedienelemente eingestellt werden können. Wird ein Motor nun mit zwei Potentiometern bzw. über zwei verschiedene Kommunikationswege gesteuert, wie es in verteilten und kollaborativen Systemen die Regel ist, so sind die beiden Potentiometer nicht synchron und geben dem Benutzer keine Information über die tatsachliche Stellgröße an. Um diese Problematik zu lösen, wurden bidirektional arbeitende Bedienelemente entwickelt, die in PIACS untereinander synchronisiert sind.

Die Integration der Geräte (Sensoren, Aktoren etc.) erfolgt in PIACS über ein separates Stand-Alone-Serverwerkzeug. Mit diesem Werkzeug werden die Geräteadapter konfiguriert und die Prozessdaten festgelegt, die in PIACS im EDIT-Mode für die Verknüpfung mit 3D- und anderen Objekten genutzt werden können.

3.3 Front-End

Mit der PIACS-Benutzeroberfläche (Front-End) ist ein Anwender in der Lage das System zu konfigurieren, zu parametrisieren und mit ihm zu interagieren. Dazu wurden drei Benutzeroberflächen für verschiedene Einsatzgebiete entwickelt:

- **Weboberfläche:** Die Weboberfläche ist eine auf HTML5, CSS3, JavaScript, WebGL und Websockets basierende Webanwendung, welche aufgrund der berücksichtigten Standards vollkommen plattformunabhängig ist. Sie unterstützt den vollen Funktionsumfang des Systems, inklusive 3D-Darstellungen, Einbindung und Parametrisierung neuer Gerate, Modifikation der Benutzeroberflache und das dynamische Verknüpfen von virtuellen Bedienelementen mit realen Geräten.
- **Java-SWING-Serveroberfläche**: Die Java-SWING Oberflache bietet einen ähnlichen Funktionsumfang wie die Weboberfläche, verzichtet aber auf aufwändige Darstellungsweisen. Sie lässt sich nur auf dem Hostrechner des Kommunikationsservers ausführen und ist somit in erster Linie für Administratoren des Systems gedacht.
- **Native Android-Oberfläche**: Die native Android Oberfläche bietet nur die wichtigsten Basisfunktionalitäten und ist somit bestens für ältere Smartphones und moderne tragbare Endgeräte (wearables) wie beispielsweise Smartwatches gedacht.

Die Anbindung der verschiedenen Benutzeroberflächen erfolgt, ähnlich wie bei den Geräten durch ein eigens dafür entwickeltes Adapter-System. Für die Verbindung zu einer Benutzeroberfläche wird jeweils immer nur eine Instanz eines solchen Client-Interface benötigt, es gibt aber auch Benutzeroberflächen, die mehrere Verbindungsarten unterstützen, um dem Benutzer möglichst viel Freiheiten zu lassen.

3.4 2D/ 3D-Visualisierung

Neben der experimentellen 3D-Darstellung von Prozessdaten, sollen in PIACS auch klassische 2D-grafische Elemente (z.B. Bedienelemente) verwendet werden. Um eine konsistente Kombination aus beidem zuzulassen, müssen die 2D-Elemente auch im 3D--Raum positionierbar sein. Zu diesem Zweck wurde mit dem CSS3-Webstandard die Möglichkeit geschaffen normale HTML-DOM-Elemente mit 3D-Transformationen zu versehen (CSS-3D-Transformation). Gerade bei der Kombination mit 3D-Modellen entstehen dabei aber einige Komplikationen. Die Problematik ist nicht neu und in der Computergrafik schon seit einigen Jahren kein Thema mehr. Im Bereich der Webtechnologien ist diese Technik aber neu und bringt aufgrund des Mischens von DOM und 3D-Elementen neue Herausforderungen mit sich.

Als 3D-Engine wurde für PIACS *Three.js* [THREE14] verwendet, da dieses als einzige Engine neben WebGL-Transformationen auch CSS3D-Transformationen beherrscht. Die Verwendung von WebGL zur Darstellung von Benutzeroberflachen, also die Kombination aus 3D- und 2D-Elementen, stellt jedoch in gewisser Weise eine Zweckentfremdung dar, da die meisten Engines nur für die reine 3D-Darstellung konzipiert sind.

Folgende Probleme mussten deshalb in PIACS durch z.T. umfangreiche zusätzliche JavaScript-Berechnungen gelöst werden.

- Kamerabewegungen,
- Positionierung von Objekten,
- 2D-Positionierung von Objekten mit variabler Tiefe,
- Dynamische Höhe des Sichtfelds.,

Eine ausführliche Beschreibung der Probleme und der dazu erforderlichen Lösungsalgorithmen findet sich in [MEY14].

Die Interaktion mit den 3D-Objekten in PIACS erfolgt per Maus bzw. per Touch (Zoomen, Schwenken etc.) sowie mit verfügbaren 2D-Bedienelementen (z.B. Knopf, Schieberegler), die Prozessdaten und/oder 3D-Objekte beeinflussen können.

4 Anwendung

Zur Einbindung von unterschiedlichen Automatisierungsgeräten in den Kommunikationsserver von PIACS sind PlugIn‘s für die Industrieschnittstellen OPC DA, OPC UA und Modbus TCP verfügbar. Darüber hinaus können auch Geräte aus der Gebäudeautomatisierung bzw. Smarthome in PIACS eingebunden werden. Dazu gehören u.a. PlugIn‘s für das Funksystem FS-20, Homematic, X10 und Pioneer Receiver. Um PIACS möglichst flexibel zu halten und es auch anderen Entwicklern zu ermöglichen das System zu erweitern, wurden weiterhin zwei Adapter entwickelt, die ein einfaches ASCII-Zeichen Protokoll verwenden, um mögliche Kanäle und Prozessdaten zu empfangen und zu steuern.

- **ASCII-Seriell:** Dieser Adapter verwendet die serielle Schnittstelle und ist somit ideal geeignet, um Mikrocontroller Projekte in das System mit einzubinden.
- **TCP:** Der andere Adapter verwendet die TCP-Schnittstelle, um mit Geräten zu kommunizieren, welche sich nicht in der Nähe des Kommunikationsservers befinden. Dieser Adapter ermöglicht es auch, auf unkomplizierte Weise Adapter und Geräteanbindungen in anderen Programmiersprachen zu ermöglichen

Die Parametrierung des Kommunikationsservers erfolgt Stand-Alone auf der Server-Seite über einen in Java realisierten Konfigurator.

PIACS ist für Testzwecke öffentlich unter `http://piacs.ccad.eu` verfügbar. Nach Einloggen in das System als Administrator können die verfügbaren 3D-Objekte und 2D-Bedienelemente konfiguriert und mit Echtzeit-Prozessdaten verknüpft werden. Mit den im System vorhandenen Objekten kann die Zone C (Fertigungsautomatisierung) der Forschungs- und Trainingsfabrik Fab21 im Competence Center Automation Düsseldorf (CCAD) abgebildet und dynamisch visualisiert werden (Bild 4).

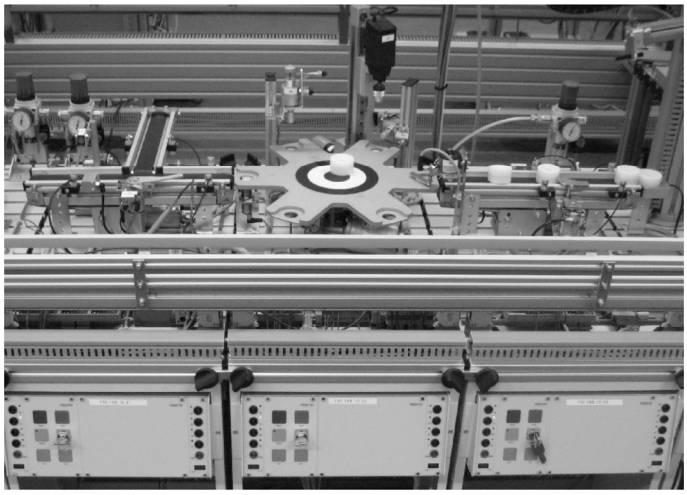


Bild 4: Zone C der Forschungs- und Trainingsfabrik Fab21

Für die dynamische Visualisierung werden die Prozessdaten aus einem OPC-Simulations-Server genutzt, mit dem die Prozessdaten-Abläufe in Zone C in Echtzeit simuliert werden können.

Der Administrator kann neue Objektinstanzen erzeugen, diese parametrieren und mit Prozessdaten verknüpfen. Bild 5 zeigt dazu die Weboberfläche von PIACS im EDIT-Modus nach Auswahl der Facility „Fab21“.

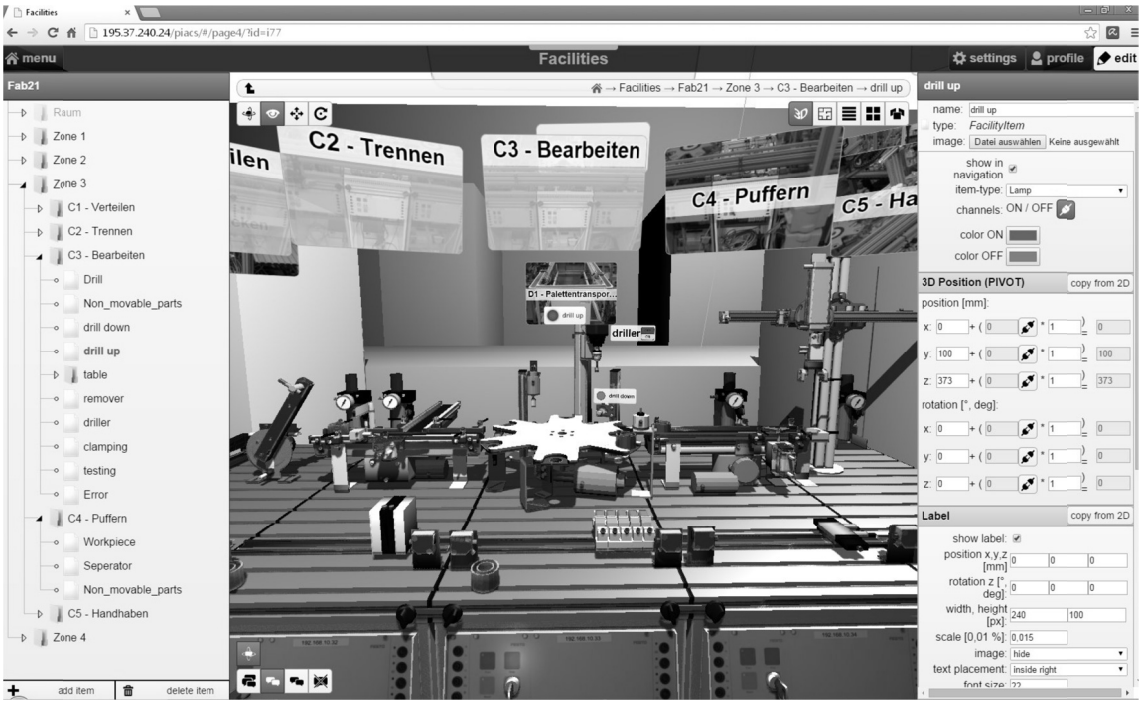


Bild 5: Weboberfläche von PIACS im EDIT-Modus

Die Objektdarstellungen sind strukturiert nach:

- **Facilities:** Zusammengefasste 2D/3D-Objekte zur Darstellung und Manipulation von Prozessdaten.
- **Items:** Darstellungs- und Bedienelemente, wahlweise in 2D und/oder 3D.
- **Adapter:** Verfügbare Adapter und Geräte im System.

Auf der linken Seite in Bild 5 befindet sich eine Auflistung aller Items der jeweiligen Facility. Der mittlere Teil beinhaltet das Darstellungsfeld für die zu konfigurierende Benutzeroberfläche und über den rechten Teil können die Items parametriert und mit Prozessdaten dynamisiert werden.

Die Implementierung der PIACS-Weboberflache nutzt die standardisierten Technologien HTML5, CSS3, JavaScript, WebGL und die beiden Frameworks jQuery und Three.js. Alle Icons und Bilder wurden als SVG-Vektor-Grafiken erstellt und in eine JavaScript-Datei eingebettet. Dies hat nicht nur den Vorteil, dass die gesamte Benutzeroberflache auch auf hochauflösenden Bildschirmen nicht an Schärfe verliert, sondern dass sich Farben und Verhalten der Grafiken auch über CSS steuern lassen und somit ein individuelles und responsive Design möglich ist.

Durch die offenen und konsistenten Schnittstellen für die Integration von Geräten und die Integration von neuen 3D-Objekten sowie Bedien- und Visualisierungselementen in die Plattform lässt sich PIACS einfach durch Dritte erweitern. Damit bietet die Plattform das Potential um im Sinne von Industrie 4.0 als eine CPS-Integrationsplattform für eine virtuelle Prozesswelt zu fungieren.

5 Resümee und Ausblick

Die Nutzung von WebGL und dem *.obj-Format für die 3D-Objekte besitzt gegenüber dem bisher für ähnliche Zwecke genutzten Lösungen mit VRML[2] [LANG13] den wesentlichen Vorteil, dass keine Player-PlugIn‘s für den Browser erforderlich sind und mit HTML5 die 3D-Objekte direkt angezeigt werden können. Im Vergleich zu VRML ist beim *.obj-Format allerdings kein Scripting innerhalb der Objekte möglich, so dass eine interne Verhaltenssimulation technischer 3D-Objekte nur schwierig zu realisieren ist. Auch die Dynamisierung einzelner 3D-Elemente mit Prozessdaten kann damit nur aufwendig über zusammengesetzte Einzelobjekte erfolgen. Zukünftig sollen eine dynamische Verhaltenssimulation über JavaScript in Verbindung mit den entsprechenden 3D-Objekten erfolgen und diese Verhaltensmuster als Dienste in PIACS integriert werden.

Das Laden der Arbeitsumgebung des Integrationsportals ist aktuell noch zeitaufwendig und benötigt auch einen ausreichend leistungsfähigen Client-Rechner. Dies soll zukünf-

[2] VRML - Virtual Reality Modelling Language

tig noch optimiert werden. Nach Laden der Arbeitsumgebung kann dann allerdings flüssig mit der 3D-Objektumgebung im Browser gearbeitet werden.

Literatur

[KAG14] KAGERMANN, H.; WAHLSTER, W.; HELBIG, J.: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. – Abschlussbericht des Arbeitskreises Industrie 4.0, Forschungsunion Wirtschaft – Wissenschaft, 2013

[GMA14] VDI/VDE-GMA: Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation, 2013

[MOV14] Movicon Next: http://www.progea.com/dede/produkte/automationplatformnext/moviconnext.aspx, 2014

[GEN14] Genesis64: http://www.iconics.com/Home/Products/HMI-SCADASoftware-Solutions/GENESIS64.aspx, 2014

[WOAS14] CCAD: Abschlussbericht WOAS. - http://195.37.242.50/downloads/WOAS/WOAS_Schlussbericht_2014.pdf, 2014

[THREE14] Three.js: http://threejs.org/, 2014

[MEY14] MEYER, L.: Architektur und Entwicklung eines webbasierten, verteilten und kollaborativen Steuerungs- und Uberwachungssystem, fur die 3DAnlagenautomatisierung, auf Grundlage von WebGL – Masterthesis, FH Düsseldorf, 2014

[LANG13] LANGMANN, R.: Web-based Simulation of Mechatronic Systems. – Konferenzband, 14th International Workshop on Research and Education in Mechatronics REM 2013, Wien, 06.-07.06.2013, S. 56- 60

Autoren

Prof. Dr.-Ing. Reinhard Langmann ist seit 1993 als Professor an der Fachhochschule Düsseldorf in der Automatisierungstechnik tätig. Dr. Langmann ist Sprecher des Competence-Center Automation Düsseldorf (CCAD) sowie Leiter der Forschungs- und Trainingsfabrik Fab21 und des Düsseldorfer Telelabors. Sein Interessensschwerpunkt liegt auf dem Gebiet der Internettechnologie für die Industrieautomation. Seit 2008 ist er 1. Vorsitzender des Vereins für Angewandte Automatisierungstechnik in Lehre und Entwicklung an Hochschulen VFAALE e.V.

Prof. Dr. Christian Geiger ist seit Ende 2004 als Professor für Mixed Reality und Visualisierung an der Fachhochschule Düsseldorf tätig. Er ist Sprecher des Forschungsschwerpunkts Nutzerwelten und forscht in den Gebieten Mensch-Technik-Interaktion, 3D-Visualisierung und Augmented Reality.

AR für mobile Assistenzsysteme

Zugriffsgesicherte Augmented Reality Lösung zur mobilen Instandhaltungsunterstützung mit zustandsorientierten Arbeitsanweisungen

Matthias Neges, Mario Wolf, Michael Abramovici
Lehrstuhl für Maschinenbauinformatik (ITM), Ruhr-Universität Bochum
Universitätstrasse 150, 44780 Bochum
Tel. +49 (0) 234 / 32 22 109, Fax. +49 (0) 234 / 32 14 443
E-Mail: matthias.neges@itm.rub.de

Zusammenfassung

Technische Anlagen und Produktionsstätten sind in aller Regel komplexe Systeme, die aus einer heterogenen Landschaft von Subsystemen bestehen. Die technischen Dokumentationen und Instandhaltungshistorien stehen bei Wartungstätigkeiten häufig nicht vollständig oder nur in Papierform zur Verfügung, während der aktuelle Status der Anlage nicht mit den vorhandenen Informationen überlagert werden kann um den die Durchführung der geplanten Aktion zu vereinfachen. In diesem Beitrag wird ein Ansatz präsentiert, der Servicetechnikern eine interaktive, intuitive und protokollierbare Methode zur Verfügung stellt, auch an unbekannten Systemen Instandhaltungsmaßnahmen vorzunehmen. Hierzu wird über Graphen die Funktion des Systems abgebildet und algorithmisch eine Problemlösung nach Aufnahme des Ist-Zustandes bereitgestellt, die den Techniker schrittweise in der Herstellung eines bestimmten Betriebszustands unterstützt. Das hier dargestellte Konzept nutzt Augmented Reality (AR) nicht nur als reines Visualisierungswerkzeug für Informationen und Arbeitsanweisungen, sondern ebenfalls für die Aufnahme des Ist-Zustandes und die Erstellung von erweiterten, multimedialen Serviceberichten. Das vorgestellte Konzept umfasst insgesamt sechs Phasen: den über RFID authentifizierten Zugriff auf die digitalen Daten, die Synchronisation eines mobilen Endgerät mit dem Server, die Bereitstellung von Informationen über AR, die Aufnahme des Ist-Zustandes über AR, die algorithmische Problemlösung, die Bereitstellung von intuitiven AR-Arbeitsanweisungen zur Herstellung des Betriebszustandes und die Dokumentation der durchgeführten Arbeiten in AR-Serviceberichten.

Schlüsselworte

Augmented Reality, mobile Geräte, Kennzeichnungstechnik, Authentifizierung, Instandhaltung

Secure Access Augmented Reality Solution For Mobile Maintenance Support Utilizing Condition-Oriented Work Instructions

Abstract

In most cases, technical facilities and manufacturing plants are complex systems with a heterogeneous variety of subsystems. Technical documentation and maintenance protocols are often either incomplete, not available on site or available only in print form. Additionally there is often no information about the current state of the maintenance object, which could be used to analyze and enhance the task at hand. In this paper, we present an approach that offers service technicians an interactive, intuitive and documentable way to maintain unknown machinery. To realize context awareness, we use graph-based algorithms to guide the technician in achieving the machine's desired operating condition after evaluating the current state of the machine, once again with guidance from our mobile application. Our approach makes use of augmented reality not only as a tool for visualization, but also as interactive input method to capture the current machine state and to generate multimedia maintenance reports, i.e. by taking pictures with redlined components and saving them in their actual context Our concept is divided into six phases: Rfid-based authorized access to maintenance relevant data, synchronization of data between server and mobile application, visual provisioning of context relevant data, capturing the current state of the maintenance object, provisioning of intuitive stepwise instructions to achieve certain operating conditions and the documentation of executed actions, used materials, tools etc.

Keywords

Augmented Reality, mobile devices, marking technology, authentication, maintenance

1 Einleitung

Im Laufe des letzten Jahrzehnts haben mobile Endgeräte (Smart Devices), wie Smartphones und Tablets, den Markt vollkommen durchdrungen [AGN14]. Diese Geräte ermöglichen weitreichende Anwendungen in den Bereichen von Mobile Computing, mobiler Kommunikation und des mobilen Internets.

Wenn man die Ansätze des Internets der Dinge mit verschiedenen Typen von Mobilgeräten nutzen möchte, ergeben sich aus den individuellen Eigenschaften der Gerätetypen auch verschieden Einsatzzwecke. Betrachtet werden hier sowohl Smartphones, die durch ihre große Verbreitung besonders in den Fokus rücken. Tablets, die durch die größeren Displays bessere Möglichkeiten für umfangreiche Darstellung bieten und Head Mounted Devices (HMDs), wie Google Glass, die eine uneingeschränkte Nutzung beider Hände ermöglichen. Kommunikationsstandards wie Universal Mobile Telecommunications System (UMTS), Long Term Evolution (LTE), und Bluetooth, zusammen mit den Möglichkeiten von Kennzeichnungstechnologien ermöglichen den nächsten Schritt in Richtung Ubiquitous Computing, bzw. Internet der Dinge [COO13] .

Um beispielsweise eine Wartungsaufgabe im Rahmen einer Instandhaltung durchführen zu können, muss sich der Servicetechniker zunächst einen Überblick über das Instandhaltungsobjekt und dessen Funktionsweise verschaffen, was wiederum zu einem erhöhten Zeitaufwand führt [WF07]. Dabei kann der Einsatz von Smart Devices und Technologien wie Augmented Reality (AR) den Servicetechniker bei der Durchführung von komplexen Wartungsaufgabe unterstützen [vKP10]. Eine kontextsensitive Unterstützung eines Wartungsprozesses mit AR beispielsweise ermöglicht bei einem komplexen Rohrleitungssystem eine Reduzierung der Arbeitszeit um bis zu 55% und halbiert gleichzeitig die Fehlerrate [HWT13].

Um sicherzustellen dass die nötigen Arbeiten an kritischen Maschinenteilen tatsächlich ausgeführt werden, ist eine Authentifizierung des Technikers vor Ort, direkt an der betroffenen Anlage erforderlich [LT06]. Der Sinn der Authentifizierung liegt in der eindeutigen Identifizierung des Technikers und der zu wartenden Maschine. Die überwiegende Mehrheit der Smart Devices verfügt über Kameras und erlaubt so die Nutzung von optischen Kennzeichnungstechnologien wie QR-Codes, Datamatrix oder klassischen Barcodes. Geräte aktuellerer Generationen verfügen zum Teil über Schnittstellen zur Nutzung von elektromagnetischen Kennzeichnungstechnologien wie Near Field Communication (NFC) oder Radio Frequency Identification (RFID), die auch eine kryptographisch sichere Authentifizierung erlauben.

Die hier vorgestellte Lösung geht über die heute üblichen Instandhaltungsdokumentationen hinaus, da nicht nur eine Darstellung bereits vorhandener Daten realisiert werden soll, sondern auch eine aktive Analyse des aktuellen Zustands der betreffenden Anlage. Unser Fokus liegt auf dem Abgleich des erfassten Ist-Zustands mit einem hinterlegtem Logikmodell, durch den kontextsensitive Anweisungen zur Verfügung gestellt werden

können. Weiterhin unterstützen die multimediale Instandhaltungsreports dabei sowohl den Servicetechniker bei der Arbeit, als auch die Dokumentation im Nachgang.

2 IT-Ansätze für die Instandhaltung

2.1 Computerized Maintenance Management Systems

Instandhaltungsorientierte Firmen und Dienstleister setzten heutzutage auf sogenannte CMMS, Computerized Maintenance Management Systems [MRP+10], welche auch als EAM (Enterprise Asset Management) bekannt sind. Diese Systeme haben die folgenden Aufgaben:

- Management von Zeitplänen für die Instandhaltung
- Auftragsverwaltung mit Planung, Kosten und Dokumentation
- Asset Management: Verwaltung von Maschinen, Spezifikationen, Checklisten, Garantieinformationen, Inspektionsintervalle, etc.
- Lagerverwaltung: Verwaltung und Organisation von Ersatzteilen, Werkzeugen und Betriebs- und Verbrauchsmaterial
- Statistikanalyse für einzelne Maschinen
- Unterstützung eines kontinuierlichen Verbesserungsprozesses Betriebskennzahlen nach [DIN 31051] Soll-/ Ist-Wert-Vergleich], Asset-Kostenüberwachung)

CMMS sind typischerweise als Desktop-Applikationen ausgelegt. Dies ermöglicht nur eingeschränkte Nutzbarkeit der Daten auf einem Mobilgerät, da besonders die erheblich kleinere Display-Größe, als auch die Bedienung und Einsatzumstände eine vereinfachte und optimierte Darstellung bedingen. Ansätze für derlei Optimierung werden in [SG13] und [BKK+10] diskutiert.

2.2 RFID-Einsatz in der Instandhaltung

Die Nutzung von Auto-ID-Verfahren mit Smart Devices ermöglicht die eindeutige Identifikation von Anlagen, Maschinen und Teilen und ist im Gegensatz zur manuellen Eingabe von z.B. Bezeichnungen nicht anfällig für Fehler [Fin02].

Weiterhin können mit Kennzeichnungstechnologien auch zusätzliche Nutzdaten gespeichert werden, wie z.B. ganze URLs zu weiteren Ressourcen, jeweils bezogen auf das konkrete, gekennzeichnete Produkt. Diese zusätzlichen Ressourcen können mithilfe von modernen Mobilgeräten abgerufen und genutzt werden. Dabei kann entweder auf auf dem Gerät gespeicherte Daten zugegriffen werden, oder die mobile Internetverbindung für den Datenabruf aus einer Datenbank genutzt werden. Die Nutzung von Kennzeichungstechnologien mit Smart Devices kann also die Referenzierung von multimedialem Inhalt und damit die Verknüpfung von realer und virtueller Welt verstärken und

somit die Vision des Internets der Dinge zum Teil umsetzen [FF05], [EOP07]. Wie man [MRP+10] entnehmen kann, gibt es verschiedene erprobte Arten die für die Instandhaltung relevanten Informationen zu speichern und verfügbar zu machen. Sogenannte RFID-Tags bieten sogar die Möglichkeit lokal, direkt an der jeweiligen Maschine, Daten zu hinterlegen. Doch anstatt sich auf RFID als Träger-Technologie festzulegen, soll die vorgestellte Lösung mit verschiedenen, vor allem aber optischen Arten von Kennzeichnungstechnologien einsetzbar sein.

2.3 Entscheidungsfindung in Arbeitsplänen für die Instandhaltung

In einer aktuellen Untersuchung haben Schlick et al. [SJM14] die Lernerfolge abhängig vom Instruktionsmedium analysiert. Ergebnis der Studie ist dabei der bessere Erfolg von grafischen Darstellung bei Aufgaben, die mit nur geringer Wiederholrate (im Vergleich zum Beispiel zu Fließbandarbeiten) ausgeführt werden. Diese Lernerfolge sprechen umso mehr für eine Modernisierung der bisher üblichen Arbeitsanweisungen und könnten ebenso mit einer standardisierten Symbolik (vgl. [VDI2860]) kombiniert werden. Arbeitsanweisungen sind heutzutage häufig noch schrittweise und in Textform ausformuliert. In aller Regel beschränken sich die Auswahlen auf einfache Entscheidungen zwischen zwei Optionen. Die Entscheidung welcher Schritt als nächster zu tun ist, kann allerdings über verschiedene andere Methoden getroffen werden, die eine größere Interaktivität erlauben würden [AKW14].

2.4 AR-Einsatz in der Instandhaltung

Augmented Reality (AR) ermöglicht eine Erweiterung dieses digitalen Ansatzes und kombiniert die virtuellen 3D-Modelle und Informationen mit dem realen Produkt. Dies kann mit Hilfe von Smart Devices, wie z.B. HMDs, Smartphones oder Tablets, realisiert werden. Um den Blickwinkel des Betrachters erfassen zu können, werden in der Regel künstliche Marker verwendet. Dies sind sehr markante Bilder mit bekannten visuellen Mustern und Dimensionen. Basierend auf grafischen Erkennungsalgorithmen werden diese Marker in Echtzeit erkannt und als Referenz-Objekte benutzt, um virtuelle Objekte in das Brillendisplay oder in die Live-Ansicht der Kamera einzublenden. Große Projektinitiativen wie ARVIKA, STAREMATE oder ARTESAS beschäftigten sich mit der Weiterentwicklung und den Einsatzpotentialen von AR im industriellen Umfeld. Vorhandene AR-Lösungen für Montage- und Instandhaltungsaufgaben setzen jedoch eine proprietäre Hard-und Software ein, sind aber bedingt durch höheres Gewicht unhandlich und berücksichtigen vordefinierte Umgebungen. Es existieren jedoch ebenso Instandhaltungsaufgaben, die in einem für den Servicemitarbeiter unbekannten Umfeld erfolgen müssen und eine leichte, benutzerfreundliche Standard-Hardware sowie eine flexible, kostengünstige Software erfordern. Als Beispiel kann hier die Wartung eines komplexen Rohrleitungssystems in öffentlichen Gebäuden genannt werden. Aktuell verfüg-

bare HMDs können dabei keine zufriedenstellende Nutzung garantieren. Aus diesem Grund werden Tablets und Smartphones zunehmend für AR-Anwendungen eingesetzt.

Die AR-Technologie führte in den letzten Jahren zur Entstehung zahlreicher Softwarefirmen, die sich mit kommerziellen Software-Lösungen zur Erstellung von AR-Anwendungen, wie das metaioSDK oder Vuforia, beschäftigen. Diese bedienen sich der Ergebnisse aus den genannten Projekten und bieten somit bereits robuste, meist optische, Tracking-Verfahren an, welche vollständig in eigenen Programmbibliotheken implementiert sind. Ein stabiles 2D-Tracking von verschiedenen Marker-Typen, sowie ein punktbasiertes und ein kantenbasiertes 3D-Tracking gehören mittlerweile mit zum Funktionsumfang.

3 Vorhergehende Arbeiten

3.1 Casestudy mit bebilderten Anleitungen

Im Rahmen einer Fallstudie in Zusammenarbeit mit dem Technischen Hilfswerks (THW) Sektion Bochum wurden anhand der Untersuchung von schwerem Bergungsgeräten Methoden und Lösungen getestet, die es ermöglichen die Anwender bei der Inbetriebnahme und Bedienung dieser Geräte zu unterstützen. Zu diesem Zweck wurden Kennzeichen an den Geräten angebracht, die vom Benutzer über ein Smartphone eingelesen werden konnten. Dies sicherte einen schnellen Einstieg in die richtige Dokumentation für das entsprechende Gerät und dessen Bedienung.

Die Kurzdokumentation wurde in Form eines PDF-Dokuments hinterlegt und aufgrund der im Kennzeichen gespeicherten URL abgerufen. Die Dokumentation enthält schrittweise bebilderte Anweisungen (Fotos) mit Annotationen, die den Benutzer bei jedem wesentlichen Arbeitsschritt unterstützen. Hierbei wurden keine Zustandsaufnahmen oder vorliegende Fehlfunktionen der Geräte berücksichtigt. Das Hauptaugenmerk dieser Studie lag in der Gegenüberstellung von Erfahrungswerten gegenüber einer kompakten Anleitung. Zwei Testgruppen mit jeweils gleichen Inbetriebnahme-Aufgaben wurden gegenübergestellt, um die Gesamtzeit, die Fehler(-anfälligkeit) und das Sicherheitsgefühl der Benutzer über eine anschließende Befragung gegenüber der traditioneller Dokumentation oder einer freien Bedienung zu bewerten.

Die Studie zeigte sehr gute Erfolge bei der Anwendung der Methode, da die Benutzergruppe mit Vorwissen zwar minimal länger für den Ablauf brauchte, aber eine erheblich niedrigere Fehlerrate aufwies. Die Benutzergruppe ohne Vorwissen hingegen war sogar erst mit der speziell angefertigten Anleitung in der Lage die Maschinen in Betrieb zu nehmen.

3.2 AR-Arbeitsanweisungen und AR-Serviceberichte

Die Autoren publizierten in [AN14], [AN13] einen Ansatz zur Erstellung von Schritt-für-Schritt AR-Arbeitsanweisungen. Hierbei wurde über ein 3D-Viewer die Generierung von einzelnen Arbeitsschritte unter Berücksichtigung einer Tätigkeitsbeschreibung, der entsprechenden Visualisierung der betroffenen Bauteile und zusätzliche visuellen Hinweisen in Form von z.B. Richtungspfeilen oder 3D-Kommentarlabels intuitiv ermöglicht. Die so generierten Arbeitsanweisungen können über eine formalisierte Struktur als XML exportiert und so in eine AR-Anwendung auf dem Smart Devices umgesetzt werden. Neben der reinen Visualisierung der AR-Arbeitsanweisungen berücksichtigt das Konzept die Erstellung von Serviceberichten und nutzt damit die bereitgestellten Informationen für eine genaue Problembeschreibung. Der Servicetechniker kann aufgrund dieser Informationen betroffene Bauteile über ein überlagertes, virtuelles 3D-Modell auswählen und Defekte, wie z.B. Bruchstellen, mittels kommentierter Fotos dokumentieren. Die so erstellten Serviceberichte liefern somit, in einer ebenfalls formalisierten Struktur, Informationen über den realen Zustand des Instandhaltungsobjektes, welche wiederum durch den 3D-Viewer interpretiert werden können. Im Mittelpunkt der dargestellten Lösung steht ein Web-basiertes Managementsystem zur Verwaltung der Daten, der Arbeitsaufträge und Synchronisation zum Smart Device.

Eine detaillierte Evaluation der kommerziellen AR-Werkzeuge zeigt hier, dass die implementierten Tracking-Verfahren die erforderliche Robustheit und Genauigkeit für den Einsatz in AR Instandhaltungsprozessen liefern können und somit die Basis für industrielle AR-Anwendungen liefern.

3.3 Zustandsorientierte Arbeitsanweisungen

Im Rahmen des Kompetenzzentrums für hydraulische Strömungsmaschinen der Ruhr-Uni Bochum wurde in [AKW14] ein Konzept entwickelt, mithilfe dessen verschiedenartige, interaktive Instandhaltungssupportmethoden auf Mobilgeräten zur Verfügung gestellt werden können. Besonders die Berücksichtigung des aktuellen Maschinenzustands spielt in diesem Ansatz eine entscheidende Rolle. So wird dem Benutzer die Möglichkeit gegeben, sowohl eine Analyse des aktuellen Zustands, als auch an den Zustand angepasste Arbeitsanweisungen zu erhalten.

Die durchgeführte Casestudy mit über 160 Teilnehmern ohne besonderen technischen Hintergrund wurde an einem eigens konstruierten Pumpenteststand anhand von zwei Szenarios durchgeführt (siehe Kapitel 5). Das erste Szenario umfasste eine Ist-Zustandsaufnahme inkl. Fehlerdetektion. Die Wiederherstellung eines funktionalen Zustands des Systems durch Änderung der Konfiguration wurde im zweiten Szenario fokussiert. Durch Verwendung eines NFC (Near Field Communication)-fähigen Tablets und entsprechender Kennzeichen wurde die korrekte Erkennung des Systems sichergestellt. Während der Durchführung der Studie standen den Probanden auf dem Tablet sowohl technische Dokumentationen als herstellergeliefertes Dokument, als auch eine

dynamische Ist-Zustands-Analyse zur Verfügung. Auf Basis des erfassten Ist-Zustands wurden daraufhin mögliche sofortige Wiederinbetriebnahme-Anleitungen auf Basis einer algorithmischen Auswertung angeboten.

Ohne technische Dokumentation war kein Proband in der Lage mit Sicherheit einen bestimmten Betriebszustand zu erreichen. Durch die Verwendung der entwickelten Applikation wurde eine 100%ige Erfolgsquote bei Fehlerdetektion und Wiederinbetriebnahme erreicht.

3.4 Bewertung

Textuell umfangreiche und grafisch aufwändige Beschreibungen sind präzise und erfolgsversprechende Anleitungen. Sie beanspruchen jedoch Zeit, um vom Benutzer vom digitalen Medium auf die Realität übertragen werden zu können. Es mangelt an Orientierung in der Realität. AR-gestützte Anleitungssequenzen stellen diesen direkten Bezug zwischen Realität und digitalen Informationen her. Um flexibel auf den aktuellen Zustand der Maschine reagieren zu können und situationsabhängige Anleitungen zur Problemlösung bereitstellen zu können, müssen diese Ansätze jedoch erweitert werden. Hierzu kann AR nicht nur als reine Visualisierungstechnik, sondern vielmehr auch für die interaktive Aufnahme des Ist-Zustandes eingesetzt werden. Die Ist-Zustandsanalyse bietet deutlich erhöhte Sicherheit im Umgang mit einer unbekannten Maschine und erlaubt eine angemessene Reaktion auf Fehlerfälle.

4 Konzept

Bild 1 zeigt eine Übersicht über das Gesamtkonzept und dessen sechs Phasen. Nach der Authentifizierung über z.B. einen QR-Code oder ein RFID-Tag, erhält der Servicetechniker Zugriff auf die Instandhaltungsunterlagen (Phase 1). Darin enthalten sind neben dem digitalen 3D-Modell, welches für die spätere AR-Visualisierung benötigt wird, und dem Logikmodell, auf dessen Basis eine Zustands-Analyse erfolgen kann, auch weitere beliebige Dokumentation, wie Instandhaltungsberichte oder technische Datenblätter (Phase 2). Auf Grundlage dieser Informationen und Daten erhält der Servicetechniker einen Überblick über das ihm vorliegende System. Wie von den Autoren in [AN14] dargestellt, kann eine Erkennung des realen Objektes und die kontextspezifische Einblendung der Informationen und digitalen 3D-Modelle durch eine Kombination aus punktbasierten und kantenbasierten Erkennungsverfahren realisiert werden (Phase 3). Durch die Überlagerung des realen Rohrleitungssystems mit dem digitalen 3D-Modell erhält der Servicetechniker zudem die Möglichkeit, durch die Markierung der entsprechenden Elemente auf dem Display den Ist-Zustand des Systems aufzunehmen. Hierzu wird der Zustand der Elemente, wie z.B. *„Ventil geöffnet“*, *„Ventil geschlossen“*, *„Ventil defekt“*, erfasst und für die Analyse bereitgestellt (Phase 3).

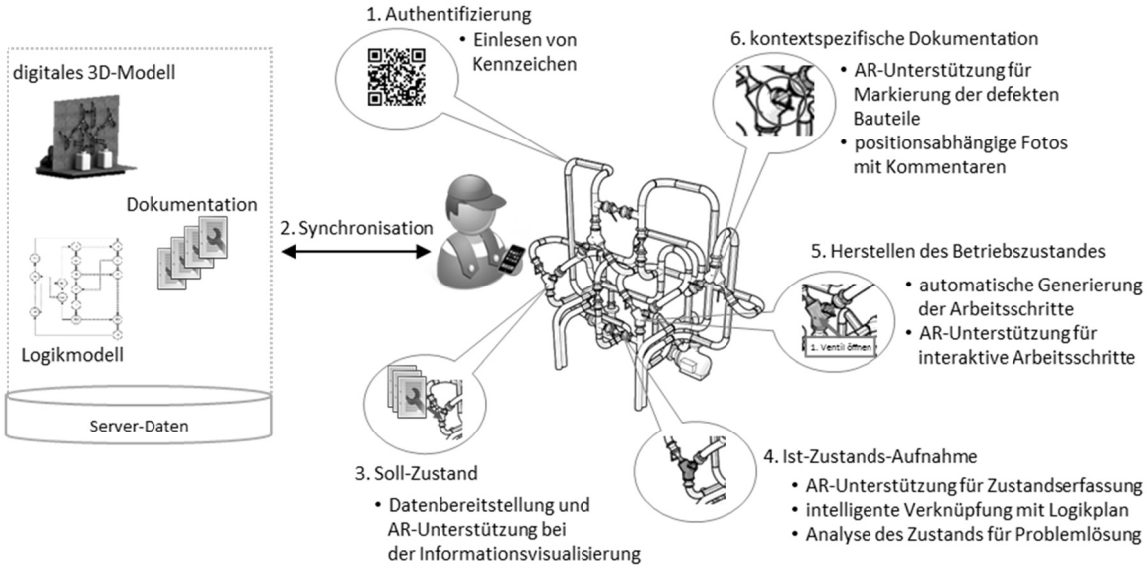


Bild 1: Die sechs Phasen des Gesamtkonzepts

Durch die interaktive Ist-Zustands-Aufnahme über AR lässt sich eine Relation zur Funktion und Logik des Systems herstellen. Dies ermöglicht eine algorithmische Auswertung und Problemfindung auf Basis von Graphen. Diese Analyse bildet den Fokus des Beitrages und wird daher detailliert in Kapitel 4.1 und 4.2 beschrieben. Die Ergebnisse der Analyse lassen sich nachfolgend auf die Generierung der Arbeitsanweisungen übertragen. Dies setzt eine starke Modularisierung der Arbeitsschritte voraus, um diese automatisch auf Basis des vorliegenden Ist-Zustandes zu generieren (Phase 4). In [AN14] stellten die Autoren bereits einen Ansatz für die Visualisierung von AR-Arbeitsanweisungen und AR-Serviceberichte vor. Diese Ansätze lassen sich jedoch im Hinblick auf die erweiterte Aufnahme des Ist-Zustandes des Systems erweiterbar. Sind Änderungen, z.B. die Veränderungen von Ventilstellungen, erforderlich um die korrekte Funktionsweise des Systems oder eine generelle Anpassung der Funktionsweise herzustellen, so kann der aufgenommene Ist-Zustand ebenfalls für einen Servicebericht verwendet werden. Der Bericht enthält somit z.B. alle Ventilstellungen oder eine Aussage über die Funktionstüchtigkeit der einzelnen Elemente. Da während des gesamten Instandhaltungsprozesses eine kontinuierliche AR-Unterstützung gewährleistet werden kann, sind die Position und der Blickwinkel des Servicetechnikers in Relation zum physikalischen System stets auslesbar. Dies ermöglicht es, positionsbezogene Fotos zu erstellen und diese mit Kommentaren und Redlinings zu versehen, welche wiederum auf das virtuelle Modell zurückgeführt werden können (Phase 5 und 6).

Der Fokus dieses Beitrages liegt, wie bereits beschreiben, zunächst auf der Analyse des Ist-Zustandes und der Verknüpfung der Informationen zwischen Funktion, Logik und Realität.

4.1 Analyse des Ist-Zustands

Ist die genaue Funktionsweise einer Maschine oder Anlage bekannt, kann diese als Graph abgebildet werden, welcher wiederum durch Algorithmen zur Laufzeit ausgewertet werden kann. Rohrleitungssysteme eignen sich hier besonders gut. Kürzeste-Pfade-Algorithmen (genauer Single Pair Shortest Path, SPSP) können eingesetzt werden, um bei vorhandener Redundanz im Rohrsystem, trotz Fehlerfall, einen bestimmten Fluss zu berechnen.

Weiterhin erlaubt der algorithmische Ansatz ebenfalls eine Funktionsanalyse anhand der aktuellen Systemkonfiguration, wie z.B. Ventilstellungen, Defekte etc. In allen bisher aufgeführten Darstellungs-Ansätzen gibt es eindeutige Bezeichnungen für Unterelemente, wie Bauteilbezeichnungen oder Knoten-/Kantennamen. Als Grundvoraussetzungen für die Berechenbarkeit von gerichteten Graphen müssen vor allem die Repräsentationen eines Systems konsistent und alle wesentlichen Elemente vorhanden sein. D.h. alle funktionsrelevanten Elemente aus dem Funktionsplan müssen sowohl in der Logikdarstellung, als auch in dem physikalischen System auffindbar sein (siehe Bild 2).

Sollen graphen-theoretische Algorithmen für die Simulation von hydraulischen Systemen genutzt werden, so sind die folgenden drei Grundelemente zu wählen.

- **Knoten**: Jede Art von Armatur, Verzweigung, Pumpe und ähnlichem wird als Knoten im Graph erfasst.
- **Kanten**: Rohrleitungen und Ventile werden mit Kanten abgebildet. Für eine bessere Übersicht in der grafischen Darstellung sollten beide unterscheidbar ausgewiesen werden.
- **Knoten- oder Kantenkosten**: Bieten aber Potential für die Modellierung von weiteren strömungsmechanischen Kenngrößen, was hier noch nicht umgesetzt wird.

4.2 Verknüpfung von Realität, Virtualität und Logik

In den bisherigen Ansätzen wurde die eindeutige Identifikation von Bauteilen vor allem für die Verknüpfung mit Informationen, Anweisungen und Animationen genutzt.

Der zuletzt gezeigte Zusammenhang zwischen Modell (Funktionsplan), Logikdarstellung (gerichteter Graph) und physikalischem System soll hier konkretisiert werden (siehe Bild 2). Als Bespiel dient ein Ventil in einem komplexeren hydraulischen System. Im Funktionsplan ist dieses Ventil mit einem Standardsymbol versehen und z.B. mit *Ventil 1* bezeichnet. Im gerichteten Graphen wird dieses Ventil durch eine Kante zwischen zwei Knoten, z.B. den *Knoten 1* und *Knoten 2* ausgedrückt. Besagtes *Ventil 1* lässt sich an dem hydraulischen System ebenfalls eindeutig durch seine Position identifizieren. Für das geplante Vorgehen sind besonders die logische Darstellung und die

Projektion in die Wirklichkeit relevant, so dass die Assoziation von physikalischem Bauteil zu dem korrespondierenden Element im Graph stattfinden muss. Die Verknüpfung dieser Elemente setzt natürlich digitale Repräsentation von sowohl dem Logik-Modell, als auch von dem physikalischen System voraus.

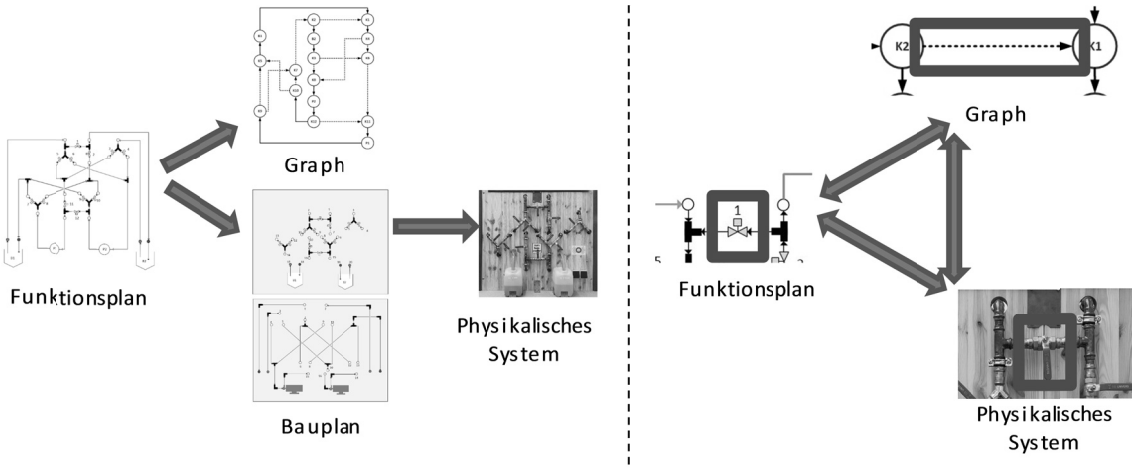


Bild 2: Zusammenhang zwischen Funktionsplan, Graph und realem System (links) und Beispiel einer Verknüpfung (rechts)

Für die AR-Unterstützung liegt, wie in [AN13] publiziert, ein digitales 3D-Modell vor, welches neben seiner Geometrierepräsentation zusätzlich durch eine Baugruppenstruktur beschrieben wird. Das Datenmodell der Baugruppenstruktur berücksichtigt neben den exakten Bezeichnungen der Bauteile und der Zuordnung zu Baugruppen auch weitere Metadaten, wie z.B. Versionsstände oder Freigabedaten. Dieses Datenmodell ist hierbei so flexibel, dass zusätzlich zu dem jeweiligen im System hinterlegten Graph, auch die Nummer oder der Name des spezifischen Elements (Kante oder Knoten) für jedes Bauteil referenziert werden kann. Mit den bekannten graphen-theoretischen Algorithmen der SPSP-Klasse können nun die kürzesten validen Pfade in dem hydraulischen System berechnet werden.

5 Prototypische Implementierung und Validierung

Als Anschauungsobjekt für die Validierung der vorgestellten Lösung dient das komplexe hydraulische System (siehe Bild 3), an dem auch die letzte Casestudy durchgeführt wurde. Das hydraulische System verfügt über zwei separat schaltbare Kreiselpumpen, zwei Vorratsbehälter mit Zu- und Ableitungen und insgesamt zwölf schaltbare Ventile. Die Konstruktion des hydraulischen Systems erlaubt die Förderung des Fluides sowohl von jeder der beiden Pumpen durch jeden der beiden Behälter, als auch durch beide Pumpen in Reihe. Der jeweilige Fluidkreislauf ist abhängig von der Kombination der jeweiligen Ventilstellungen.

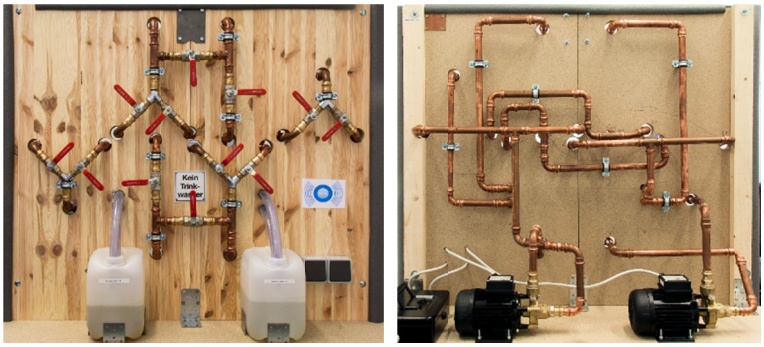


Bild 3: Vorderseite (links) und Rückseite (rechts) des hydraulischen Systems

Zur Konstruktion von Fehlerzuständen verfügt das System über zwei Sabotage-Schalter, um die beiden Pumpen vom Strom zu trennen, wodurch die eigentlichen Schalter ihre Funktion verlieren. Weiterhin können die Stellhebel der Ventile leicht demontiert werden, um einen Defekt in offener oder geschlossener Stellung zu simulieren.

Im hier betrachteten Anwendungsfall wird der Ausfall der primären Pumpe durch die Sabotage-Schaltung simuliert. Ziel ist es weiterhin Fluid durch Behälter 1 zu fördern. Die zugrunde gelegten Graphen, bzw. die algorithmische Analyse zur Lösung des beschriebenen Problems ist beispielhaft in Bild 4 dargestellt.

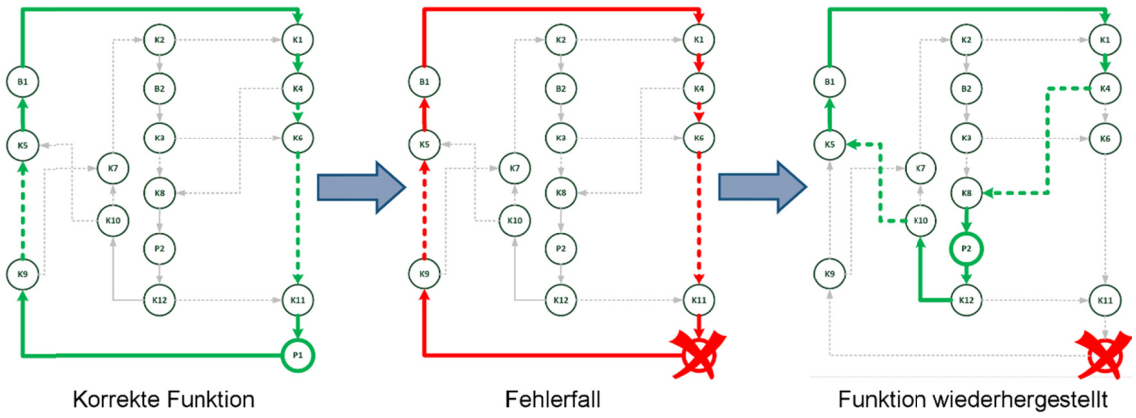


Bild 4: Logischer Ablauf vom Anwendungsfall „Ausfall Pumpe 1“

Das Anwendungsszenario wird gegenüber der in Kapitel 3 gezeigten Casestudy erweitert und die Funktionalitäten in einem Prototyp für das Apple iPad Air implementiert, die aktuelle Iteration des Projekts wurde jedoch auf Android portiert. Der Prototyp orientiert sich an der in [AN13] dargestellten Lösung, wird jedoch um Verarbeitung, Auswertung und Analyse der Graphen erweitert.

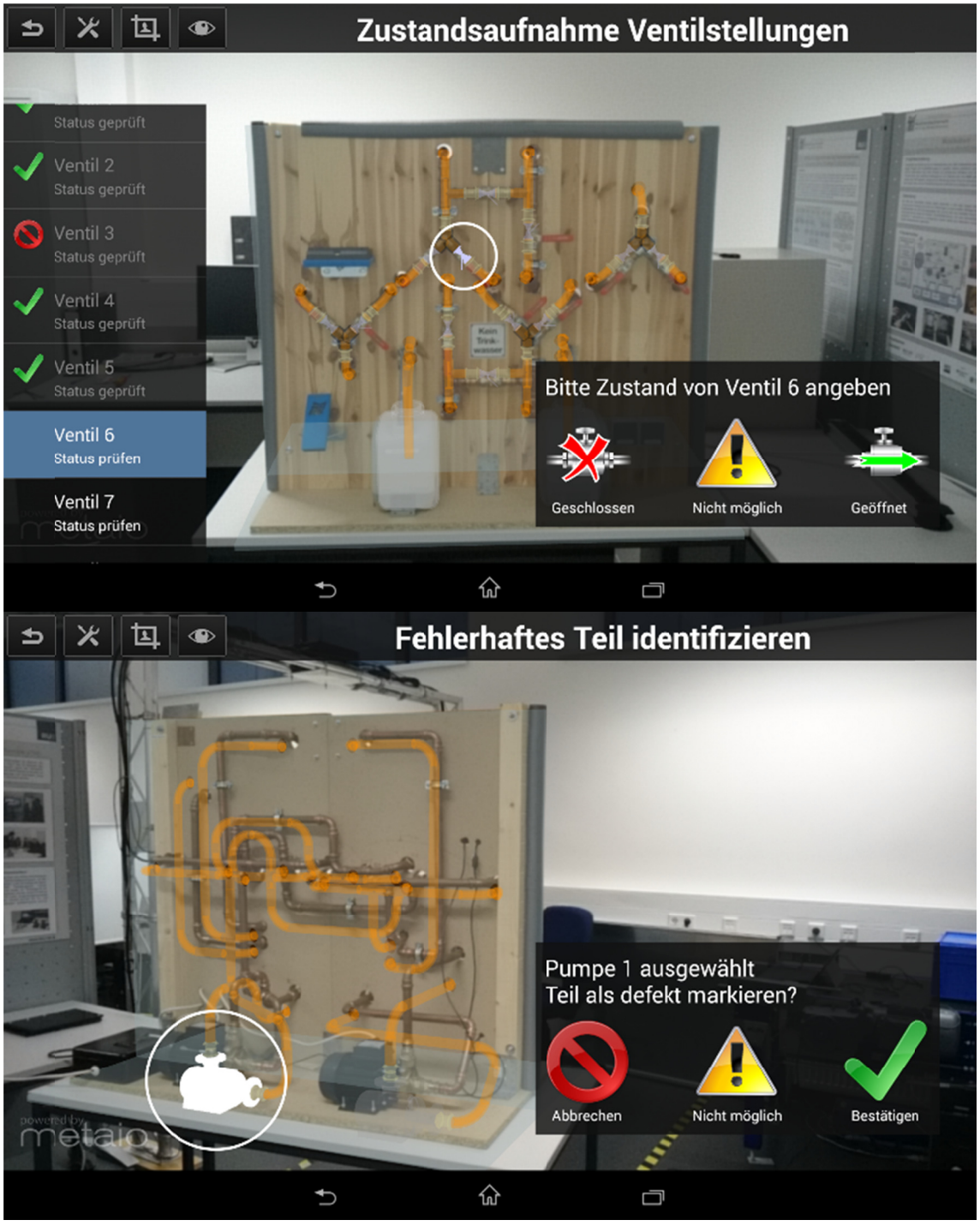


Bild 5: Aufnahme des Ist-Zustandes mittels AR (links) und Anzeige der zustandsorientierten AR-Arbeitsanweisungen (rechts)

Für die AR-Unterstützung wird das metaioSDK 6 von Metaio verwendet. Die Interaktion mit der AR-Überlagerung wird hierbei ebenfalls erweitert, sodass die Aufnahme des Ist-Zustandes über die Auswahl der einzelnen überlagerten Elemente über das Display aufgenommen werden kann (siehe Bild 5). Die algorithmische Analyse des Ist-Zustandes ermöglicht eine zustandsorientierte Generierung der Arbeitsanweisung und

somit eine hohes Maß an Instandhaltungsunterstützung für den Servicetechniker (siehe Bild 5).

Die detaillierte Aufnahme des Ist-Zustandes wird hierbei nicht nur für die eigentliche Analyse und Problemlösung verwendet, sondern fließt ebenfalls als Dokumentation in das in [AN13] dargestellte Konzept der Serviceberichte ein.

6 Resümee und Ausblick

Heutige Technologien, sowohl im Bereich der Hardware bei mobilen Endgeräten, wie Tablets oder Smartphones, als auch die Entwicklungen im Bereich der Augmented Reality, bieten enorme Potentiale für eine Unterstützung der Instandhaltung. Die heutigen Ansätze berücksichtigen zwar den Einsatz von AR bei der Visualisierung von Informationen, beziehen sich aber auf festgelegte Ablaufpläne. Es existieren jedoch Möglichkeiten die Funktion einer Maschine oder Anlage algorithmisch zu analysieren und so zustandsorientierte Problemlösungen bereitzustellen. Hier am Beispiel eines komplexen Rohrleitungssystems dargestellt, kann die Abbildung als Graph eine solche automatische Problemlösung bieten. Hierbei wird AR als intuitives Werkzeug zur Aufnahme des Ist-Zustandes eingesetzt, indem eine Verknüpfung von Funktion, Logik, Realität und des virtuellen 3D-Modells erzeugt wird. Die aus den Vorarbeiten entstandenen Ergebnisse bilden hierbei die Basis für eine zustandsorientierte AR-gestützte Instandhaltung. Innerhalb des Gesamtkonzepts fokussiert der Beitrag zunächst die algorithmische Analyse des Ist-Zustandes und die Verknüpfung der Logikelemente. Der entwickelte Prototyp wird hierbei anhand eines realen Rohrleitungssystems als Demonstrator validiert.

Da hier lediglich ein Teilaspekt der Gesamtlösung präsentiert wurde, wird die Generierung der modularen AR-Arbeitsanweisung auf Basis der Analyseergebnisse ein Schwerpunkt der Folgeaktivitäten sein. Weitere Teilaspekte werden die automatische Generierung der dafür notwendigen Graphen aus den virtuellen Modellen. Für eine Steigerung der Zugriffssicherheit wird darüber hinaus zukünftig eine kryptografisch sichere Methode eingesetzt. Daher werden RFID-Tags die heutigen QR-Codes oder Barcodes ersetzen.

Derzeit wird eine Neuauflage der Casestudy von 2014 durch die Autoren vorbereitet, um den neuen Prototypen äquivalent durch eine Vielzahl von Testanwendern validieren und die neuen Funktionalitäten und Optimierungen hinsichtlich der gesteigerten Usability bewerten zu können.

Literatur

[AGN14] ABRAMOVICI, M.; GÖBEL, J.-C.; NEGES M.: Smart Engineering as enabler for the 4th Industrial Revolution. Berkeley, 6th International Conference on Integrated Systems Design and Technology, ISDT'14, USA, 26 March 2014.

[AKW14] ABRAMOVICI, M.; KREBS, A.; WOLF, M.: Approach to ubiquitous support for maintenance and repair jobs utilizing smart devices. In: Proceedings of the tenth International Symposium on Tools and Methods of Competitive Engineering, TMCE, , Budapest, Hungary, ISBN 978-9-461-86176-4, pp. 83-93, 2014, May 19-23 2014.

[AN14] ABRAMOVICI, M., NEGES M.: Durchgängige Lösung zur Unterstützung von Wartungsprozessen durch Augmented Reality mit Smart Devices. In: Tagungsband Entwerfen, Entwickeln, Erleben. Methoden und Werkzeuge in der Produktenentwicklung. EEE'2014. Dresden, Deutschland: TUDpress, pp. 199-212. 2014

[AN13] ABRAMOVICI, M.; NEGES M.: Potentiale der Anwendung von kommerziellen Smartphones und Tablets für Augmented Reality unterstützende Wartungsaufgaben. In: Tagungsband Augmented & Virtual Reality in der Produktentstehung. 11. Paderborner Workshop. Paderborn, Deutschland: HNI-Verlag, pp. 261-272, 2013.

[BKK+10] BERNHARD, J.; KÖNIG, A.; KRUPP, M.; PFLAUM, A. AND STRAUß; W.: Mobile Technologien im Vor-Ort-Service, Mobile Servicewelten im Maschinenbau, VDMA-Verlag, Frankfurt a. M., Deutschland, pp. 70-132, 2010

[COO13] COSKUN, V.; OK, K.; OZDENIZCI, B.: Professional NFC Application Development for Android. Wiley Verlag, UK, 2013.

[DIN 31051] DEUTSCHES INSTITUT FÜR NORMUNG E.V.: DIN 31051: Fundamentals of maintenance“, Beuth Verlag, Berlin, Deutschland, 2012.

[EOP07] EISENHAUER, M.; OPPERMANN, R.; PRINZ, W.: Internet der Dinge. Springer-Verlag, Heidelberg, Deutschland, ISBN 978-3-540-36729-1, 2007.

[FF05] FLEISCH, E.; FRIEDEMANN, M.: Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis: Visionen, Technologien, Anwendungen, Handlungsanleitungen. Springer-Verlag, Berlin, Deutschland, 2005.

[Fin02] FINKENZELLER, K.: RFID-Handbuch: Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten. Hanser Verlag, München, Deutschland, 2002.

[HWT13] HOU, L.; WANG, X.; TRUIJENS, M.: Using Augmented Reality to Facilitate Piping Assembly: An Experiment-Based Evaluation. Journal of Computing in Civil Engineering, 2013.

[LT06] LEGNER, C.; THIESSE, F.: RFID-based maintenance at Frankfurt airport, Pervasive Computing, IEEE, Volume:5 , Issue: 1, pp. 34-39, ISSN: 1536-1268, 2006

[MRP+10] MANZINI, R.; REGATTIERI, A.; PHAM, H.; FERRARI, E.: Maintenance Information System and Failure Rate Prediction. Maintenance for Industrial Systems, Reliability Engineering, Pham, Vol. 2, Springer-Verlag, London, UK, pp. 189-217, 2010.

[SG13] SCHULZ,D.; GITZEL, R.: Seamless Maintenance – Integration of FDI Device Management & CMMS. Proceedings of IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), ISBN: 978-1-4799-0862-2, 2013.

[SJM14] SCHLICK, C. M.; JESKE, T.; MÜTZE-NIEWÖHNER, S.: Unterstützung von Lernprozessen bei Montageaufgaben. Flexible Produktionskapazität innovativ managen, Springer-Verlag Berlin, Deutschland, ISBN 978-3-642-39895-7, pp. 163, 2014.

[vKP10] VAN KREVELEN, D.; W.; R. POELMAN: A survey of augmented reality technologies, applications and limitations. International Journal of Virtual Reality, pp. 1-19, 2010.

[VDI 2860] VDI-RICHTLINIE 2860: Montage- und Handhabungstechnik - Handhabungsfunktionen, Handhabungseinrichtungen, Begriffe, Definitionen, Symbole. Beuth Verlag, Berlin, 1990

[WF07] WOHLGEMUTH, W.; FRIEDRICH, W.: Advanced Augmented Reality Technologies for Industrial Service Applications: ARTESAS. BMBF-Leitprojekt ARTESAS, zusammenfassender Schlussbericht: Siemens AG, 2007.

Autoren

Prof. Dr.-Ing. Michael Abramovici promovierte an der TU Berlin bei Prof. Spur und ist seit 1994 Professor für Maschinenbauinformatik an der Ruhr-Universität Bochum. Er ist Direktor des Instituts für Unternehmensführung (ifu) der Ruhr-Universität Bochum sowie Gastprofessor an der Tongji-Universität in Shanghai, China. Er ist Vorstandsvorsitzender der Wissenschaftlichen Gesellschaft für Produktentwicklung – WiGeP. Seit 2009 ist Prof. Abramovici Mitglied der Deutschen Akademie für Technikwissenschaften (acatech).

Dr.-Ing. Matthias Neges ist seit 2008 wissenschaftlicher Mitarbeit am Lehrstuhl für Maschinenbauinformatik der Ruhr-Universität Bochum. Sein Forschungsgebiet liegt im Bereich der virtuellen Produktenwicklung, mit Fokus auf die Virtual & Augmented Reality. Er promovierte im Oktober 2014 zum Thema „Smart Augmented Reality für die Indoor-Navigation und Instandhaltung“ und ist für die Gruppe „Virtual Engineering“ am Lehrstuhl verantwortlich.

Mario Wolf, M. Sc. ist seit 2012 wissenschaftlicher Mitarbeit am Lehrstuhl für Maschinenbauinformatik der Ruhr-Universität Bochum. Sein Forschungsgebiet liegt im Bereich der industriellen Nutzung von Smart Devices zur Instandhaltungsunterstützung, mit besonderem Fokus auf kryptografisch sicheren Anwendungen. Er ist für das lehrstuhleigene Smart Product Labor und die Arbeit im Kompetenzzentrum für hydraulische Strömungsmaschinen verantwortlich.

.

Mobile Assistenzsysteme für sicheren Betrieb und Wartung von Maschinen und Anlagen

Dr.-Ing. Simon Adler, Alexa Kernchen, Tilman Reipsch, Eric Bayrhammer
Prof. Dr. sc. techn. Ulrich Schmucker
Fraunhofer-Institut für Fabrikbetrieb und -automatisierung IFF
Sandtorstr. 22, 39106 Magdeburg
Tel. +49 (0) 391 / 40 90 776
E-Mail: simon.adler@iff.fraunhofer.de

Zusammenfassung

Die Nutzung digitaler Maschinen- und Anlagenmodelle in der Betriebsphase ist ein wesentliches Element im Kontext von Industrie 4.0. Dabei sollen die digitalen Modelle direkt aus der Entwicklung, idealerweise in durchgängigen digitalen Entwurfssystemen, übernommen werden und in der Betriebsphase einer nutzbar sein.

Am Fraunhofer IFF wurde dazu das System V-ASSIST entwickelt, das Betriebszustände durch Vergleich mit dem parallel laufenden virtuellen Maschinenmodell überwacht und kontextbezogen analysiert. Im Fehlerfall wird der Bediener durch das mobile System (u.a. Tablet, HMDs) direkt zu dem fehlerhaften Bauteil navigiert und mit Zustandsdaten der Anlage ortsbezogen unterstützt. Vor Ort hat er Zugriff auf detaillierte Fehlerbeschreibungen inkl. möglicher Fehlerursachen sowie Maßnahmen zur Störungsbehebung.
V-ASSIST basiert auf dem am IFF entwickelten durchgängigen digitalen Entwurfssystem VINCENT (Virtual Numeric Control Environment), welches notwendige Informationen aus den Entwurfsdomänen der Mechanik, Elektrotechnik sowie Steuer- und Regelungstechnik verknüpft und konsistent hält. Dadurch wird ein weitgehend paralleler Entwurf einer Maschine oder Anlage ermöglicht. Entwicklungszeit und –kosten werden reduziert, der Entwurf wird weitestgehend abgesichert und die Steuerungssoftware kann entwickelt und getestet werden, bevor die Maschine oder Anlage gebaut wird.

V-ASSIST verlängert die Nutzungskette der digitalen Daten in die Betriebsphase und unterstützt effektiv die Arbeit von Bedienern und Servicetechnikern einer Maschine. Der Beitrag beschreibt neben dem Grundkonzept von VINCENT detailliert den Aufbau, die Eigenschaften, Einbindung, Nutzung sowie Erweiterungsmöglichkeiten von V-ASSIST. Anhand praktisch realisierter Beispiele werden die Nutzungsmöglichkeiten demonstriert.

Schlüsselworte

Mobile Assistenz, Virtual Engineering, Digital Engineering, virtuelle Inbetriebnahme

Mobile Assistance-System for safe operation and service of machines and plants

Abstract

The use of digital models of machines and plants during their operation is a fundamental element of Industry 4.0. The digital models are drawn directly from development, ideally in integrated digital systems, and can be used during the operation of machines and plants.

The Fraunhofer Institute IFF developed the V-ASSIST system, which monitors and contextually analyzes operating condition by comparing a real virtual model running in parallel. In the event of a malfunction, the mobile system (on, among other devices, a tablet or HMD) navigates the operator directly to the malfunctioning component and supports him or her with localized data on the plant's condition. The operator has access on site to detailed descriptions of the malfunction, including potential causes and corrective actions.

V-ASSIST is based on the integrated digital design system VINCENT (Virtual Numeric Control Environment) developed by the Fraunhofer Institute IFF, which links and keeps all data from the design domains of mechanics and electrical and control engineering consistent. This makes it possible to design a machine or plant largely in parallel. Development times and costs are reduced, the design is extensively validated and the control software can be developed and tested before the machine or plant is built.

V-ASSIST extends the chain of use of the digital data during operation and effectively supports operators and service technicians' work on a machine. This paper describes the basic concept of VINCENT as well as the design, features, integration, use and upgrade options of V-ASSIST in detail. Examples from the field demonstrate the potential uses.

Keywords

Mobile Assistance-Systems, Virtual Engineering, Digital Engineering, virtual Operations

1 Motivation

Bei der traditionellen Maschinen- und Anlagenentwicklung sind die beteiligten Disziplinen häufig noch sequentiell tätig. Dabei wirken sich der Informationsverlust durch fehlende Durchgängigkeit, die fehlende Rückkopplung zwischen den Disziplinen sowie der häufig auftretende Projektverzug bei Entwicklungsfehlern, negativ auf Entwicklungszeit, -kosten und -qualität aus [Sch14]. Für eine effizientere Entwicklung komplexer Anlagen ist daher ein Wandel vom sequentiellen Workflow zum durchgängigen digitalen Engineering erforderlich. Durch das digitale Engineering können Tätigkeiten parallelisiert, die Gesamtentwicklungszeit signifikant verkürzt und gewünschte Eigenschaften frühzeitig abgesichert werden. Die dabei erstellten Daten sind über die Anlagenentwicklung hinaus auch zur Unterstützung der Inbetriebnahme sowie beim Anlagenbetrieb nutzbar. Hierdurch wirkt sich das digitale Engineering positiv auf Kosten, Zeit und Qualität im gesamten Anlagenlebenszyklus aus [BKS+09].

Um digitales Engineering zu ermöglichen muss eine konsistente Datenhaltung und Integration vorhandener Entwurfswerkzeuge gewährleistet werden. Eine wesentliche Herausforderung ist hierbei der semantische Bezug zwischen den (Daten-)Sichten der unterschiedlichen Disziplinen.

Für die Unterstützung beim Anlagenbetrieb (u.a. Prüfung, Wartung oder Fehlerbehebung) müssen relevante Teilinformationen automatisiert zur Verfügung gestellt werden. Neben den anlagenspezifischen Daten, bietet perspektivisch die Anbindung und semantische Kopplung an ERP-Systeme der Betreiber weitere Potentiale.

2 Lösungsansatz

Im Folgenden wird der am Fraunhofer IFF entwickelte Ansatz zur Realisierung des digitalen Engineerings für die Entwicklung von Sondermaschinen und –anlagen beschrieben. Hierbei werden zunächst die beteiligten Komponenten erläutert und Aspekte der Datendurchgängigkeit im Entwicklungsprozess betrachtet. Nachdem die Ausweitung der digitalen Daten aus den Entwicklungsprozessen in den Anlagenlebenszyklus als V-ASSIST beschrieben wurde, wird ein Ausblick auf die weiteren Arbeiten und Untersuchungen gegeben.

2.1 Virtual Numeric Control Environment (VINCENT)

Durch das *Virtual Numeric Control Environment* (VINCENT) wird die Kooperation, Durchgängigkeit und Parallelisierung zwischen der Konstruktion und Steuerungsprogrammierung adressiert.

Die traditionelle Entwicklung im Maschinenbau sieht die sequentielle und getrennte Tätigkeit von Konstruktion, Elektrotechnik, Montage (Aufbau), Steuerungsprogrammierung und Inbetriebnahme vor [Sch14]. Die Funktionen und Mechanismen, die zu Be-

ginn vom Konstrukteur entwickelt werden, führen zu Vorgaben in der Steuerungsprogrammierung am Ende der Entwicklungskette [Böh14]. Das Abstraktionsniveau zwischen Konstruktion und Steuerungsprogrammierung unterscheidet sich jedoch signifikant und resultiert bisher in einem Bruch der Durchgängigkeit. Wesentlich sind die Definition der Bewegungsabläufe und funktionalen Zusammenhänge der Maschinenkomponenten, da aus diesen die Steuerungsprogrammierung abgeleitet wird.

Durch VINCENT werden Probleme durch die Abhängigkeit zwischen Konstruktion und Steuerungsprogrammierer aufgehoben, indem die Programmierung bereits mit der virtuellen Anlage erfolgt, die mit einer realen Steuerung gekoppelt ist. Im Gegensatz zur traditionellen Vorgehensweise kann der Steuerungsprogrammierer frühzeitig Probleme, die bei der virtuellen Inbetriebnahme auftreten, an den Konstrukteur zurückmelden, was zu hohen Einsparungen an Entwicklungszeit und –kosten führt.

2.2 Virtuelle kooperative Steuerungsprogrammierung

Um die parallele Arbeit und Kooperation der Disziplinen auf eine neue Ebene zu heben, wurden in Anlehnung an das Teach-In-Verfahren bei Robotern, Methoden zum Virtuellen Teachen von Sondermaschinen entwickelt und in VINCENT umgesetzt [Böh14]. Dabei werden die Bewegungen der Maschine am virtuellen Modell vorgeführt, aufgezeichnet und in einen Steuerungscode überführt.

Dafür werden zunächst die CAD-Daten der Anlagenkomponenten vom Konstrukteur in VINCENT importiert, entsprechend der Kinematik der Anlage gruppiert und hierarchisch in Beziehung gesetzt. Zu jeder Gruppe definiert er Bewegungsachsen (Translation, Rotation) sowie entsprechende Freiheitsgrade. Auf Grundlage seines Wissens über die Funktionsweise der Anlage bewegt der Konstrukteur interaktiv nacheinander die entsprechenden Bewegungsachsen (siehe Bild 1a) und legt so schrittweise den Bewegungsablauf der Anlage interaktiv als Untersequenzen (siehe Bild 1b) fest.

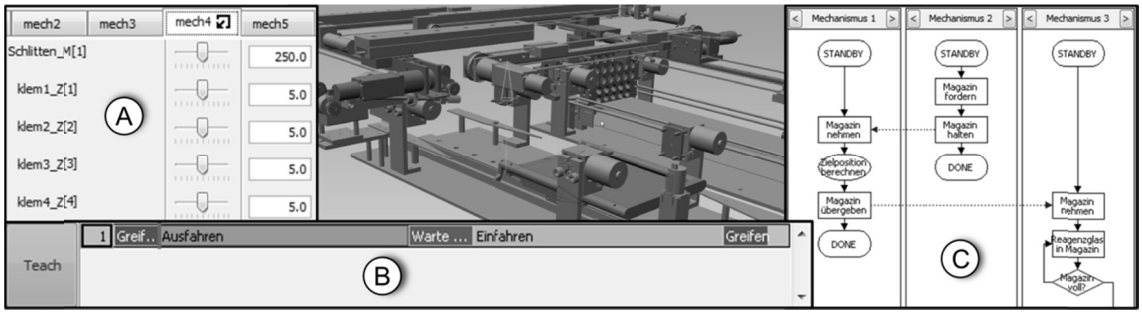


Bild 1: a) Übersicht über das Interface zur kooperativen Steuerungsprogrammierung. Bewegung der Objekte entlang der Achsen, b) Untersequenzen, c) strukturierte Hauptsequenzen mit Abhängigkeiten

Der Steuerungsprogrammierer bekommt anhand dieser Untersequenzen zunächst einen ersten Eindruck von der Anlagenfunktion. Er optimiert die Untersequenzen aus Sicht

der Steuerungsprogrammierung und erstellt Hauptsequenzen, die diese Untersequenzen in einen logischen Gesamtablauf für die Maschine bringen (siehe Bild 1c) Die Arbeit an den Untersequenzen erfolgt stets in Abstimmung zwischen Konstrukteur und Programmierer, wodurch eine direkte Kommunikation mithilfe des virtuellen Modells entsteht [KB13]. VINCENT generiert den Steuerungscode als Programmgerüst für die Zielhardware, der durch den Steuerungsprogrammierer erweitert (Bedienhandlungen, Fehlerreaktionen etc.) wird.

Bei dieser kooperativen Steuerungsprogrammierung wird bereits eine Echtzeitkollisionsprüfung durchgeführt, um mögliche Fehlerquellen beim Anlagenbetrieb frühzeitig zu identifizieren. Für die weitergehende Steuerungsentwicklung und das Testen des Steuerungsprogramms wird VINCENT direkt mit realen Steuerungen gekoppelt. Die Steuerungsbefehle werden dann von der virtuellen Anlage umgesetzt. Dies bietet unter anderem auch die Möglichkeit der Inbetriebnahme von Teilanlagen, indem bereits real aufgebaute Anlagenelemente mit bisher nur virtuell verfügbaren Komponenten gemeinsam betrieben werden.

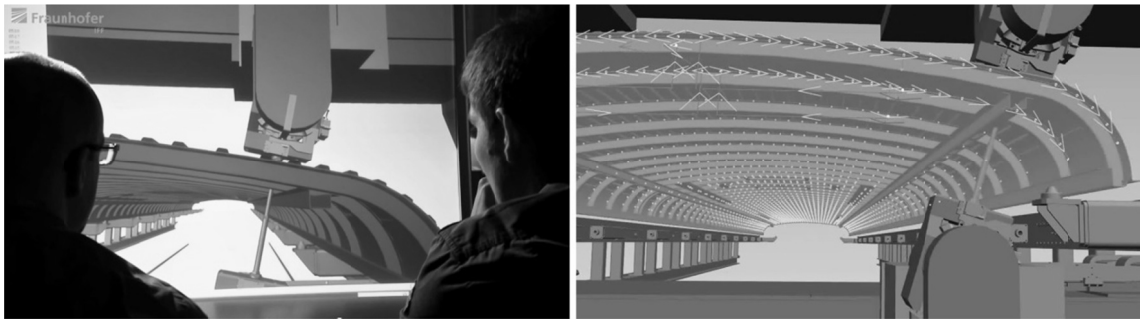


Bild 2: VINCENT für das virtuelle Punktschweißen. Kollisionen zwischen Anlage und Werkstück sind durch Tests am virtuellen Modell direkt sichtbar (rechts).

VINCENT wird zum Beispiel als Offline-Programmiersystem für die virtuelle Planung von Schweißpunkten bei der Firma SMC Calvörde genutzt[1] (siehe Bild 2). Hierbei wird das NC-Programm automatisch, auf Basis der Eingabe der Schweißpunkte generiert. Durch Simulation der Bewegung kann die potentielle Gefahr von Kollisionen zwischen den Robotern oder mit dem Werkstück bereits in der virtuellen Planung ausgeschlossen werden.

2.3 Datenintegration

Die Grundlage für das digitale Engineering sowie die Unterstützung bildet die Datenintegration aus verschiedenen Domänen zu einer gemeinsamen durchgängigen Datenbasis. Die Sichten der unterschiedlichen Domänen müssen dabei erhalten bleiben.

[1] http://www.smcalvoerde.de/index.php?option=com_content&view=article&id=167&Itemid=1227

Ein Beispiel für gemeinsame Daten unterschiedlicher Domänen ist die Festlegung der Untersequenzen in VINCENT (Abs. 2.2). Diese Angaben können sowohl durch den Konstrukteur als auch aus steuerungstechnischer Sicht erfolgen. Ein anderes Beispiel ist die Kopplung der Anlagenplanung mit den Daten aus dem Anlagenbau. Wenn die Anlage aufgebaut wird, werden auch Anlagenkomponenten von Drittanbietern genutzt, zu denen entsprechende Dokumentationen vorliegen. Diese Komponentendaten müssen, für die digitale Unterstützung in der Betriebsphase, mit den Konstruktionsdaten assoziiert werden.

Für diese Arten der Datenintegration wurde der *Engineering Modeling Linker* (EMELI) als Datenmanagementsystem entwickelt. Zudem bietet EMELI die Schnittstelle für den Live-Austausch von Daten verschiedener Zielanwendungen und –domänen. Dieser Zugriff erfolgt dynamisch über ein COM-Interface (Component Object Model) und bildet die Grundlage für die Assistenz zur Fehlerbehebung mittels V-ASSIST. Dem Bearbeiter eines Elektroplans wird es dadurch ermöglicht über die Bauteilbeschreibung im E-Plan direkt auf die 3D-Darstellung im Anlagenzusammenhang in VINCENT zuzugreifen.

2.4 V-ASSIST

Bisher stellt der dargestellte Lösungsansatz zur Realisierung des digitalen Engineerings einen Workflow ausgehend von der Konstruktion zur Steuerungsentwicklung dar, wobei eine domänenübergreifende Basis durch die Verknüpfung heterogener Datenbasen über EMELI hergestellt wird. Durch V-ASSIST wird die Durchgängigkeit der digitalen Daten bis in die Anlagenbetriebsphase realisiert.

2.4.1 Anforderungen

Die Zielstellung von V-ASSIST ist die Assistenz des Anwenders bei Wartung, Prüfung, Inbetriebnahme und Betrieb von Anlagen durch mobile Endgeräte. Mobile Endgeräte wie Smartphones, Tablets oder Head-Mounted-Displays (HMD) sind leistungsstark und bieten neben einem festen Verbund aus Bildschirm und Kamera verschiedene Sensoren. Durch die drahtlose Datenübertragung ermöglichen sie den direkten Zugriff auf dezentrale Datenbanken. V-ASSIST ist die Methodik zur Assistenz anlagenspezifischer Tätigkeiten mittels durchgängiger Daten des virtuellen Engineerings. Im Folgenden wird die V-ASSIST-Softwareanwendung der mobilen Geräte vereinfacht als V-ASSIST bezeichnet.

Insbesondere beim Betrieb mehrerer Sondermaschinen verschiedener Hersteller (Anlagenfragmentierung) stellen diese Tätigkeiten häufig eine Herausforderung dar. Bei fehlerhafter Bedienung, Wartung oder Betrieb droht im Fehlerfall ein zusätzlicher Schaden an der Anlage, aus dem zusätzliche Kosten resultieren.

Wenn ein Fehlerzustand eintritt wird über die Steuerung ein Fehlercode zur Verfügung gestellt. Dieser Code beschreibt den Fehler und ggf. die Fehlermeldende Komponente,

nicht aber dessen Ursache. Um die Ursache zu ermitteln muss der Anlagenzustand detaillierter untersucht werden. In klassischen Systemen gehen die funktionalen Zusammenhänge lediglich aus der Anlagendokumentation hervor. Beispielsweise bei elektrischen Problemen ist daher zusätzlich der E-Plan zu konsultieren, der selbst als digitales Dokument langwieriges Suchen und Nachverfolgen von Verbindungen erfordert. Insbesondere bei älteren Anlagen sind diese Dokumentationen nur bedingt verfügbar.

V-ASSIST ermöglicht den in situ Zugriff auf relevante Informationen, unterstützt die Suche nach Fehlerquellen und gibt Lösungsstrategien zur Fehlerbehebungen an. Neben anlagespezifischen Daten sind die Informationen zur Umgebung potentiell relevant. Einige Tätigkeiten erfordern beispielsweise Personal mit speziellen Qualifikationen (u.a. Ex-Zonen), spezielle Ressourcen (z.B. Kran), Betriebsmittel oder Ersatzteile. Die semantische Verbindung dieser Daten ist aktueller Forschungsgegenstand und wird abschließend als Ausblick beschrieben.

2.4.2 Assistenz beim Betrieb mobiler Sondermaschinen

Die Inbetriebnahme von Sondermaschinen erfordert neben der korrekten Bedienung auch häufig eine korrekte Vorbereitung von Abhängigkeiten in der Umgebung. Insbesondere bei Maschinen zur Materialverarbeitung sind die Bedienung und korrekte Vorbereitung ausschlaggebend für die geforderten Genauigkeiten.

Bei mobilen Sondermaschinen müssen deren Montage, Einmessung, Inbetriebnahme und Betrieb bei jedem Einsatz in gleichbleibender Qualität garantiert und häufig durch Messsysteme geprüft und für den Auftraggeber dokumentiert werden. Jede Situation vor Ort kann zu umgebungsspezifischen Anpassungen führen die nicht nur dokumentiert werden müssen, sondern auch für die weitere Anlagenentwicklung relevant sein können.

V-ASSIST bietet entsprechend ein Modul zur Assistenz bei der Inbetriebnahme. Dieses Modul wurde in Zusammenarbeit mit der VAKOMA GmbH in Magdeburg realisiert [Adl14], um den internationalen Einsatz einer mobilen Sondermaschine zur Großgetriebehabilitierung zu unterstützten. Dieser Vorgang umfasst u.a. die Korrektur der Getriebelager durch Nachfräsen, wobei die Drehmaschine durch Messungen mit einem Lasertracker eingemessen wird. Experten können über ein Autorensystem die hierbei einzuhaltende Reihenfolge der Arbeitsprozesse als Arbeitspakte, Untergruppen und Einzelaufgaben festlegen. Abhängigkeiten zwischen Aufgaben können frei definiert werden. Die textuelle Beschreibung der Arbeitsaufgabe kann durch Medieninhalte (Bilder, Video, VR/AR) sowie durch beliebige weitere Dokumente von Drittanbietern angereichert werden. Diese Beschreibungen dienen zunächst zur Vorbereitung und haben besonders Relevanz für den Einsatz mobiler Sondermaschinen mit wechselnden Teams.



Bild 3: V-ASSIST Modul zur Montageunterstützung mit VR-Ansicht der Arbeitsschritte sowie Checkliste mit Zusatzdokumenten

Aus Arbeitsaufgaben und Abhängigkeiten wird zusätzlich eine Art Checkliste erzeugt, bei dem jeder Vorgang quittiert werden muss (siehe Bild 3). Erst wenn eine Tätigkeit abgeschlossen ist, können Folgetätigkeiten bearbeitet werden. Zu jeder Tätigkeit kann im Autorensystem zusätzlich festgelegt werden, ob Eingaben (u.a. Messergebnisse) abgefragt werden sollen. Der Autor legt zur Plausibilisierung Gleichungen fest, mit denen die Eingaben, auch zwischen verschiedenen Tätigkeiten, automatisch validiert werden. In dem konkreten Fall der Vakoma GmbH, werden Abweichungen gegenüber Erwartungswerten als neues Know-How und für die Weiterentwicklung der Sondermaschine genutzt.

Der Anwender ergänzt Bilder, Videos oder Texteingaben. Die Bearbeitung der Aufgaben, erfasste Kenngrößen, Ergebnisse der Validierung sowie zusätzliche Angaben durch den Anwender bilden dann die Basis für die Generierung der Dokumentation. Durch die Abhängigkeit der Arbeitsaufgaben voneinander wird die Einhaltung der Reihenfolgen, als wesentlicher Aspekt der Qualitätssicherung, gewährleistet. Durch die konsequente Erfassung und automatische Validierung von Messwerten, werden insbesondere Anpassungen bei unerwarteten Umgebungsbedingungen identifiziert und erfasst.

2.4.3 Assistierte Fehlerlokalisation

Wenn ein Fehler an der Anlage auftritt, dann wird dies durch einen Fehlercode beschrieben und bildet die Ausgangsbasis für die Fehlersuche. Allerdings ist bereits bei mittelgroßen Anlagen (ca. 20 bewegte Achsen) eine direkte Identifikation eines Fehlers oder einer Komponente und deren Lokalisierung nur schwer möglich [BAK+14]. Die

Interpretation von Fehlermeldungen zum Auffinden möglicher fehlerhafter Komponenten, erfordert das Studieren der Dokumentationen.

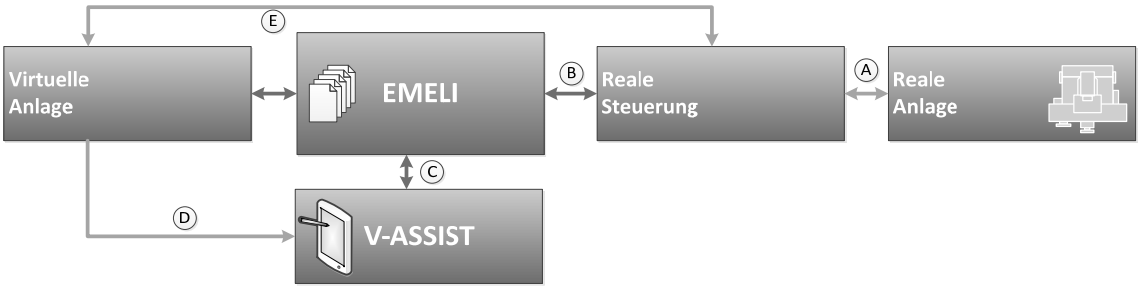


Bild 4: Integrationsschema zur Nutzung des digitalen Engineering in der Betriebsphase.

Daher besteht beim Anlagenbetrieb eine Live-Kopplung von der realen Steuerung mit der virtuellen Anlage (siehe Bild 4 E). Wenn die reale Anlage einen Fehlerzustand erreicht wird ein Fehlercode in der Anlagensteuerung generiert (Bild1a) und an EMELI übertragen (Bild 4 B). EMELI ermittelt die zu diesem Fehlercode zugeordneten Komponenten und meldet diese EMELI-spezifischen Identifikatoren an das gekoppelte V-ASSIST-System (Bild 4 C). Der Anwender wird nun auf seinem mobilen Gerät über diesen Fehler informiert und erhält Informationszugriff auf die Anlage, den Fehlercode sowie erste Informationen zur Fehlerart.

Wenn der Anwender vor Ort ist, wird er durch Augmented Reality (AR) bei der Fehlerlokalisation unterstützt. Er betrachtet hierzu die Anlage durch die Kamera seines mobilen Endgerätes und bekommt die Fehlermeldung direkt ortsbezogen an der entsprechenden Anlagenkomponente überlagert dargestellt. Hierfür ist die einmalige Registrierung zwischen der realen Anlage und dem virtuellen Modell erforderlich. Eine Möglichkeit ist die Anbringung von ID-Markern an der Anlage [WLS08]. Diese QR-Code ähnlichen künstlichen Marker können durch Verfahren der Bildverarbeitung robust im Kamerabildern erkannt werden. Durch ihre Struktur und bekannte Größe wird aus ihnen die relative Kamerapose (Position und Orientierung) des mobilen Gerätes bestimmt. Über einen kalibrierten Markerverbund wird bei Bedarf eine größere räumliche Abdeckung erreicht. Diese Kalibrierung erfolgt durch eine Videoaufnahme, nach Anbringung der Marker, automatisiert. Ergänzend wird für V-ASSIST ein Mono-SLAM-Tracking verwendet [KM09]. Hierbei wird aus der Bewegungsparalaxe eine Tiefenkarte (Mapping) und die Kamerapose (Localization) bestimmt. Bei SLAM wird eine große Datenmenge erzeugt und verarbeitet. Aufgrund der Ressourcenlimitierungen mobiler Geräte, kann die Kamerapose damit nur in einem begrenzten Arbeitsbereich kontinuierlich bestimmt werden. Indem der Bereich zwischen ID-Markern mittels SLAM überbrückt wird, wird der Bewegungsbereich vergrößert und die Markeranzahl reduziert.



Bild 5: Überlagerung eines Sensorfehlers an einer beweglichen Anlagenkomponente liefert sofort Informationen über Fehlerart und -ort.

Für die Fehlerlokalisation erfolgt eine Anfrage durch V-ASSIST an EMELI nach den Identifikatoren der CAD-Modelle von den zum Fehlercode gehörenden Komponenten. Über diese Identifikatoren wird anschließend bei VINCENT das Grenzvolumen (Axis-Aligned Bounding Box (AABB)) der virtuellen 3D-Modelle angefordert. Die Grenzvolumen dieser Objekte werden nun kontinuierlich von VINCENT an V-ASSIST übertragen bis der Fehler behoben wurde (Bild 4 D). Durch die direkte Kopplung von VINCENT zur realen Steuerung (Bild 4 E) ist das virtuelle Modell mit der realen Anlage stets synchron.

Durch die Kamera des mobilen Systems und durch das Tracking werden Überlagerungen virtueller Inhalte lagesynchron in das Kamerabild der realen Anlage eingeblendet. In V-ASSIST wird im Zentrum des Grenzvolumens das Symbol für die Fehlerart am Fehlerort eingeblendet und aktualisiert (Bild 5), auch bei beweglichen Komponenten [GBT+12]. Nach Lokalisierung des Fehlers erfolgt die assistierte Fehlerbehebung.

2.4.4 Assistierte Fehlerbehebung

Durch die assistierte Fehlerlokalisation (Abs. 2.4.3) unterstützt V-ASSIST die Fehlersuche. Die assistierte Fehlerbehebung unterstützt den Experten bei der Behebung des Fehlers in der Dokumentensuche, durch Aufzeigen von Handlungsempfehlungen sowie durch das Hinzuziehen eines Remote-Experten.

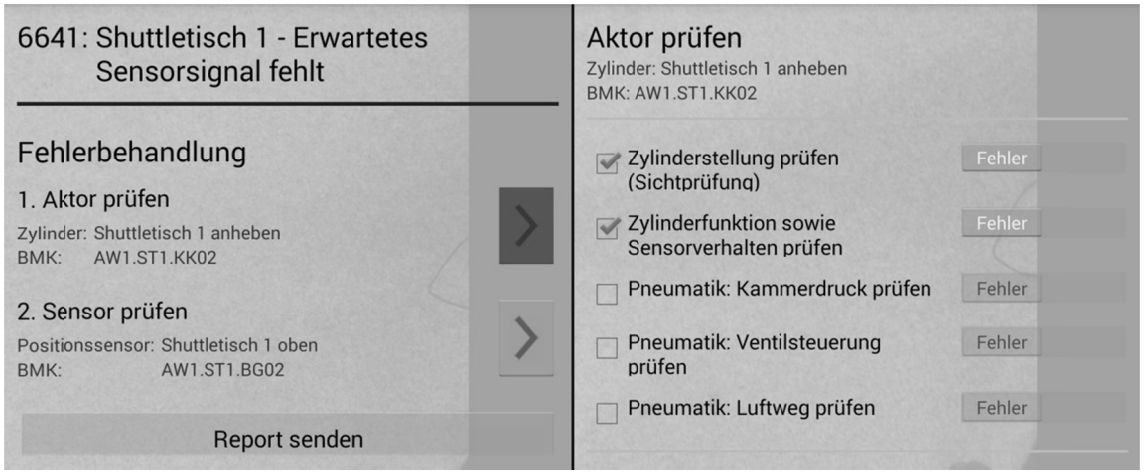


Bild 6: Seitenansicht im Live-Kamerabild des mobilen Gerätes zeigt mögliche Fehlerursachen an der Komponente (links) sowie Schritte zur Fehlerbehebung (rechts).

Beim Auftreten eines Fehlers werden die Komponenten, die durch den Fehlercode einbezogen werden, an V-ASSIST gemeldet. Aus einer Wartungsdatenbank, die an EMELI gekoppelt ist, werden entsprechend verknüpfte (Wartungs-)Dokumente und Problemlösungsstrategien bereitgestellt (siehe Bild 6) [BAK+14]. Damit werden mögliche Fehlerursachen (z.B. Aktor- oder Sensorfehler) sowie Schritte zur Fehlerbehebung beschrieben. Diese Schritte können abgearbeitet werden, wobei angegeben wird, ob im jeweiligen Schritt ein Fehler vorlag. Das System kann hierdurch adaptiv reagieren und in seinen Vorschlägen die Auftrittshäufigkeit in der Rangfolge der priorisierten Lösungsstrategien einbeziehen.



Bild 7: Der Remote Experte bekommt das Kamerabild des Anwenders übertragen und kann über grafische Annotationen bei der Fehlerbehebung anleiten.

Wenn ein Fehler nicht durch den Betreiber behoben werden kann, ist eine Beauftragung des Anlagenherstellers erforderlich. Wenn der Fehler nicht genau spezifiziert werden kann, dann treten nicht nur durch die Anreise, Ersatzteilbestellung und Reparatur hohe Kosten auf, sondern vielmehr durch die damit verbundenen Stillstandszeiten. In V-ASSIST ist daher eine Videokonferenz mit einem Remote-Experten vorgesehen. Er bekommt das Videobild vom mobilen Endgerät übertragen und kann seine Erläuterun-

gen zusätzlich durch grafische Annotationen unterstützen (siehe Bild 7). Über das Gespräch mit dem Service-Mitarbeiter kann er damit das korrekte Vorgehen im Kamerabild zeigen. Zusätzlich kann er seinen Bildschirminhalt übertragen, um Erklärungen anhand schematischer Darstellungen vorzunehmen. Ein ähnliches System wurde von [GNT+14] entwickelt und in einer Nutzerstudie mit positiven Ergebnissen getestet. Es ist geplant, die durch das Tracking ermittelte Pose des mobilen Endgerätes optional an das VINCENT zu übertragen, so dass in der CAD-Darstellung dieselbe Sicht wie in der vor Ort Situation eingenommen wird.

In aktuellen Arbeiten werden Methoden entwickelt, um die assistierte Fehlersuche weiter zu unterstützen. Hierbei soll die intelligente Dokumentennavigation anhand funktionaler Nachbarn realisiert werden. Des Weiteren besteht Forschungsbedarf in der Klassifikation von Baukomponenten zur Bestimmung von Ähnlichkeitsmaßen. Wenn an einer Baukomponente häufiger Fehler auftreten wäre, bei ähnlichen Komponenten sowie bei funktionalen oder räumlichen Bezügen, ein Querverweis zu dieser Komponente möglich. Des Weiteren soll über ein entsprechendes Klassifikationsschema eine abgeleitete Fehlerbehandlungsroutine ermöglicht werden, so dass der Autorenaufwand zur Beschreibung von Problemlösungsstrategien reduziert wird.

3 Resümee und Ausblick

V-ASSIST ist ein Lösungskonzept für den sicheren Betrieb und die Wartung von Maschinen und Anlagen. Es basiert auf der konsequenten Nutzung von Daten aus den Phasen der Anlagenkonstruktion und nutzt diese von der Montage bis in den Betrieb. Hierdurch werden Zeit und Kosten, durch die Unterstützung der Qualitätssicherung und durch Verkürzung von Kommunikationswegen, reduziert.

Aktuell erfordert V-ASSIST eine entsprechende Datenbasis, um insbesondere Assistenzfunktionen mit direktem Bezug zum aktuellen Anlagenzustand bereitzustellen, da dies eine (lesende) Kopplung mit der Anlagensteuerung erfordert. Module der Montageunterstützung und für den Remote-Experten sind jedoch autark einsetzbar.

Insbesondere die Anbindung an Bestandsanlagen, zu denen keine oder wenig Daten aus dem Betrieb vorgelagerten Phasen bestehen, stellt weiterhin eine Herausforderung dar. Auch die Anbindung an komplexe firmenspezifische ERP-Systeme erfordert kundenspezifische Integrationen. Es sollen fortführend Methoden entwickelt werden, um diesen Integrationsaufwand zu reduzieren und eine dynamische Kopplung verfügbarer Datenbasen zu ermöglichen. V-ASSIST kann hierdurch perspektivisch flexibel an die Bedürfnisse und die Dateninfrastrukturen von Anlagenbetreibern adaptiert werden.

Literatur

[Adl14] ADLER, SIMON: Schritt für Schritt geleitet. In: *IFFocus* (2014), Nr. 1/2014

[BAK+14] BAYRHAMMER, E.; ADLER, S.; KENNEL, M.; KROSSING, G.: Integration moderner HMI-Lösungen in den Alltag des Maschinenbaus. In: *17. IFF-Wissenschaftstage 2014, Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme*. Magdeburg, 2014. ISBN 2196-7598, S. 55-60

[BKS+09] BÖHME, T.; KENNEL, M.; SCHUMANN, M.; WINGE, A.: Automatisierte Erstellung domänenübergreifender Modelle und echtzeitfähige Kopplung von Simulation, Visualisierung und realen Steuerungen. In: *HNI-Verlagsschriftenreihe*. Bd. 252. Paderborn, 2009. ISBN 978-3-939350-71-2, S. 155-170

[Böh14] BÖHME, T.: Steuerungsprogrammierung durch Vorführen an virtuellen Modellen. In: *17. IFF-Wissenschaftstage 2014, Digitales Engineering zum Planen, Testen und Betreiben technischer Systeme*. Magdeburg, 2014. ISBN 2196-7598, S. 25–30

[GBT+12] GAY-BELLILE, V.; BOURGEOIS, S.; TAMAAZOUSTI, M.; COLLETTE, S. N.; KNODEL, S.: A mobile markerless Augmented Reality system for the automotive field. 2012

[GNT+14] GAUGLITZ, S.; NUERNBERGER, B.; TURK, M.; HÖLLERER, T.: World-stabilized Annotations and Virtual Scene Navigation for Remote Collaboration. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, *UIST '14*. New York, NY, USA : ACM, 2014. ISBN 978-1-4503-3069-5, S. 449-459

[KB13] KENNEL, M.; BAYRHAMMER, E.: Virtuelles Teachen - Zuverlässige Steuerungsprogrammierung für Sondermaschinen durch Ablaufspezifikation am funktionalen 3D-Modell. In: *11. MAGDEBURGER MASCHINENBAU-TAGE*. Magdeburg, 2013 . ISBN 978-3-940961-90-7

[KM09] KLEIN, G.; MURRAY, D.: Parallel Tracking and Mapping on a camera phone. In: *8th IEEE International Symposium on Mixed and Augmented Reality, 2009. ISMAR 2009*, 2009, S. 83-86

[Sch14] SCHMUCKER, U.: Virtuelles Engineering zur Entwicklung von Sondermaschinen durchgängig nutzen. In: *wt Werkstattstechnik online* (2014), Nr. 3-2014

[WLS08] WAGNER, D.; LANGLOTZ, T.; SCHMALSTIEG, D.: Robust and unobtrusive marker tracking on mobile phones. In: *7th IEEE/ACM International Symposium on Mixed and Augmented Reality, 2008. ISMAR 2008*, 2008, S. 121-124

Autoren

Dr.-Ing. Simon Adler erwarb das Diplom der Medieninformatik 2005 an der FH-Wedel (Schleswig-Holstein). Nachdem er ein Jahr in der Entwicklung und Projektleitung im Bereich der Spieleentwicklung in Jena tätig war, nahm er 2006 das Studium in zum Master of Science in computational Visualistics an der Otto-von-Guericke Universität in Magdeburg auf. Nach seinem Abschluss 2008 wurde er wissenschaftlicher Mitarbeiter am Fraunhofer IFF. Hier befasst er sich mit mobilen Assistenzsystemen sowie mit dem Thema der Echtzeit-Chirurgiesimluation in dem er 2014 am Institut für Simulation und Graphik der Otto-von Guericke Universität promovierte. Herr Adler ist Gruppenleiter für die Arbeitsgruppe „mobile Assistenzsysteme“ im Geschäftsfeld Virtual Engineering des Fraunhofer IFF. Ein Schwerpunkt ist die mobile Assistenz für Betrieb, Wartung und Prüfung technischer Anlagen.

Alexa Kernchen studierte Informatik an der Otto-von-Guericke Universität in Magdeburg und erwarb 2012 das Diplom. Bereits zu Studienzeiten war sie im Bereich der Augmented Reality

und bildbasierter Verfolgung von Objekten am Fraunhofer IFF in Magdeburg als Hilfswissenschaftlerin tätig. Seit 2012 setzt sie diese Tätigkeit als wissenschaftliche Mitarbeiterin fort. Schwerpunkte liegen auf der Entwicklung und Umsetzung praxisnaher Software im Bereich mobiler Assistenzsysteme für Betrieb und Wartung von Maschinen und Anlagen.

Prof. Dr. sc. techn. Ulrich Schmucker wurde am 15.03.54 in Magdeburg geboren. Von 1972-77 studierte er theoretische Mechanik an der Lomonossov-Universität in Moskau (Russland). Von 1977-1983 war er wissenschaftlicher Mitarbeiter am Institut für Mechanik an der Akademie der Wissenschaften in Berlin. Von 1983-1991 war er Leiter des Geschäftsbereiches Sensorentwicklung am Institut für Automatisierungstechnik in Berlin, wo er seine Dissertation 1986 und seine Habilitation 1990 über Sensorsysteme in der Robotik abschloss. Von 1992 an war er Abteilungsleiter für Automatisierung am Fraunhofer Institut für Fabrikbetrieb und -automatisierung (IFF). Seit 2007 leitet er das Geschäftsfeld „Virtual Engineering“ am Fraunhofer IFF und ist seit 2008 Professor für mobile Robotersysteme an der Otto-von-Guericke Universität. Er hat ca. 30 Patente und über 100 Veröffentlichungen.

Tilman Reipsch ist seit 2011 Student an der Otto-von-Guericke Universität Magdeburg im Studiengang Computervisualistik. Seit 2013 arbeitet er neben dem Studium als Hilfswissenschaftler am Fraunhofer IFF in Magdeburg. Schwerpunkte seiner Tätigkeit ist die Anwendungsentwicklung im Bereich der Augmented Reality und mobilen Assistenzsysteme.

Eric Bayrhammer schloss sein Studium zum Diplomingenieur für Computervisualistik an der Otto-von-Guericke-Universität Magdeburg im Jahr 2006 ab. Anschließend war er zunächst als Hilfswissenschaftler am Fraunhofer IFF tätig und wurde hier kurze Zeit später wissenschaftlicher Mitarbeiter. Seit dem konnte er sein Themenfeld in Forschungs- und Industrieprojekten, von der automatisierten Generierung von Simulations- und Animationsmodellen, hin zur durchgängigen virtuellen Produktentwicklung, virtuellen Inbetriebnahme und Produktkonfiguration ausbauen. Herr Bayrhammer ist Gruppenleiter für die Arbeitsgruppe „Informatik“ im Geschäftsfeld Virtual Engineering des Fraunhofer IFF.

.

Fahrsimulation und Fahrerassistenzsysteme

Systematik zur Entwicklung von rekonfigurierbaren Fahrsimulatoren

Dr.-Ing. Bassem Hassan
Fraunhofer-Institut für Produktionstechnologie IPT,
Projektgruppe Entwurfstechnik Mechatronik
Zukunftsmeile 1, 33102 Paderborn
Tel. +49 (0) 5251 / 54 65 436, Fax. +49 (0) 5251 / 54 65 102
E-Mail: bassem.hassan@ipt.fraunhofer.de

Prof. Dr.-Ing. Jürgen Gausemeier, M. Eng. Kareem Abdelgawad,
Dr.-Ing. Jan Berssenbrügge, Dipl.-Ing. Michael Grafe
Heinz Nixdorf Institut, Universität Paderborn
Fürstenallee 11, 33102 Paderborn
Tel. +49 (0) 5251 / 60 62 67, Fax. +49 (0) 5251 / 60 62 68
E-Mail: juergen.gausemeier@hni.uni-paderborn.de

Zusammenfassung

Fahrsimulatoren werden seit Jahrzehnten erfolgreich in verschiedenen Anwendungsbereichen eingesetzt. Sie unterscheiden sich weitgehend in ihrer Struktur, Genauigkeit, Komplexität und in ihren Kosten. Heutzutage werden Fahrsimulatoren in der Regel individuell für eine spezielle Aufgabe entwickelt und haben typischerweise eine festgelegte Struktur. In speziellen Anwendungsbereichen werden jedoch unterschiedliche Varianten eines Fahrsimulators benötigt. Es besteht daher Handlungsbedarf für die Entwicklung eines rekonfigurierbaren Fahrsimulators, der es dem Betreiber ermöglicht, ohne umfassende Fachkenntnisse einfach und schnell anwendungsspezifische Varianten des Fahrsimulators zu erstellen. In dieser Arbeit wird eine Systematik für die Entwicklung von rekonfigurierbaren Fahrsimulatoren entwickelt. Das Vorgehensmodell beschreibt die benötigten Entwicklungsphasen, die vollständigen Aufgaben jeder Phase und die in der Entwicklung eingesetzten Methoden. Das Konfigurationswerkzeug orchestriert die Lösungselemente des Fahrsimulators und ermöglicht dem Betreiber des Fahrsimulators, durch Auswählen einer Kombination von Lösungselementen nach dem Prinzip des morphologischen Kastens verschiedene Varianten des Fahrsimulators zu erstellen. Die Entwicklungssystematik wurde durch die Erstellung von drei unterschiedlichen Varianten eines rekonfigurierbaren Fahrsimulators validiert.

Schlüsselworte

Fahrerassistenzsysteme (FAS), rekonfigurierbarer Fahrsimulator, Konfigurations-Mechanismus, Lösungselemente

A Design Framework for Developing a Reconfigurable Driving Simulator

Abstract

Driving simulators have been used successfully in various application fields for decades. They vary widely in their structure, fidelity, complexity and cost. Nowadays, driving simulators are usually custom-designed for a specific task and they typically have a fixed structure. Nevertheless, using the driving simulator in an application field, such as the development of the Advanced Driver Assistance Systems (ADAS), requires several variants of the driving simulator. Therefore, there is a need to develop a reconfigurable driving simulator which allows its operator to easily create different variants without in-depth expertise in the system structure. In order to solve this challenge, a Design Framework for Developing a Reconfigurable Driving Simulator has been developed. The design framework consists of a procedure model and a configuration tool. The procedure model describes the required development phases, the entire tasks of each phase and the used methods in the development. The configuration tool organizes the driving simulator's solution elements and allows its operator to create different variants of the driving simulator by selecting a combination of the solution elements, which are like building blocks. The design framework is validated by developing an ADAS reconfigurable driving simulator and by creating three variants of this driving simulator.

Keywords

Driver Assistance Systems (DAS), Reconfigurable Driving Simulator, Configuration Mechanism, Solution Elements

1 Einleitung

Der wachsende komplexe Einfluss der modernen Fahrerassistenzsysteme (FAS) auf das Fahrzeug macht es zunehmend schwieriger, diesen zu verstehen und somit das Zusammenspiel zwischen FAS, Fahrzeug, Fahrzeugumfeld und dem Fahrer zu analysieren. Fahrsimulatoren spielen eine entscheidende Rolle bei der Entwicklung neuer Fahrzeuge und ihren Teilsystemen, durch ihre reproduzierbaren Testbedingungen und sicheren Testumgebungen sowie als Mittel zur Reduzierung der Entwicklungszeit und -kosten. Dennoch sind Fahrsimulatoren für einen bestimmten Zweck entworfen und gebaut, um eine spezifische Analyse-Aufgabe in einer vordefinierten Umgebung zu unterstützen. Die Anpassung solcher Systeme zur Unterstützung neuer Funktionen oder Anwendungen ist sehr komplex, zeitaufwendig und oft nicht machbar. So entsteht mit der zunehmenden Rolle der FAS ein Bedarf für sehr anpassungsfähige und rekonfigurierbare Systeme, die problemlos auf neue Funktionen zugeschnitten werden können. Ein solch rekonfigurierbarer Fahrsimulator ähnelt einem Baustein-Konzept. Die verschiedenen Simulationsmodelle, Software- und Hardware-Komponenten, die den Fahrsimulator bilden, müssen dabei weitgehend kombinierbar sein. Darüber hinaus gibt es Simulationsmodelle mit unterschiedlichen Detaillierungsgraden, die von einfachen Low-Fidelity-Modellen bis hin zu ihren jeweiligen komplexen High-End-Modellen reichen, welche eine detaillierte Simulation aufzeigen. Auch die Hardware-Komponenten reichen von einfachen bis hin zu komplexen Komponenten, die unterschiedliche Simulator-Setups bilden, welche für spezifische Aspekte der FAS im Testverfahren passen [HBA+13]. Ein rekonfigurierbarer Fahrsimulator erfordert kompatible Software- und Hardware-Schnittstellen und einen zuverlässigen Kontrollmechanismus, um konsistente Konfigurationen des Systems zu erreichen. Diese Arbeit ist Teil des Projekts TRAFFIS (Test- und Trainingsumgebung für fortgeschrittene Fahrerassistenzsysteme). Der zweite Abschnitt dieses Beitrags wird den Stand der Technik beschreiben. Der dritte Abschnitt beschreibt der Systematik zur Entwicklung eines rekonfigurierbaren Fahrsimulators. Der vierte Abschnitt wird eine Zusammenfassung und die zukünftige Arbeit darstellen [Has14].

Lösungsansatz

Die Entwicklung von rekonfigurierbaren mechatronischen Systemen, die fast ausschließlich aus standardisierten modularen Komponenten bestehen, folgt dem Prinzip des "morphologischen Kastens". Der typische Virtual Prototyping-Zyklus besteht aus drei Phasen: Modellierung, Simulation und Analyse. Die Modellierung ist das Entwickeln von vereinfachten formalen Modellen des Systems in der Entwicklung. Die Systemmodelle stellen die Systemeigenschaften. Der Simulationsprozess stellt die Berechnung der Systemmodells mit Hilfe numerischer Algorithmen dar, um das Systemverhalten zu simulieren. Der Analyseprozess ist die Ergebnisdaten der Simulation sind zu interpretieren. Im Wesentlichen sind dabei die relevanten Informationen zu extrahieren, aufzubereiten und anschaulich zu visualisieren [GEK01], [Kre12].

Zur Rekonfiguration eines bestehenden Fahrsimulators, besteht die Notwendigkeit, eine Phase zwischen der Modellierungs- und Simulationsphase hinzuzufügen (siehe Bild 1). Die neue Phase ist die in Bild 1 gezeigte Konfigurationsphase. In der Konfigurationsphase kann der Fahrsimulator Betreiber die gewünschten Lösungselemente auswählen, um eine aufgabenspezifische Variante des Fahrsimulators zu generieren.

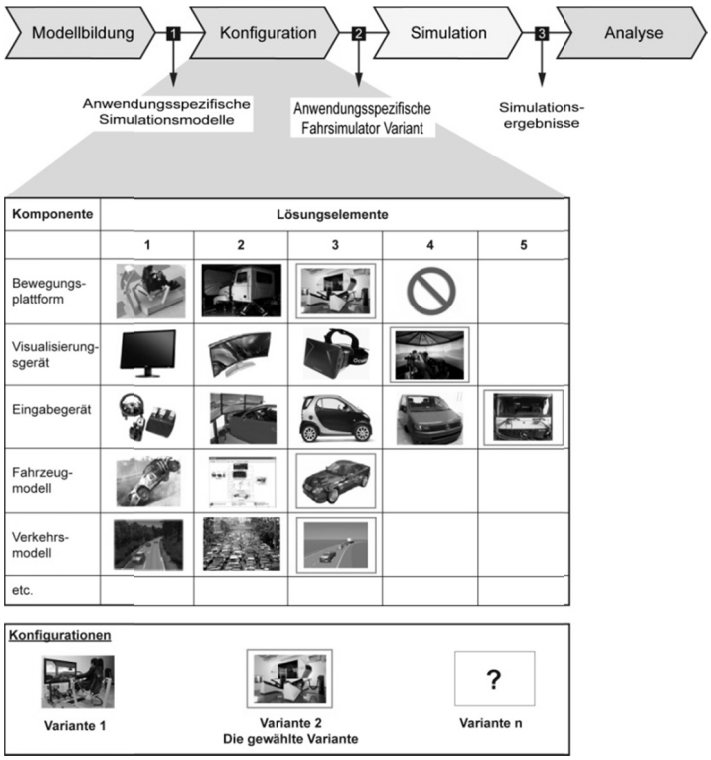


Bild 1: Entwicklungssystematik des rekonfigurierbaren Fahrsimulators

2 Verwandte Arbeiten

Moderne Fahrsimulatoren zeigen ein breites Spektrum an Anwendungen, z. B. Fahr- und Sicherheitstrainings, Fahrzeugbewertung, Straßengestaltung und Fahrdynamiksimulation. Sie können für Forschungszwecke verwendet werden, um das Verhalten des Fahrers zu untersuchen, neue Fahrzeug-Teilsysteme wie FAS zu entwickeln und zu evaluieren. Sie bieten Fahrertrainings und vieles mehr. Zudem reichen Fahrsimulatoren von teuren Low-Cost-Desktop-Systemen, die auch in kleinen Unternehmen eingesetzt werden, bis High-End-Systemen, welche bei einem ganzen Fahrzeug, das auf einer beweglichen Plattform montiert wird, eingesetzt wird.

Aufgrund der großen Variation der Fahrsimulatoren-Komplexität, der Genauigkeit, der Kosten und der Verwendungszwecke, stellt der Spezifikations- und Auswahlprozess eines Fahrsimulators einen mühsamen Prozess dar [WH09], [Neg07]. Üblicherweise ist der Fahrsimulator-Benutzer kein Experte in Fahrsimulator- Techniken und kann sich nicht einfach für eine Fahrsimulator-Anpassung mit den jeweiligen Anforderungen entscheiden. Es wurden Methoden für die Auswahl eines Fahrsimulators entwickelt. Zum

Beispiel stellt [Neg07] eine Methode, die es den Entwicklern von neuen Fahrzeug-Technologien erlaubt, ein Fahrsimulator-Konzept zu entwerfen und zu gestalten, um entsprechende Anforderungen anzupassen. Dies erfolgt in Form von Beratung, welches Teilsystem in welchem Anwendungsszenario benutzt werden muss. Im Folgenden werden drei Beispiele von früheren Versuchen zum Aufbau eines rekonfigurierbaren Fahrsimulators dargestellt.

2.1 Fahrsimulator der Universität von Zentralflorida (UCF)

Der UCF Fahrsimulator im Zentrum für fortgeschrittene Simulation von Transportsystemen ist ein fahrtüchtiger Simulator mit einer hohen Genauigkeit und einer immensen virtuellen Umgebung. Der UCF Fahrsimulator dient als Forschungssystem für multiplanare Untersuchungen im Forschungsbereich des Verkehrswegebaus [AYR+07]. Der UCF Fahrsimulator wurde so entwickelt, dass sowohl das simulierte Fahrzeug als auch das Fahrzeug-Mock-Up austauschbar sind [GKR03].

2.2 Trainingssimulator der Universität Valencia

Der Trainingssimulator in Valencia ist ein Kransimulator, gerichtet an Arbeitskräfte und Betreiber von Hafenanlagen und dient als Trainingsumgebung, welche beim Be- und Entladen von Schiffen vonnöten ist. Dieser Fahrsimulator wurde so entwickelt, dass sowohl das Model des Krans als auch das Projektionssystem austauschbar sind.

2.3 BVG Straßenbahn-Fahrsimulator

Der BVG Simulator der Berliner Verkehrsbetriebe, Betreiber des größten Straßenbahnen Netzwerks in Deutschland [GGT13], bietet eine Trainingsumgebung für Fahrer der Straßenbahnen. Der BVG Simulator wurde so entwickelt, dass die Simulation des Straßenbahn-Zugs jederzeit austauschbar ist. Angeboten werden drei Möglichkeiten des Simulators, welche durch Austauschen einer oder mehrerer Komponenten Konfigurierbarkeit und Flexibilität des Simulators bieten. Diese Beispiele zeigen, dass es für den Endbenutzer bis heute keine Methode oder Herangehensweise zum einfachen Umstrukturieren des Systems gibt, ohne tiefgreifendes Wissen über die Architektur und die Komponenten des Systems zu haben.

3 Vorgehensmodell zur Entwicklung eines rekonfigurierbaren Fahrsimulators

Das Vorgehensmodell ist der Kern *der Systematik zur Entwicklung eines rekonfigurierbaren Fahrsimulators*. Das Vorgehensmodell beschreibt die benötigten Entwicklungsphasen, die vollständigen Aufgaben jeder Phase und die in der Entwicklung eingesetzten Methoden. Die Entwicklung eines rekonfigurierbaren Fahrsimulators ist in sechs

Phasen gegliedert. Bild 2 zeigt das Vorgehensmodell als Phasen/Meilenstein-Diagramm mit den einzelnen Phasen, den durchzuführenden Tätigkeiten und den jeweiligen Resultaten. Generell wird als erstes das Fahrsimulatorsystem entwickelt. Darauf aufbauend wird anschließend ein gewünschte Fahrsimulatorvariante generiert.

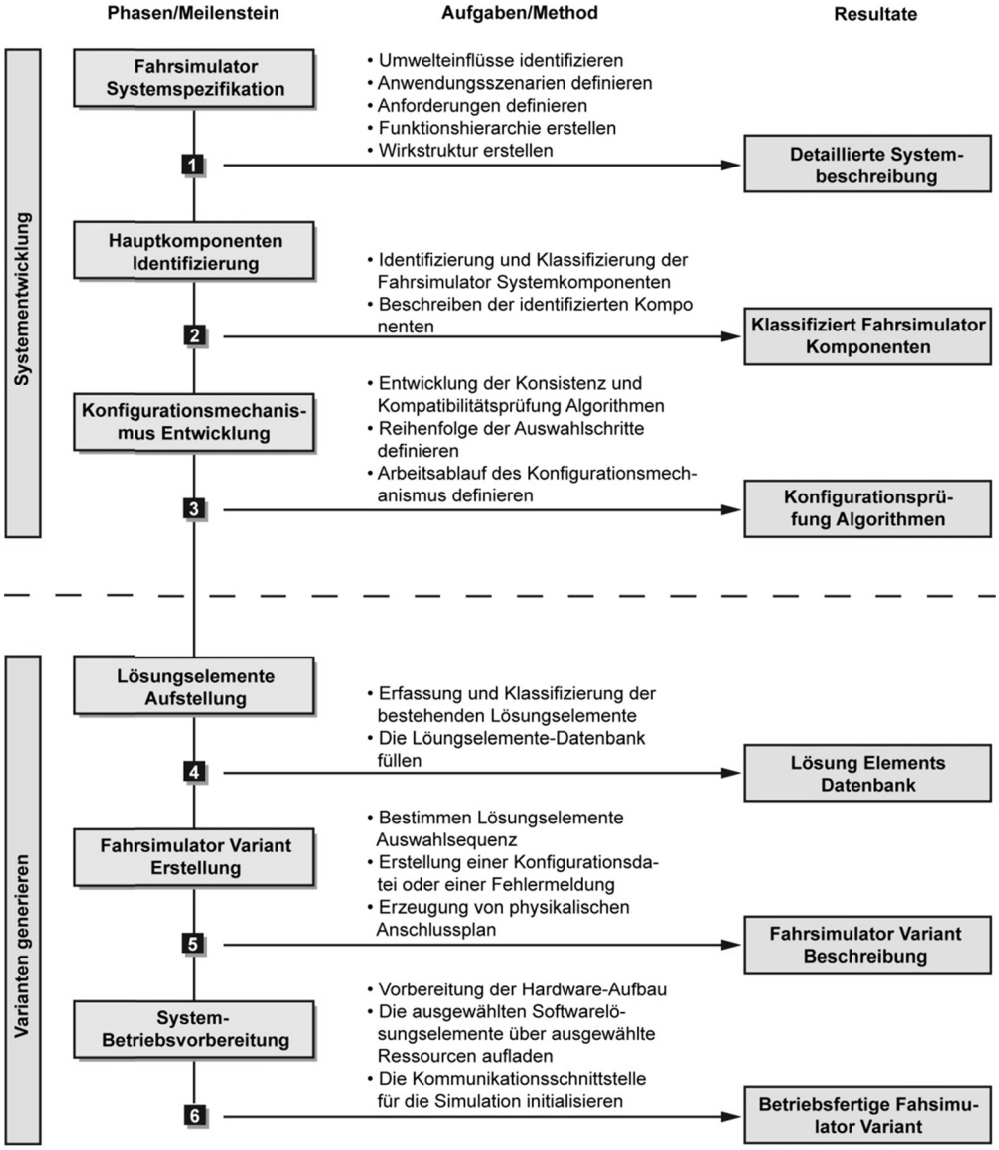


Bild 2: Vorgehensmodell zur Entwicklung eines rekonfigurierbaren Fahrsimulators

Die einzelnen Tätigkeiten innerhalb der Phasen und die eingesetzten Hilfsmittel werden in den folgenden Abschnitten kurz erläutert.

3.1 Systemspezifikationen

Das Ziel der ersten Phase ist, ein Fahrsimulatorsystem zu spezifizieren. Fahrsimulatoren sind komplexe fachübergreifende mechatronische Systeme. Daher gibt es die Notwendigkeit, eine domainübergreifende Spezifikationstechnik zu benutzen. Die Spezifikati-

onstechnik **CONSENS** (**CON**ceptual design **S**pecification technique for the **EN**gineering of complex **S**ystems) [VG12] wird im Rahmen dieser Arbeit verwendet werden.

Wie in Bild 3 gezeigt, teilt die entwickelte Spezifikationstechnik das Lösungsprinzip in verschiedene Teilmodelle, die mit folgenden Aspekten übereinstimmen: Anforderungen, Umfeld, Anwendungsszenarien, Funktionen, Struktur, Form und Verhalten.

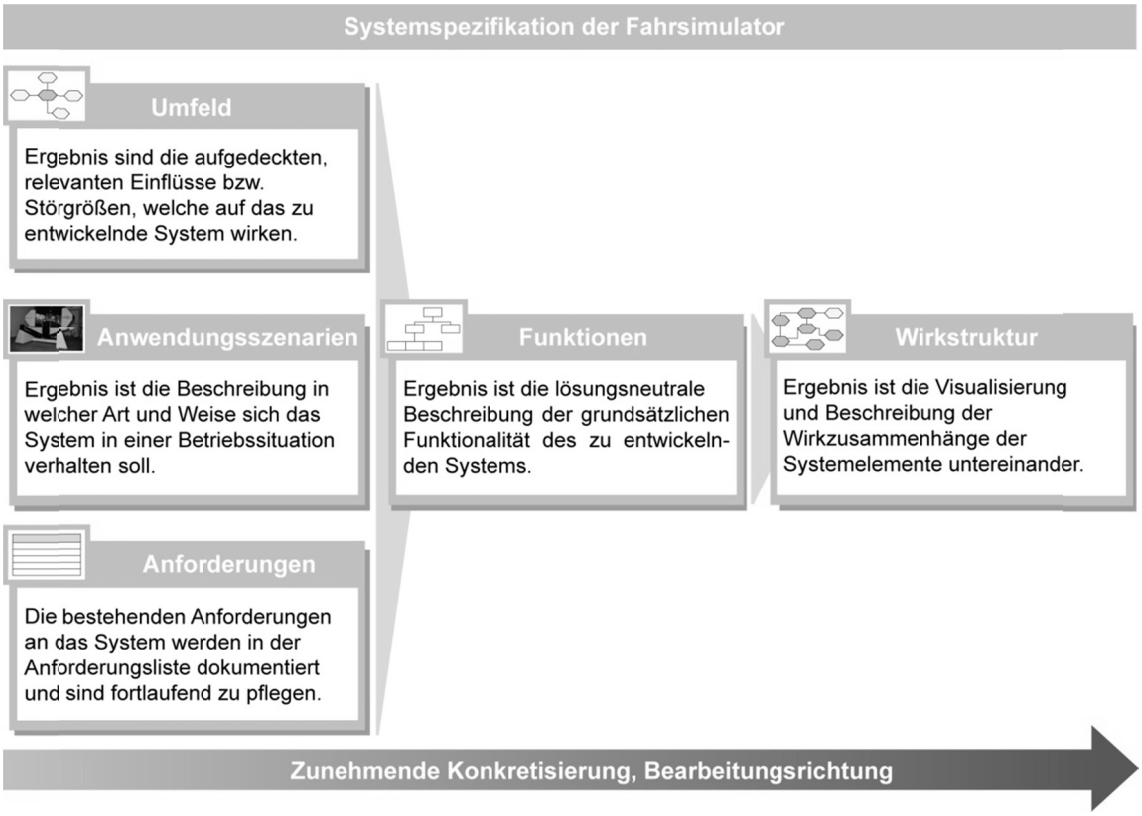


Bild 3: Arbeitsablauf um einen rekonfigurierbaren Fahrsimulator zu spezifizieren, nach CONSENS [VG12]

Wir benutzen CONSENS um Lösungsprinzipien des Fahrsimulators zu entwickeln. Dieser Prozess wurde in mehreren Workshops mit Entwicklern des Fahrsimulators und Benutzern, die im TRAFFIS Projekt involviert sind, entwickelt. Die wichtigsten Resultate werden im nächsten Abschnitt beschrieben.

Umfeld: Die Untersuchung des Fahrimulatorumfelds zeigt, dass externe Einflüsse von Fahrer, Fahrsimulatorassistent, Fahrlehrer, sowie von Energiequellen und dem Baugrund auf den Fahrsimulator einwirken können.

Anwendungsszenarien: Wie in der Einleitung beschrieben, fokussiert das Projekt TRAFFIS auf das Testen und Training von FAS. Das verwendete FAS-Beispiel ist ein Steuergerät für Scheinwerfer (Headlight Control Module). HCM ist ein Fahrerassistenzsystem, welches von Varroc entwickelt wurde [Sch07], und dient dem Zweck der Steuerung von Scheinwerfern, um eine optimale Straßenausleuchtung anhand der möglichst uneingeschränkten Benutzung des Fernlichts zu gewährleisten.

Die Projektpartner definierten sieben Applikationsszenarien, welche in drei verschiedene Fahrsimulatorvarianten klassifiziert und zusammengefast werden:

1. TRAFFIS Komplettversion: Dieses Applikationsszenario hat die komplexeste Struktur mit einer vollsteuerbaren Bewegungsplattform. Ziel ist es eine FAS Kontrolleinheit in zeitgleicher Verbindung mit einem Kamerasystem (HiL) zur Observation des Fahrers (DiL).
2. TRAFFIS portable Version: Dieses Applikationsszenario ist eine abgespeckte Version der Komplettversion. Ziel ist die Schulung von LKW-Fahrern mit dem neuen FAS System. Das Training findet im Logistikzentrum vor Ort statt, daher wird eine portable Version mit einer vereinfachten Bewegungsplattform benötigt.
3. TRAFFIS Light Version: Der Zweck der Applikationsszenarios ist der Test der FAS Software nach dem Software-in-the-Loop (SiL)-Prinzip. Hierbei wird ein computerbasierendes System ohne Bewegungsplattform benutzt. Diese Version des Systems dient FAS Entwicklern, in ersten Phasen des Projekts, zum täglichen Testen der Kontrollalgorithmen.

Anforderungen: Dieses Partialmodell sammelt und ordnet die Systemanforderungen, die im Rahmen der Systementwicklung erfüllt und umgesetzt werden müssen. Die Anforderungsliste umfaßt funktionale und nichtfunktionale Anforderungen [GFD+09]. Dabei wird zusätzlich zwischen Forderungen und Wünschen unterschieden [PBF+07].

Funktionen: Dieses Partialmodell wird auf der Basis der Partialmodelle zum Umfeld, den Anwendungsszenarien und den Anforderungen erstellt. Es beschreibt das System und die Funktion sämtlicher Systemkomponenten in einer Top-Down-Hierarchie [GFD+09]. Jeder Block beschreibt eine Unterfunktion des Systems. Funktionskataloge unterstützen dabei nach BIRKHOFER [Bir80] und LANGLOTZ [Lan00] das Erstellen der Funktionshierarchie.

Wirkstruktur: Das Partialmodell zur Wirkstruktur wird auf Basis der Ergebnisse der vorangegangenen Partialmodelle, insbesondere dem Funktionsmodell, aufgebaut. Die Wirkstruktur beschreibt das Gesamtsystem detaillierter anhand der Wirkprinzipien der einzelnen Systemkomponenten. Es definiert die Systemkomponenten, deren Eigenschaften und Wirkflächen sowie das Zusammenspiel der Komponenten. In Abhängigkeit von der Modellierungstiefe kann jedes Systemelement abstrakt durch sein Wirkprinzip oder Lösungsmuster beschrieben warden. Zusätzlich beschreiben Stoff-, Energie- und Informationsflüsse sowie die logischen Beziehungen untereinander das Zusammenwirken der Systemelemente [GFD+09].

3.2 Komponenten-Identifikation

Auf Basis der Ergebnisse der ersten Phase ist das Ziel der zweiten Phase die Identifikation, Klassifikation und Definition der Fahrsimulatorkomponenten. Dabei wird bei der

Identifikation der Systemkomponenten des Fahrsimulators zwischen Schlüsselkomponenten, optionalen Komponenten und Lösungselementen unterschieden:

Jedes Fahrsimulatorsystem beinhaltet Unterkomponenten, welche im Kontext dieser Arbeit als **optionale Komponenten** klassifiziert werden können. Eine optionale Komponente beschreibt die Aufgabe eines Subsystems lösungsneutral durch seine Funktion. Im Gegensatz dazu sind andere Systemkomponenten für den Aufbau eines funktionstüchtigen Fahrsimulators obligatorisch. Diese Komponenten werden als **Schlüsselkomponenten** bezeichnet (siehe Bild 4). Um eine einsatzfähige Fahrsimulatorvariante zu erzeugen, muß jede Systemkomponente durch ein **Lösungselement** abgebildet werden, welches eine spezifische Umsetzung der Funktion der entsprechenden Systemkomponente darstellt.

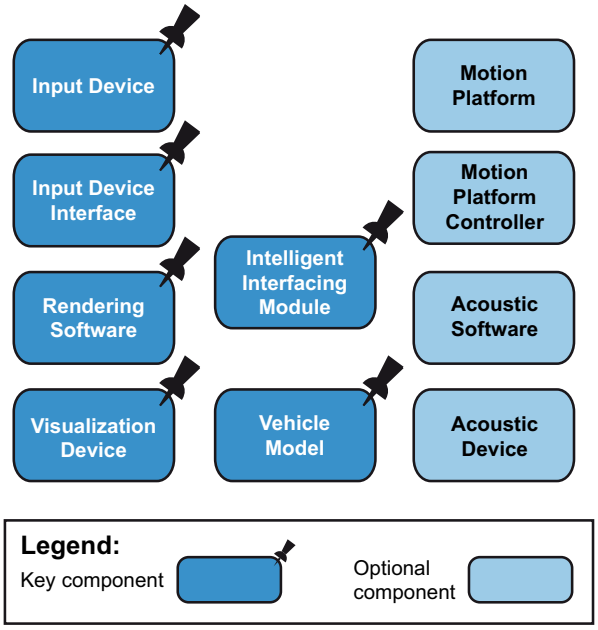


Bild 4: Beispiel für identifizierte optionale Komponenten und Schlüsselkomponenten.

Das Hauptresultat der zweiten Phase in die Identifikation der Hauptkomponenten des Fahrsimulators einschließlich der Lösungskonzepte. Die wichtigsten identifizierten Komponenten werden wie folgt beschrieben:

Input Device: Dies ist das reale Fahrzeug-Mock-Up (Fahrerkabine), welche die Mensch-Maschine-Schnittstelle zwischen dem Fahrer und dem Simulator darstellt.

Input Device Interface: Diese Software-Komponente verbindet die physischen Eingabesignale (Energieflüsse) des Eingabegerätes mit den entsprechenden computerinternen digitalen Signalen (Informationsflüsse).

Intelligent Interfacing Module (IIM): Diese Software spielt während der Laufzeit des Simulators eine wichtige Rolle. Sie liest die Konfigurationsdateien, welche die Kommunikation und Interface-Netz-Topologie zwischen den einzelnen Softwarekomponen-

ten beinhaltet, aus und errichtet die Kommunikation zwischen diesen Komponenten anhand von Echtzeit-Spezifikationen.

Vehicle Model: Diese Software-Komponente bildet das Fahrzeug und dessen sämtliche Komponenten durch einen Satz mathematischer Gleichungen ab, welche die Fahrdynamik des Fahrzeugs definieren.

Rendering Software: Die Rendering Software stellt das virtuelle Szenario auf einem Projektionssystem dar. Die Software rendert alle typischen Ereignisse und präsentiert dem Fahrer detailliert die aktuelle Fahrsituation, einschließlich anderer Verkehrsteilnehmer, der Strecke und der vollständigen Umgebung abseits des simulierten Fahrzeugs.

Visualization Device: Die gerenderten Szenen müssen dem Fahrer in der Fahrerkabine dargestellt werden. Folglich versorgt die Rendering Software verschiedene Visualisierungsendgeräte wie Bildschirme oder Projektoren.

Motion Platform: Ein aktiver Mechanismus, der es erlaubt, haptische Gefühle anhand Bewegungen darstellen zu können.

Motion Platform Controller: Um einen Controller für Aktoren des Bewegungssystems synthetisieren zu können, werden verschiedene Aktorelemente eingesetzt, um garantieren zu können, dass die Bewegungsplattform des Simulators dem gewünschten Bahnverlauf, generiert vom Bewegungsalgorithmus, folgt [AT12].

Akustiksoftware: Diese Software-Komponente erzeugt den computergenerierten Ton in der virtuellen Szene.

Akustisches Gerät: Dies umfaßt das notwendige Soundsystem für die Tonerzeugung im Fahrsimulator.

Simulationsrechner: Dieses stellte eine Ressourcen-Komponente dar, welche die Softwarekomponenten des Systems verarbeitet.

Schnittstelle zum Simulationsrechner: Dieses ist eine Ressourcenkomponente, welche die Hardware-Komponenten mit dem Simulationsrechner verbindet, indem die physikalischen Signale (Energieflüsse) in ihre entsprechenden digitalen Signale (Informationsflüsse) umgewandelt werden.

3.3 Entwicklung des Konfigurationsmechanismus

Dieses ist der dritte und letzte Schritt der Entwicklungsphase. Ziel dieses Schrittes ist die Entwicklung eines Konfigurationsmechanismus, der sicherstellt, daß die ausgewählten Lösungselemente zusammenarbeiten. Diese Prüfung erfolgt nach Auswahl der gewünschten Struktur und Lösungselemente. Der Konfigurationsmechanismus stellt die Konsistenz und Kompatibilität der ausgewählten Struktur und sämtlicher Lösungselemente sicher. Hiernach wird das Konfigurationsfile generiert. Es beinhaltet eine Liste

der ausgewählten Lösungselemente, die Topologie der Schnittstellen und gewählten Ressourcen.

Der Konfigurationsmechanismus überprüft zunächst die ausgewählten Lösungselemente. Die Installation folgt in der nächsten Phase. Die Entwicklung des Konfigurationsmechanismus vor der Installation der Lösungselemente ermöglicht dem Mechanismus auch unbekannte Lösungselemente zu berücksichtigen, welche in Zukunft hinzugefügt werden.

Es gibt zwei unterschiedliche Beziehungsarten zwischen ausgewählten Lösungselementen. Die erste Beziehung ist die logische Konsistenz zwischen den ausgewählten Lösungselementen. Die zweite Beziehung ist die Kompatibilität zwischen den Schnittstellen der ausgewählten Lösungselemente. Beide Beziehungsarten müssen durch den Konfigurationsmechanismus geprüft und bestätigt werden.

Algorithmus zur Konsistenzprüfung: Die Konsistenzbeziehung kann auf zwei Ebenen bestimmt warden. Die erste Ebene ist die logische Abhängigkeit zwischen Komponenten, welche festlegt, ob eine logische Korrelation zwischen den Komponenten besteht. Die zweite Ebene ist die logische Konsistenz zwischen Lösungselementen.

Logische Abhängigkeiten zwischen Komponenten: Die logische Beziehung beschreibt, ob zwei Komponenten logisch von einander abhängig sind. Die Bewegungsplattform und das Eingabegerät sind z.B. zwei voneinander abhängige Komponenten. Die Abhängigkeit besteht darin, daß das Eingabegerät auf die Bewegungsplattform montiert warden muß. Daher müssen die Abmaße und die Lastaufnahme der Plattform den Daten des Eingabegerätes entsprechen.

Algorithmus zur Kompatibilitätsprüfung: Eine der Zielsetzungen an die Erstellung eines rekonfigurierbaren Fahrsimulators ist die Möglichkeit, ein oder mehrere Lösungselemente hinzufügen, zu entfernen oder auszutauschen. Um einen rekonfigurierbaren Fahrsimulator aufzubauen, müssen die Schnittstellen zwischen den Anwendungen und den Modellen automatisch verbunden warden. Es besteht Bedarf für einen Algorithmus, der überprüft, ob die gewählten Lösungselemente untereinander kompatibel sind. Kompatibilität meint hier, ob die Schnittstellen der gewählten Lösungselemente untereinander passen. Jede Software-Komponente hat ein programmiersprachen-spezifisches Benennungssystem für die Ein- und Ausgabesignale. Zudem ist es erforderlich, das rekonfigurierbare System fortwährend um neue, unbekannte Lösungselemente zu erweitern. Dafür wurde ein allgemeines Schnittstellenkonzept für die Lösungselemente entwickelt, das sowohl bestehende Lösungselemente verwaltet und überprüft, als auch neue Lösungselemente, welche zukünftig dem System hinzugefügt werden können.

Generisches Schnittstellenkonzept für Lösungselemente: Um die sämtliche Lösungselemente vollständig miteinander verbinden zu können, ohne ein tiefgründiges Wissen über jedes Elements zu besitzen, wird jedes Element als Black-Box angesehen. Das bedeutet, dass nur die Schnittstellen für die Ein- und Ausgabe in Betracht gezogen

werden müssen. Um den Konfigurationsprozess flexibel und erweiterbar gestalten zu können, kann jedes Lösungselement, sofern die Ein- und Ausgabeschnittstellen definiert sind, hinzugefügt werden. Die einzig nötige Aufgabe um ein Lösungselement integrieren zu können, ist das Abbilden der Ein- und Ausgaben auf die vordefinierten Ein- und Ausgaben der ursprünglichen Komponenten, was als “signal multiplexing” bezeichnet wird.

Bild 5 zeigt ein Beispiel für das Konzept, welches ein Fahrzeug-Modell als Lösungselement integriert, bezeichnet als Black-Box, bei dem alle Signalbeschreibungen den aktuellen Komponenten zugeordnet werden müssen.

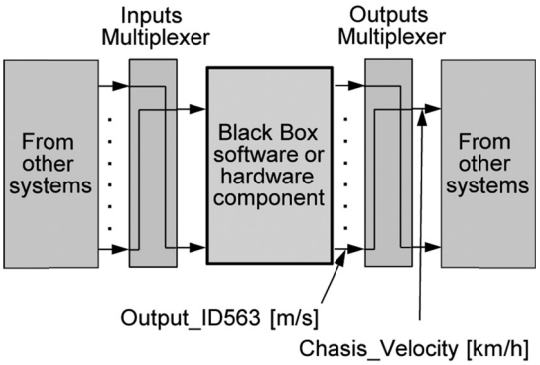


Bild 5: Integration eines Modells (Black-Box) in Fahrsimulator-System

Das Output-Signal, welches “Output_ID563 [m/s]” genannt wird, stellt das Fahrzeug mit der Testgeschwindigkeit und der Einheit m/s dar. Bei der aktuellen Komponente ist der Name jedoch "Chassis_Velocity" mit der Einheit km/h. Um dieses Fahrzeug-Modell zu integrieren, muss der Anwender alle Eingangs- und Ausgangssignale mit unterschiedlichen Namen und Einheiten den eindeutigen Namen und Einheiten der aktuellen Komponenten zuordnen. Dieses Komzept erlaubt dem Benutzer neue Lösungselemente mit geringem Aufwand in das bestehende System zu integrieren. Dieser Integrationsprozess wird für jedes neue Lösungselement ausgeführt.

3.4 Aufstellung der Lösungselemente

In dieser Phase wird eine benutzerfreundliche Oberfläche entwickelt, um den Benutzer die Wahl des geeigneten Lösungskonzepts und die Entwicklung einer Fahrsimulationskonfiguration zu ermöglichen. Das Ergebnis der dritten Phase ist der Konfigurationsmechanismus, der einen komplexen sequentiellen Prozess darstellt.

Um eine benutzerfreundliche Schnittstelle zu erstellen, muss in dieser Phase ein Konzept und eine Integration einer einfach zu bedienenden Software entwickelt werden. Die Software unterstützt den Anwender während des Auswahlprozesses anhand einer vordefinierten Reihenfolge, bis der Benutzer die Konfiguration abgeschlossen hat. Der Vorteil einer solchen Software ist das Ausführen eines Konfigurationsmechanismus im Hin-

tergrund ohne störende Einflüsse auf den Benutzer. Bild 6 zeigt einen Screenshot der entwickelten Software.

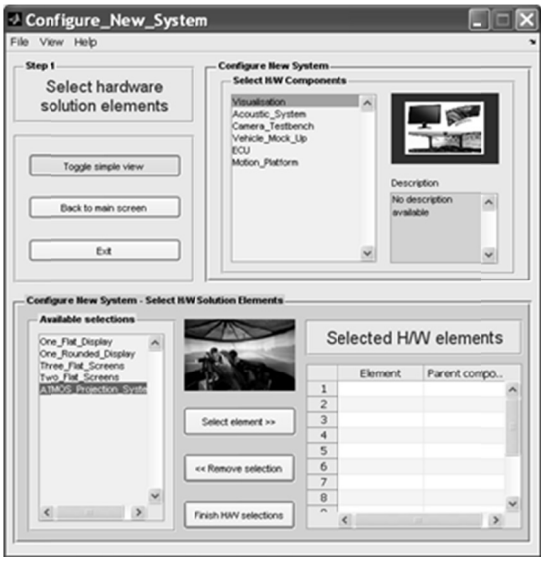


Bild 6: Screenshot von der Konfigurationssoftware

3.5 Erstellen einer Simulatorvariante

Um eine angemessenen Auswahlsequenz für die Lösungselemente sicherzustellen, müssen die Systemkomponenten und deren Interaktionen untersucht werden. Die Reihenfolge der Auswahl kann auf der Grundlage des Verwendungsbereichs geändert werden.

Die Komponenten des Fahrsimulators wurden bereits anhand der drei Hauptklassen klassifiziert: Hardware, Software und Ressourcen. Die Struktur des Fahrsimulators basiert auf den Hardware- und Softwarekomponenten und einigen Ressourcen.

Um die Reihenfolge der Auswahl sinnvoll zu erfolgen, werden die identifizierten Komponenten in Gruppen von Software und / oder Hardware auf der Basis der bei der Spezifikation der Wirkstruktur identifizierten Gruppen unterteilt. Dann kann der Auswahlprozess starten.

3.6 Starten der Simulation

In dieser Phase wird die Schnitstelle zwischen generierten Konfigurationen und der Simulationssoftware entwickelt. Das Resultat der vierten Phase ist eine Konfigurationsdatei. Diese enthält die ausgewählten Komponenten, Schnittstellentopologie, ausgewählte Ressourcen und Signale für die Analyse. Um die erzeugte Konfigurationsdatei verwenden zu können, besteht die Notwendigkeit, die Konfigurationsdatei mit den Schnittstellen der Simulationssoftware zu verknüpfen. Die Verbindung zwischen der Konfigurationsdatei und der Simulationssoftware wird anhand der Softwarekomponenten “As-

sistant“ und “Moderator” ermöglicht. Die Kopplung wird erfolgt anhand der nachfolgenden Schritte:

Softwareverteilung über Ressourcen: Die Komponente „Assistent“ liest die Konfigurationsdatei des Fahrsimulators aus. Sie ruft die Modelle und Softwarekomponenten aus der Dateiablage auf und verteilt sie auf die zur Verfügung stehenden Ressourcen (z.B. Rechner, Rendering-Knoten, etc.).

Software-Initialisierung: Die Komponente „Moderator“ initialisiert die Kommunikation zwischen allen Systemkomponenten, wie in der Schnittstellentopologie der Konfigurationsdatei definiert, und bereitet das System für den Start vor.

Simulations-Laufzeit: Sobald der Benutzer die Simulation startet, sorgt die Komponente „Moderator“ für die Kommunikation zwischen allen Systemkomponenten und protokolliert die ausgewählten Signale zur Analyse, während der Laufzeit, in einer Konfigurationsdatei, die in der Simulationsdatenbank abgelegt wird.

Nachbearbeitung: Nachdem eine Simulationssitzung beendet ist, speichert die Moderator-Software die Signale der Simulations-Laufzeit für eine anschließende Analyse.

In Bezug auf die Schnittstellentopologie (Moderator-Task während der Laufzeit) ist es eine Herausforderung, alle 23 Software-Komponenten des TRAFFIS Full Simulators miteinander zu verknüpfen. Diese Aufgabe wird oft manuell durch Verbinden der Ein- und Ausgabesignale der Komponenten abgeschlossen. Darüberhinaus muss die Konnektivität manuell angepasst werden, sobald sich die Struktur des Fahrsimulators verändert, um Vorgaben des Anwenders berücksichtigen zu können. Daher gibt es einen Bedarf für eine automatisierte Methode, um die Schnittstelle zwischen den ausgewählten Lösungselementen anpassen zu können. Dies erlaubt den Benutzer die Systemstruktur verändern oder Lösungselemente ersetzen zu können, ohne tiefgründiges Wissen über die Schnittstellentopologie haben zu müssen. Das Konzept der Kommunikationstopologie ist, dass alle Komponenten durch die Moderator-Software anhand eines Datenbusses für Ein- und Ausgabesignale verbunden werden, anstatt einer direkten Verbindung der Komponenten.

4 Resümee und Ausblick

Dieser Beitrag beschreibt ein Konzept für einen aufgabenspezifischen, rekonfigurierbaren Fahrsimulator zum Test und Training von FAS. Das Konzept zeigt, wie die einzelnen Teilkomponenten und ihre jeweiligen Simulationsmodelle mit unterschiedlichem Genauigkeits- und Detailgrad, in ein gültiges und funktionierendes Simulator-Setup, basierend auf zuvor definierten Anforderungen, konfiguriert wird. Wir haben das Konzept im Rahmen des Projektes TRAFFIS erfolgreich angewendet und konnten vielversprechende Ergebnisse auf der Basis dreier Simulatorprototypen (TRAFFIS Full Version, TRAFFIS Portable Version und TRAFFIS Light Version) erreichen. Diese Simula-

torvarianten sind abhängig von den unterschiedlichen Anwendungen, die auf das Testen und Training von FAS in einer Testumgebung fokussieren.

Innerhalb des Kontexts des TRAFFIS Projektes planen wir, das Konzept anhand der Entwicklung eines dritten Fahrsimulatorprototypen zu evaluieren und erarbeiten momentan weitere komplexere Anwendungsszenarien für den Test und die Entwicklung von FAS in einer komplexeren Fahrsimulatorumgebung.

Danksagung

Diese Arbeit als Teil des TRAFFIS Projektes (Test- und Trainingsumgebung für fortgeschrittene Fahrerassistenzsysteme), welches von der Europäischen Union "ERDF: European Regional Development Fund" und dem Ministerium für Wirtschaft, Energie, Industrie, Handel und Handwerk von Nordrhein-Westfalen finanziert wird – Deutschland mit dem "Ziel2"-Programm.

Literatur

[AT12] AL QAISI, I.; TRÄCHTLER, A.: Constrained Linear Quadratic Optimal Controller for Motion Control of ATMOS Driving simulator. Proc. Driving Simulation Conference, 2012, Paris, France.

[AYR+07] ABDEL-ATY, M.; YAN, X.; RADWAN, E.: Using the UCF driving simulator as a test bed for high risk locations – Technical Report. Florida State Department of Transportation, Tallahassee Research Center, University of Central Florida and Center for Advanced Transportation System Florida department of transportation, Orlando-Florida, USA, 2007

[Bir80] BIRKHOFER, H.: Analyse und Synthese der Funktionen technischer Produkte. VDI-Verlag Fortschritts-Bericht VDI-Z, Reihe 1, Nr. 70, Düsseldorf, Germany, 1980

[GEK01] GAUSEMEIER, J.; EBBESMEYER, P.; KALLMEYER, F.: Produktinnovation – Strategische Planung und Entwicklung der Produkte von morgen. Carl Hanser Verlag München, 2011. ISBN 3-446-21631-6

[GFD+09] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Specification technique for the description of self-optimizing mechatronic systems. Research In: Research in Engineering Design, November 2009, Volume 20, Issue 4, Springer, London, 2009, pp. 201-223. ISSN 0934-9839

[GGT13] GAUSEMEIER, J.; GAUKSTERN, T.; TSCHIRNER, C.: Systems Engineering Management Based on a Discipline-Spanning System Model. In: C.J.J. Paredis, C. Bishop, D. Bodner (Eds.): Conference on Systems Engineering Research (CSER'13), 19-22 March 2013, Atlanta, USA, 2013

[GKR03] GUE, D.; KLEE, H.; RADWAN, E.: Comparison of Lateral Control in a Reconfigurable Driving Simulator. In: Proceedings of Driving Simulation Conference North America, October 8-10 2003, Dearborn, USA, Michigan, 2003, ISSN 1546-5071

[Has14] HASSAN, B.: A Design Framework for Developing a Reconfigurable Driving Simulator. Dissertation, der Fakultät Maschinenbau der Universität Paderborn, HNI-Verlagsschriftenreihe, Band 333, 2014

[HBA+13] HASSAN, B.; BERSSENBRÜGGE, J.; ALQAISI, I.; STÖCKLEIN, J.: Reconfigurable Driving Simulator for Testing and Training of Advanced Driver Assistance Systems. In: Proceed-

ings of 2013 IEEE International Symposium on Assembly and Task Planning (ISAM), July 30 - August 2 2013, Xi'an, China, 2013, pp. 337-339. ISBN 978-1-4799-1656-6

[Kre12] KREFT, S.: Systematik zur effizienten Bildung geospezifischer Umgebungsmodelle für Fahrsimulationen. Dissertation, der Fakultät Maschinenbau der Universität Paderborn, HNI-Verlagsschriftenreihe, Band 305, 2012

[Lan00] LANGLOTZ, G.: Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte. Forschungsberichte aus dem Institut für Rechneranwendung in Planung und Konstruktion RPK der Universität Karlsruhe, Shaker Verlag, 2000

[Neg07] NEGELE, J.: Anwendungsgerechte Konzipierung von Fahrsimulatoren für die Fahrzeugentwicklung. Dissertation, Fakultät für Maschinenwesen der Technischen Universität München, 2007

[PBF+07] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung – Methoden und Anwendung. Springer –Verlag, Berlin, 7. Auflage, 2007

[Sch07] SCHMIDT, C.: How to Make an AFS System Predictive: ADASIS Interface Implementation. In: Proceedings of the 7th International Symposium on Automotive Lighting, Darmstädter Lichttechnik Series, Volume 12, September 25-26 2007, Darmstadt, Germany. 2007, pp. 482-484. ISBN 978-3-8316-0711-2

[VG12] VAßHOLZ, M.; GAUSEMEIER, J.: Cost-Benefit Analysis – Requirements for the Evaluation of Self-Optimizing Systems. In: Proceedings of the 1st Joint International Symposium on System Integrated Intelligence 2012 – New Challenges for Product and Production Engineering, June 27-29 2012, Hannover, Germany, 2012, pp. 14-16

[WH09] WEINBERG, G.; HARSHAM, B.: Developing a Low-Cost Driving Simulator for the Evaluation of In-Vehicle Technologies. In: Proceedings of the First International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2009) September 21-22 2009, Essen, Germany, 2009

Autoren

Dr.-Ing. Bassem Hassan ist in 1982 in El Dakahlia, Ägypten geboren. Er studierte Applied Mechatronics an der Universität Paderborn, anschließend war er zwei Jahre bei dSPACE als Applikationsingenieur tätig. Von 2009 bis 2014 war er als wissenschaftlicher Mitarbeiter am Lehrstuhl für Produktentstehung des Heinz Nixdorf Instituts der Universität Paderborn unter der Leitung von Prof. Dr.-Ing. Jürgen Gausemeier tätig. Seit 2015 ist er als Senior Experte für Produktentstehung am Fraunhofer-Institut für Produktionstechnologie IPT, Projektgruppe Entwurfstechnik Mechatronik. In Rahmen seiner Tätigkeit hat er zahlreiche Forschungs- und Industrieprojekte auf dem Gebiet der virtuellen Produktentstehung geleitet und durchgeführt. Seine Forschungsschwerpunkte liegen im Bereich Entwicklung mechatronischer Systeme, insbesondere Fahrsimulatoren.

Prof. Dr.-Ing. Jürgen Gausemeier ist Seniorprofessor am Heinz Nixdorf Institut der Universität Paderborn. Seine Arbeitsschwerpunkte sind Strategische Produkt- und Technologieplanung sowie Entwicklungsmethodik Mechatronik/ Systems Engineering. Er war Sprecher des Sonderforschungsbereiches 614 „Selbstoptimierende Systeme des Maschinenbaus“ und von 2009 bis 2015 Mitglied des Wissenschaftsrats. Jürgen Gau-

semeier ist Initiator und Aufsichtsratsvorsitzender des Beratungsunternehmens UNITY AG. Seit 2003 ist er Mitglied von acatech – Deutsche Akademie der Technikwissenschaften und seit 2012 Vizepräsident. Ferner ist Jürgen Gausemeier Vorsitzender des Clusterboards des BMBF-Spitzenclusters „Intelligente Technische Systeme Ostwestfalen-Lippe (it´s OWL)“.

M. Eng. Kareem Abdelgawad studierte Maschinenbau mit Schwerpunkt Mechatronik an der Universität Paderborn. Sein Forschungsinteresse umfasst die Bereiche Modellierung und Simulation, Fahrerassistenzsysteme und Fahrsimulation. Seit 2013 ist er wissenschaftlicher Mitarbeiter am Heinz Nixdorf Institut der Universität Paderborn. Im Rahmen mehrerer Forschungs- und Industrieprojekte beschäftigt er sich mit der Entwicklung neuer Simulations- und Testkonzepte für Fahrzeugsysteme mithilfe der Technologien von Virtual Engineering.

Dr.-Ing. Jan Berssenbrügge ist wissenschaftlicher Mitarbeiter in der Fachgruppe Produktentstehung am Heinz Nixdorf Institut der Universität Paderborn. Seine Forschungsschwerpunkte liegen in den Bereichen Virtual & Augmented Reality, Virtual Engineeringn sowie Virtual Prototyping und Simulation. Herr Berssenbrügge studierte Informatik an der Universität Paderborn und promovierte am Heinz Nixdorf Institut mit einem Thema zum Virtual Prototyping von Scheinwerfersystemen.

Dipl.-Ing. Michael Grafe studierte an der TU Clausthal Maschinenbau. Er ist Oberingenieur am Heinz Nixdorf Institut der Universität Paderborn und seit 1995 maßgeblich am Auf- und Ausbau der Forschungsaktivitäten im Kontext Virtual Prototyping und Simulation am Heinz Nixdorf Institut sowie des OWL-Kompetenzzentrums für Virtual Prototyping & Simulation (OWL ViProSim e. V.) beteiligt. Seit 2011 ist er Geschäftsführer der Fraunhofer Projektgruppe Entwurfstechnik Mechatronik in Paderborn.

.

Visualisierung von Scheinwerfersystemen für das Virtual Prototyping von lichtbasierten Fahrerassistenzsystemen

Jan Berssenbrügge, Prof. Dr.-Ing. Ansgar Trächtler
Heinz Nixdorf Institut
Universität Paderborn
Fürstenallee 11, 33102 Paderborn
Tel. +49 (0) 5251 / 60 62 32, Fax. +49 (0) 5251 / 60 62 68
E-Mail: {Jan.Berssenbruegge@hni.upb.de, Ansgar.Traechtler}@hni.upb.de

Christoph Schmidt
Varroc Lighting Systems GmbH
Toyota-Allee 7, 50858 Köln-Marsdorf
E-Mail: cschmi27@varroclighting.com

Zusammenfassung

Für das Virtual Prototyping von lichtbasierten Fahrerassistenzsystemen werden zunehmend Fahrsimulatoren eingesetzt, welche die Simulation einer virtuellen Nachtfahrt ermöglichen. Hier wird das Zusammenwirken des menschlichen Fahrers mit dem Assistenzsystem untersucht, welches die Ausleuchtung des Straßenraums vor dem Fahrzeug bei Nacht optimiert. Für diese Untersuchungen kommen spezialisierten Fahrsimulatoren zum Einsatz, welche neben der normalen Fahrsimulation zusätzlich die hohen Anforderungen an die Visualisierung der nächtlichen Szenerie, die Simulation des Scheinwerferlichtes bei der virtuellen Nachtfahrt und die Anbindung an das Steuergerät der Scheinwerfer unterstützen.

Der Beitrag stellt das Visualisierungssystem des rekonfigurierbaren Fahrsimulators des Forschungsprojektes TRAFFIS vor und beschreibt die spezielle Ausrichtung auf das Virtual Prototyping eines lichtbasierten Fahrerassistenzsysteme (FAS), welches auf einem Scheinwerfer mit einer Kombination der Funktionen Glare-Free High Beam (GFHB) und Predictive Adaptive Frontlighting System (PAFS) für ein kartenbasiertes Kurvenlicht und blendfreies Fahren mit Fernlicht des Projektpartners Varroc Lighting Systems im Rahmen eines Innovationsprojektes von TRAFFIS aufsetzt.

Schlüsselworte

Glare-Free High Beam (GFHB), Predictive Adaptive Frontlighting System (PAFS), Headlight Control Module (HCM), Driver-in-the-Loop, Visualisierung von Scheinwerferprototypen, High Dynamic Range (HDR) Rendering, Tone Mapping, Virtual Prototyping

Visualisation of Automotive Headlight Systems for the Virtual Prototyping of Light-based Driver Assistance Systems

Abstract

Driving simulators that are capable of simulating a virtual drive at night are increasingly used for the virtual prototyping of light-based driver assistance systems. Here, the interplay between driver and assistance system, which enhances the illumination of the road ahead of the vehicle, is investigated. For such investigations, special driving simulators are applied that enable not only a standard driving simulation, but also cover the special requirements for the visualisation of a driving scenery at night, the simulation of automotive headlights during a virtual drive at night, and the interface to a headlight control module (HCM) that operates the physical headlight prototypes.

This paper presents the visualisation system of the reconfigurable driving simulator from the research project TRAFFIS and describes the special application focus on the virtual prototyping of a light-based driver assistance system from the project partner Varroc Lighting Systems. The light-based DAS bases on a headlight prototype that combines a glare-free high beam (GFHB) function and a predictive adaptive frontlighting system (PAFS) for a digital map based headlight swiveling and glare-free driving with maximized high beam time.

Keywords

Glare-Free High Beam (GFHB), Predictive Adaptive Frontlighting System (PAFS), Headlight Control Module (HCM), Driver-in-the-Loop, Visualisation of Headlight Prototypes, High Dynamic Range (HDR) Rendering, Tone Mapping, Virtual Prototyping

1 Einleitung

Moderne Fahrerassistenzsysteme (FAS) unterstützen den Fahrer während der Fahrt durch Hinweise zur aktuellen Fahrsituation oder durch teilweise Übernahme der Fahrzeugsteuerung. Der Eingriff in die Fahrzeugsteuerung ist sicherheitskritisch und erfordert ein intensives Testen der Funktionen des FAS. Ziel des Forschungsprojektes „Test und Trainingsumgebung für fortgeschrittene Fahrerassistenzsysteme“ (TRAFFIS) ist die Unterstützung des Virtual Prototyping solcher FAS auf der Basis eines Fahrsimulators. Hierfür wurde im Rahmen des Projektes ein rekonfigurierbarer Fahrsimulator entwickelt, der gezielt für das Virtual Prototyping von FAS konfiguriert werden kann. Eine spezielle Klasse von FAS sind lichtbasierte FAS. Diese unterstützen den Fahrer während der Fahrt bei Nacht, indem die Scheinwerfer den Straßenraum vor dem Fahrzeug situationsgerecht ausleuchten. Die lichtbasierten FAS verknüpfen dabei Informationen der Umfeldsensoren im Fahrzeug mit elektronischen Datenbasen (e-horizon), um die Scheinwerfer abhängig von der jeweiligen Fahrsituation optimal auf die Straße auszurichten. Für den Test des FAS im Simulator muss das Visualisierungssystem des Fahrsimulators neben der Darstellung des Scheinwerferlichtes auch das virtuelle Modell der realen Versuchsstrecke und die nächtlichen Lichtverhältnisse realitätsnah im Simulator wiedergeben. Dies ist notwendig, um das lichtbasierte FAS nicht nur im Fahrsimulator auf der virtuellen Versuchsstrecke zu testen, sondern im Vergleich dazu auch unter realen Versuchsbedingungen auf der echten Versuchsstrecke. Da das FAS den menschlichen Fahrer während der Fahrt unterstützt, ist das Zusammenspiel des Fahrers im Simulator mit dem simulierten FAS ausschlaggebend. Für solche Driver-in-the-Loop-Untersuchungen ist eine realitätsnahe Darstellung der realen Fahrumgebung für den Testfahrer im Simulator unerlässlich.

Der folgende Abschnitt gibt zunächst einen kurzen Überblick über bestehende Ansätze zur Scheinwerfersimulation für virtuelle Nachtfahrten. Es folgt eine kurze Darstellung lichtbasierter Fahrerassistenzsysteme und deren Funktionen zum vorausschauenden Kurvenlicht bzw. blendfreies Fernlicht. Der nächste Abschnitt stellt die wesentlichen Bestandteile des Visualisierungssystems des TRAFFIS-Fahrsimulators, die High Dynamik Range (HDR)-Darstellung der Scheinwerferlichtverteilung in der nächtlichen Szenerie, sowie die Aufbereitung der visuellen Präsentation für die 8-Kanal-Rundprojektion vor. Der vorletzte Abschnitt beschreibt den funktionalen Nachweis anhand der Driver-in-the-Loop Umsetzung zur Anbindung des Scheinwerfersteuergerätes im Rahmen des Innovationsprojektes „ADAS-Produktentwicklung“ innerhalb des Projektes TRAFFIS, gefolgt von einem kurzen Resümee.

2 Scheinwerfersimulation mit virtuellen Nachtfahrten

Fahrsimulatoren sind im Rahmen der Fahrzeugentwicklung etablierte Werkzeuge. Neben den klassischen Anwendungsbereichen für Fahrsimulatoren, wie die Fahrzeugdynamik, entwickeln sich zunehmend weitere Anwendungsgebiete, wie z.B. die Scheinwerfersimulation zur Ausleuchtung des Straßenraums bei Nacht. Hier existieren eine Reihe von Anwendungen in der Industrie [LS02][Ren01-ol][WPK+02].

Ausgangsbasis für die Darstellung des Scheinwerferlichtes im Rahmen einer Nachtfahrsimulation sind die Lichtverteilungsdatensätze eines Scheinwerferprototypen. Diese stellen im Wesentlichen einen Querschnitt durch den Lichtkegel des Scheinwerfers dar und definieren damit sämtliche Leuchteigenschaften des Scheinwerfers (siehe Bild 1). Lichtverteilungen werden entweder durch eine optische Vermessung eines realen Scheinwerfers ermittelt oder durch Rückverfolgung der Lichtstrahlen (engl. ray tracing) auf Basis des 3D-CAD-Modells des Scheinwerferprototypen mit speziellen Simulationswerkzeugen berechnet. Diese Lichtverteilungsdatensätze werden im Nachtfahrsimulator in die virtuelle Szene projiziert. Eine detaillierte Beschreibung dieses Verfahren gibt [Ber05]. Bisher verwenden Nachtfahrsimulatoren meist einfache Farbzuweisungstechniken (engl. color mapping) und ein konventionelles Beleuchtungsmodell mit einer begrenzten Dynamikbandbreite (vgl. Low Dynamic Range LDR-Rendering). Dieser eher geringe Kontrastumfang führt jedoch zu einem Verlust feiner Details bei der Darstellung der Leuchteigenschaften des simulierten Scheinwerfers.

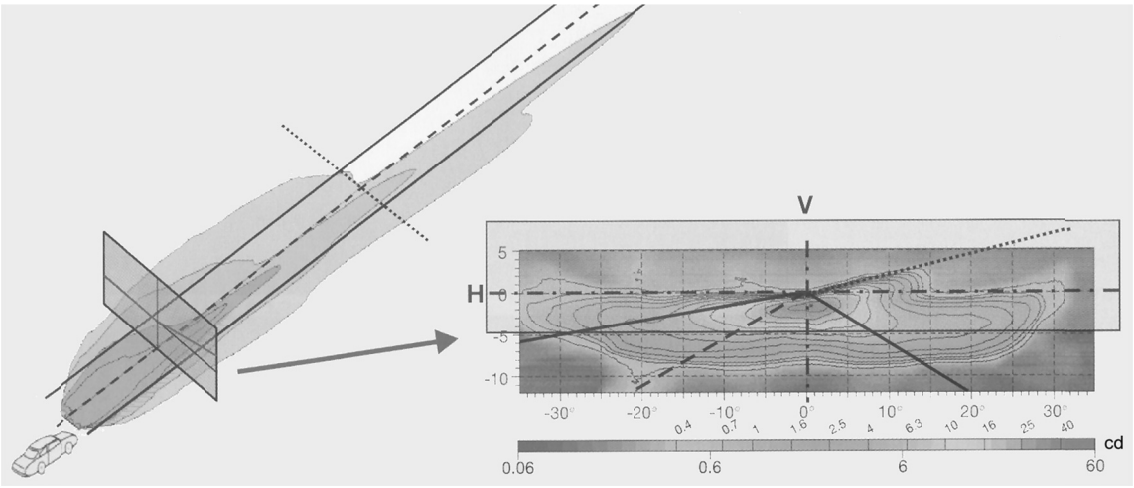


Bild 1: Lichtverteilungsdatensatz stellt einen Querschnitt durch den Lichtkegel dar und definiert damit sämtliche Leuchtcharakteristika des Scheinwerfers (Quelle: Hella)

Typische Lichtverteilungsdatensätze weisen einen Dynamikbereich von bis zu fünf Größenordnungen auf. Dies führt zu sehr kontrastreichen Szenen bei der Simulation der Ausleuchtung durch die Scheinwerfer. High Dynamic Range (HDR) Rendering ist ein sich schnell entwickelnder Bereich in der Computergraphik zur Simulation von sehr kontrastreichen Szenen. HDR-Rendering bietet die erforderliche Genauigkeit für eine

Anpassung von Kontrast und Helligkeit, um alle Aspekte einer Scheinwerferlichtverteilung für eine Analyse in der geforderter Güte darstellen zu können. Eine Beleuchtungssimulation auf der Basis eines HDR-Rendering ist in der Lage, auch feinste Details einer Scheinwerferlichtverteilung wiederzugeben, die bei einem herkömmlichen LDR-Rendering verloren gehen würden. Da beim HDR-Rendering der Kontrastumfang dem der realen Welt entspricht, muss jedes Einzelbild zur Darstellung an den gerätespezifischen Kontrastumfang der verwendeten Darstellungssystems angepasst werden. Dies erfolgt mittels sogenanntem Tone Mapping [Ash02]. Das sind Algorithmen, die den hohen Kontrastumfang auf einen darstellbaren Kontrastbereich abbilden. Die Literatur unterscheidet zischen globalen und lokalen Operatoren [RWP+06].

Globale Operatoren komprimieren die Helligkeit jedes Pixels unabhängig von seiner Umgebung im darzustellenden Bild. Globale Operatoren sind daher sehr effizient berechenbar und hochgradig parallelisierbar. Diese hohe Leistungsfähigkeit bedingt allerdings einen Verlust von Details insbesondere in Bildregionen mit einem hohen Kontrast.

Lokale Operatoren passen die Kompression jedes Pixels an seine lokale Bildumgebung an [RSS+02]. Durch Berücksichtigung der lokalen Eigenschaften der Bildumgebung jeden Pixels bleiben signifikant mehr Szenendetails erhalten. Die individuelle Überprüfung der direkten Bildumgebung eines jeden Pixels während der Kontrastanpassung ist jedoch sehr rechenintensiv. Lokale Operatoren eigenen sich daher nicht für interaktive Anwendungen. Die überragende Fähigkeit Szenendetails zu erhalten macht lokale Operatoren jedoch interessant für die Analyse der Leuchteigenschaften von Scheinwerferprototypen (siehe Bild 2) [KMS05].

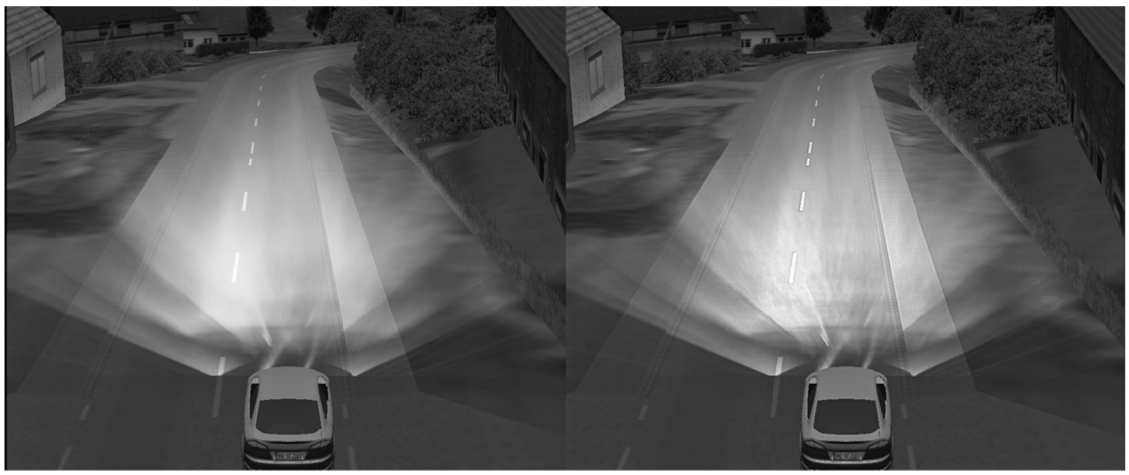


Bild 2: Detailtreue mit globalem (links) und lokalem (rechts) Tone Mapping bei der HDR-Darstellung der Straßenausleuchtung durch die Scheinwerfer nach [BB13].

Mit der Entwicklung der programmierbaren Shader-Technologie auf der GPU konnte die Darstellungsqualität und -geschwindigkeit der Scheinwerferlichtbilder und die Ausleuchtung des Straßenraums bei Nacht deutlich gesteigert werden und ermöglicht auch

detaillierte Visualisierungen der Scheinwerferlichtverteilungen in komplexen 3D-Szenen in Echtzeit. Auf Basis dieser programmierbaren GPUs wurden zudem effiziente Algorithmen für GPU-basiertes Tone-Mapping für die Echtzeitdarstellung von HDR-Scheinwerferdatensätzen entwickelt, welche die Ausleuchtungsdetails der Scheinwerferlichtkegel erhalten und in Echtzeit im Fahrsimulator darstellen [BB13].

3 Lichtbasierte Fahrerassistenzsysteme

Eine spezielle Klasse von FAS sind lichtbasierte FAS. Diese unterstützen den Fahrer während der Fahrt bei Nacht, indem die Scheinwerfer den Straßenraum vor dem Fahrzeug situationsgerecht ausleuchten. Das Steuergerät dieser lichtbasierten FAS verknüpfen dabei Informationen der Umfeldsensorik im Fahrzeug mit elektronischen Datenbasen (e-horizon), um die Scheinwerfer und deren Lichtverteilung abhängig von der jeweiligen Fahr- und Verkehrssituation optimal auf die Straße auszurichten. Das im Rahmen des Projekts TRAFFIS verwendete lichtbasierte FAS kombiniert die Lichtfunktionen vorausschauendes Kurvenlicht (Predictive Adaptive Frontlighting, PAFS) und kamerabasiertes blendfreies Fernlicht (Glare-Free High Beam, GFHB).

Das lichtbasierte FAS basiert auf horizontal und vertikal schwenkbaren Scheinwerfern, welche unterschiedliche Ausleuchtungsfunktionen übernehmen und damit die Kurvenausleuchtung verbessern. Die Funktion Adaptive Frontlighting System (AFS) arbeitet dabei auf Basis des Lenkwinkels, während das Predictive Adaptive Frontlighting System (PAFS) Informationen aus der digitalen Straßenkarte des Navigationssystems (e-horizon) mit der aktuellen Position und Geschwindigkeit des Fahrzeugs verknüpft und so ein vorausschauendes und damit frühzeitiges Einschwenken der Scheinwerfer in eine nahende Kurve ermöglicht [ISK05][BBG06]. Die Dynamic Headlamp Leveling Funktion unterstützt zudem das vertikale Ausrichten des Scheinwerfers gemäß der fahrdynamischen Situation des Fahrzeugs und des vertikalen Straßenverlaufs [BKG10].

Die Funktion Glare-Free High Beam (GFHB) basiert auf einem Kamerasystem zur Erkennung von Gegenverkehr sowie vorausfahrenden Fahrzeugen und verwendet diese Daten (U-Shape um die erkannten Objekte) für die Aussparung des Verkehrs aus dem Fernlicht durch spezielle Lichtverteilungen (L-Shape und Flat-Beam). Dadurch entsteht ein adaptiver Schattenbereich (siehe Bild 3), der abhängig von der Position des Verkehrs links, in der Mitte oder rechts vor dem Fahrzeug generiert wird.

Dieser Schattenbereich ist ein abgedunkelter Bereich im Scheinwerferlichtkegel, der durch die Kombination verschiedener Lichtmodule des Scheinwerfers (vgl. Bild 4) erzeugt wird und durch vertikales und horizontales Schwenken der Lichtmodule genau auf den Gegenverkehr oder vorausfahrende Fahrzeuge ausgerichtet wird, um so blendfreies Fahren bei unverminderter Ausleuchtung des Straßenraums zu realisieren.

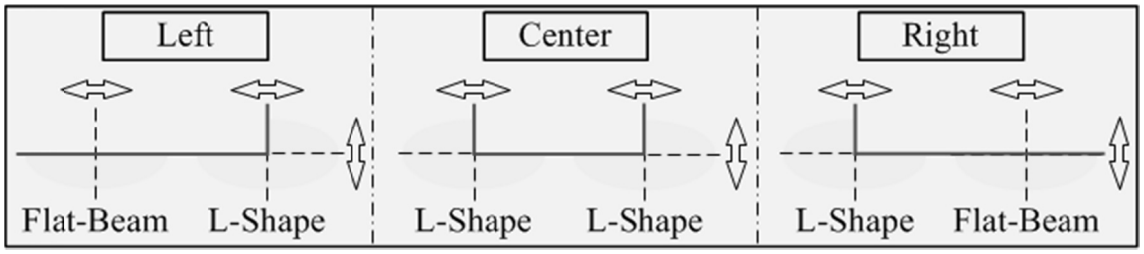


Bild 3: Adaptiver Schattenbereich im Lichtkegel zur Aussparung von Objekten links, in der Mitte oder rechts vor dem Fahrzeug (Quelle: Varroc Lighting Systems)

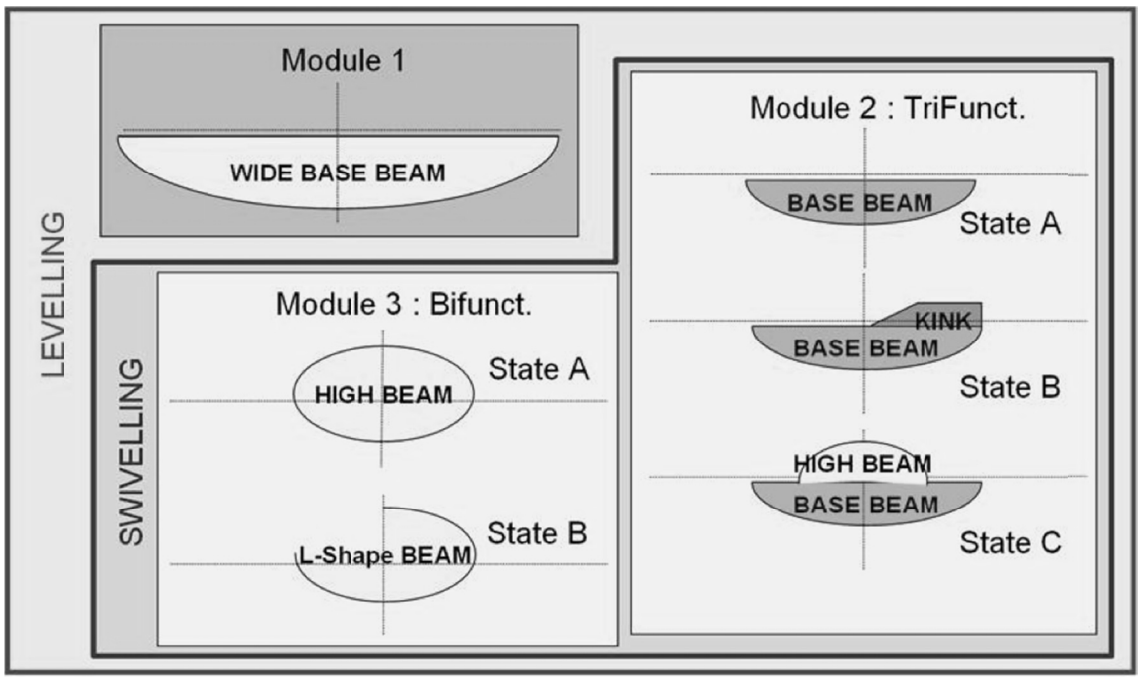


Bild 4: Vertikal und horizontal schwenkbare Lichtmodule eines Scheinwerfers mit GFHB und PAFS-Funktionalität (Quelle: Varroc Lighting Systems)

4 Visualisierung im Rahmen von TRAFFIS

Für den Test des lichtbasierten FAS muss das Visualisierungssystem neben der Darstellung des Scheinwerferlichtes auch das virtuelle Modell der realen Versuchsstrecke und die nächtlichen Lichtverhältnisse realitätsnah im Fahrsimulator wiedergeben. Dies ist notwendig, um das lichtbasierte FAS nicht nur im Fahrsimulator auf der virtuellen Versuchsstrecke zu testen, sondern im Vergleich dazu auch unter realen Versuchsbedingungen auf der echten Versuchsstrecke.

Dafür ist im Rahmen des TRAFFIS-Projektes ein Verfahren entwickelt worden, daß eine weitgehend automatisierte Generierung von virtuellen Teststrecken auf der Basis von verfügbaren digitalen Gelände- und Landschaftsmodellen, Luftbildern, Navigationsdaten u.a. ermöglicht [Kre12] (siehe Bild 5).



Bild 5: Mittlerer Streckenabschnitt der Versuchsstrecke „Eifel“ nach [Kre12] (links) und mit kartenbasierter authentischer Vegetation nach [BSK+14] (rechts).

Das Verfahren wurde im Rahmen des Projektes um eine prozedurale Generierung von authentischer Vegetation erweitert [BSK+14]. Die Abbildung authentischer Vegetation entlang der Teststrecke ist ausschlaggebend für die Ausleuchtung des Straßenraums und insbesondere für die Einsichtbarkeit kurviger Streckenabschnitte (Bild 6).



Bild 6: Verringerte Einsichtbarkeit entlang der Versuchsstrecke durch prozedural generierte authentische Vegetation entlang der Virtuellen Versuchsstrecke (Quelle: Heinz Nixdorf Institut)

Für den Test und die Entwicklung von FAS werden zunehmend Fahrsimulatoren eingesetzt, mit dem Ziel, die zahlreichen erforderlichen Versuchsfahrten von der realen Teststrecke auf eine virtuelle Teststrecke im Simulator zu verlegen und damit reproduzierbare Versuchsbedingungen zu ermöglichen. Dabei ergeben sich für die verschiedenen FAS ganz unterschiedliche Anforderungen an einen Fahrsimulator, der den Test und die Entwicklung dieser Systeme unterstützen soll. Hierfür wurde im Rahmen des Projekts TRAFFIS ein Rahmenwerk zur Entwicklung von rekonfigurierbaren Fahrsimulatoren entwickelt. Das Rahmenwerk beinhaltet eine Methode, welche es ermöglicht, schnell und weitgehend ohne Expertenwissen nach dem Prinzip des morphologischen Kastens

aus einzelnen Basiskomponenten einen anwendungsspezifischen Fahrsimulator entsprechend den jeweiligen Anforderungen maßzuschneidern [Has14].

Im Rahmen des TRAFFIS-Innovationsprojektes „ADAS-Produktentwicklung“ wurde zur Unterstützung der Entwicklung eines Steuergerätes (Headlamp Control Module – HCM) für blendfreies Fahren mit einer Kombination der Lichtfunktionen GFHB und PAFS eine Simulatorvariante entwickelt, welche die speziellen Anforderungen aus der Lichtentwicklung an das Virtual Prototyping von Scheinwerfern und damit an den eingesetzen Nachtfahrsimulator erfüllt:

Projektionssystem: Der Sehsinn nimmt über 80 % der Sinneseindrücke auf und ist damit der wichtigste Sinn des Menschen. Somit ist die visuelle Simulation die wichtigste Komponente des Fahrsimulators zur Vermittlung der simulierten Fahrt an den Testfahrer. Dementsprechend ist der Fahrsimulator mit einer 240°-Rundprojektion ausgestattet, welche das gesamte Sichtfeld des Testfahrers abdeckt, insbesondere das für die Simulation von Kurvenlicht maßgebliche laterale Sehen (Bild 7). Der Fahrer sitzt im Zentrum des Projektionssystems, so daß gleiche Abstände zur Projektion in alle Blickrichtungen gewährleistet sind und das menschliche Auge somit keine unnötige Fokussierungsleistung erbringen muß. Hochauflösende Projektoren projizieren die acht einzelnen Bildsegmente auf die Rundprojektion. Dabei erfolgt die geometrische Verzerrung jedes einzelnen planen Bildsegmentes auf das entsprechende Segment der Rundprojektion auf der Basis von Vertex Shadern in den Render-Knoten. Die vorverzerrten Bildsegmente werden mittels Edge-Blending zu einem Gesamtbild von ca. 1400 mal 8000 Bildpunkten mit einer Auflösung von ca. einem Pixel pro mm^2 zusammengesetzt. Die Sicht nach Hinten wird über einen Rück- und zwei Seitenspiegel sichergestellt (Bild 8).



Bild 7: Rundprojektionssystem im TRAFFIS-Fahrsimulator zur Darstellung der Nachtfahrt aus Fahrersicht (Quelle: Heinz Nixdorf Institut)



Bild 8: Rück- und Seitenspiegel im TRAFFIS-Fahrsimulator erweitern die Sicht nach Hinten und unterstützen bei Überhol- und Rangiermanövern. (Quelle: Heinz Nixdorf Institut)

Visuelle Darstellung: Der Fahrsimulator muß insbesondere die nächtlichen Licht- und Sichtverhältnisse während einer simulierten Nachtfahrt realitätsnah wiedergeben. Dabei werden die Lichtverteilungen der Scheinwerfer auf die Straße in der nächtlichen Szenerie projiziert. Aufgrund der großen Helligkeitsunterschiede der Scheinwerferlichtverteilungen von bis zu fünf Größenordnungen weist auch die ausgeleuchtete nächtliche Szenerie einen entsprechend großen Kontrastumfang auf. Die realitätsnahe Darstellung der Ausleuchtung des Straßenraums im Nachtfahrsimulator erfordert Beleuchtungsmodelle, welche den großen Kontrastumfang der nächtlichen Szenerie sehr detailliert wiedergeben. Die Basis hierfür bilden die zuvor genannten High Dynamic Range (HDR)-Beleuchtungsmodelle, die den großen Dynamikbereich des Scheinwerferlichtkegels in der nächtlichen Szenerie abdecken und damit selbst feinste Kontrastunterschiede in der Ausleuchtung des Straßenraums vor dem Fahrzeug auflösen und detailliert abbilden.

Da HDR-Beleuchtungsmodelle einen Helligkeitsumfang abdecken, der den realen Lichtverhältnissen entspricht, können diese Beleuchtungsmodelle nur sehr ungenau durch das Projektionssystem des Fahrsimulators abgebildet werden, da dieses den notwendigen Kontrastumfang nicht wiedergeben kann. Für das Rendering von HDR-Szenen wird somit als zusätzlicher Verarbeitungsschritt eine Tonwertanpassung (engl. tone mapping) erforderlich, welche den Kontrastumfang jedes Einzelbildes der visuellen Simulation auf den gerätespezifischen Kontrastumfang des Projektionssystems im Simulator abbildet.

Visuelle Analyse: Von Interesse für die Scheinwerferentwicklung sind die Leuchteigenschaften eines Scheinwerferprototypen, wie beispielweise maximale Leuchtweite, Bereiche gleich heller Ausleuchtung oder die Ausleuchtung von Szenendetails in Nah- und Fernbereich des Scheinwerfers. Virtual Prototyping-Werkzeuge, die eine interaktive Scheinwerfersimulation ermöglichen, wie der im Rahmen von TRAFFIS entwickelte Nachtfahrsimulator, müssen diese detaillierten, sehr kontrastreichen virtuellen Szenen in Echtzeit mit mindestens 60 Hz darstellen. Dieses ist besonders herausfordernd, wenn die Szenen auch noch mit der sehr hohen Auflösung der Rundprojektion des TRAFFIS-Fahrsimulators dargestellt werden.

5 Driver-in-the-Loop-Test von FAS im TRAFFIS-Simulator

In Kooperation mit unseren Projektpartnern von dSPACE und Varroc wurde im Rahmen der TRAFFIS-Innovationsprojekte „Closed-Loop-Testwerkzeug“ und „ADAS-Produktentwicklung“ eine Driver-in-the-Loop (DIL)-Testumgebung für die Entwicklung und den Test der GFHB- und PAFS-Funktionen auf der Basis des rekonfigurierbaren Fahrsimulators entworfen und prototypisch implementiert [AS14]. Ziel ist, dem Testfahrer die Wirkungsweise des Steueralgorithmus auf das Scheinwerfer-Modul mit den Funktionen GFHB zum blendfreien Fahren und PAFS für das vorausschauende Kurvenlicht und die daraus resultierende Ausleuchtung des Straßenraums durch die Scheinwerfer in Gegenverkehrssituationen während einer virtuellen Nachtfahrt im Simulator zu präsentieren.

Bei der Untersuchung der Funktionalität des lichtbasierten FAS ist das Zusammenwirken von Fahrer und FAS entscheidend für die Akzeptanz des lichtbasierten FAS durch den Fahrer. Dafür wird die HIL-Simulation des FAS für die Funktionen GFHB und PAFS an den TRAFFIS-Fahrsimulator angebunden und der Fahrer somit in die geschlossene Regelschleife integriert. Dies führt zu dem Driver-in-the-Loop (DIL)-System.

Der TRAFFIS-Fahrsimulator umfaßt eine Bewegungsplattform mit fünf Freiheitsgraden, auf der die Smart-Fahrerkabine vor dem Rundprojektionssystem aufgebaut ist. Durch die Simulation der Fahrzeugbewegungen auf der Bewegungsplattform und die Darstellung der Sicht aus dem Fahrzeugcockpit auf die Rundprojektion erhält der Fahrer einen realitätsnahen Eindruck der simulierten Nachtfahrt.

Für die Simulation des FAS für Blendfreies Fahren wird das Steuergerät des Scheinwerfers an das HIL-System zur Ansteuerung der Fahrzeugbewegungen angeschlossen. Aus der HIL-Simulation werden die aktuellen Fahrdaten an das Steuergerät übertragen. Der Steueralgorithmus für das blendfreie Fahren und das vorausschauende Kurvenlicht schwenkt und steuert die Lichtfunktionen entsprechend. Auf der Rundprojektion wird dem Fahrer die resultierende Ausleuchtung des Straßenraums gemäß der Ansteuerung der Scheinwerfer, der aktuellen Fahrsituation sowie der Umgebung präsentiert.

Auf diese Weise kann die Funktion des Steuergerätes getestet und das Zusammenwirken des Fahrer und des lichtbasierten FAS in zahlreichen unterschiedlichen Fahrszenarien bewertet werden (Bild 9). Hierfür wurden im Rahmen von TRAFFIS verschiedene Versuchstrecken virtuell nachgebildet und in den Simulator integriert.

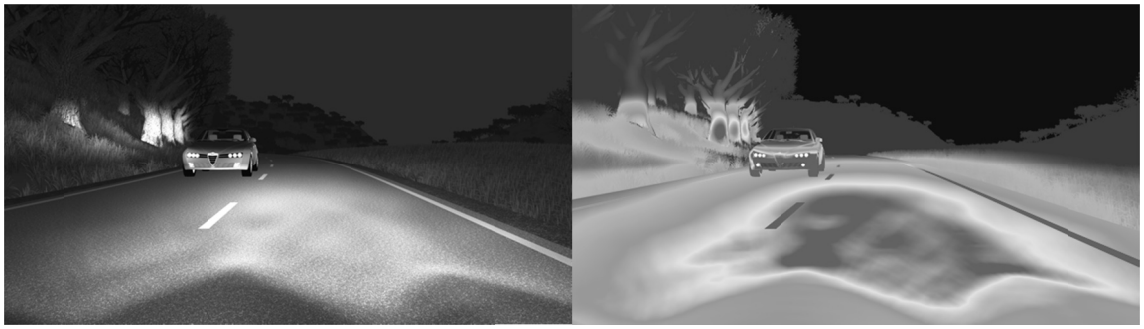


Bild 9: Visualisierung von blendfreiem Fahren bei Nacht: Ansteuerung des GFHB-Scheinwerfer durch den Algorithmus des HCM und damit Ausrichtung des U-Shape (Aussparung des Lichtkegels im Bereich des Gegenverkehrs) zur Blendungsvermeidung des Gegenverkehrs. (Quelle: Heinz Nixdorf Institut und Varroc Lighting Systems)

Neben der Präsentation des lichtbasierten FAS für den Fahrer im Simulator zur Untersuchung der Wechselwirkung von Fahrer und Assistenzsystem sowie der Akzeptanz des Systems durch den Testfahrer ermöglicht der Fahrsimulator auch die Optimierung des Steueralgorithmus des FAS durch den Entwicklungsingenieur. Durch die Visualisierung der Ausleuchtung durch die Scheinwerfer ermöglicht die VR-Umgebung im Simulator das schnelle Einstellen und Wiedergeben unterschiedlicher Kameraperspektiven, Visualisierungsparameter und Verkehrssituationen, in denen dann das Verhalten des Steueralgorithmus untersucht und angepaßt werden kann (Bild 10). Im Gegensatz zu herkömmlichen schematischen 2D-Darstellungen der Scheinwerferlichtkegel bei klassischen HIL-Simulationen basiert die Untersuchung auf der qualitativ hochwertigen und anschaulichen Visualisierung der Scheinwerferlichtkegel sowie der ausgeleuchteten nächtlichen Straßenszenerie.

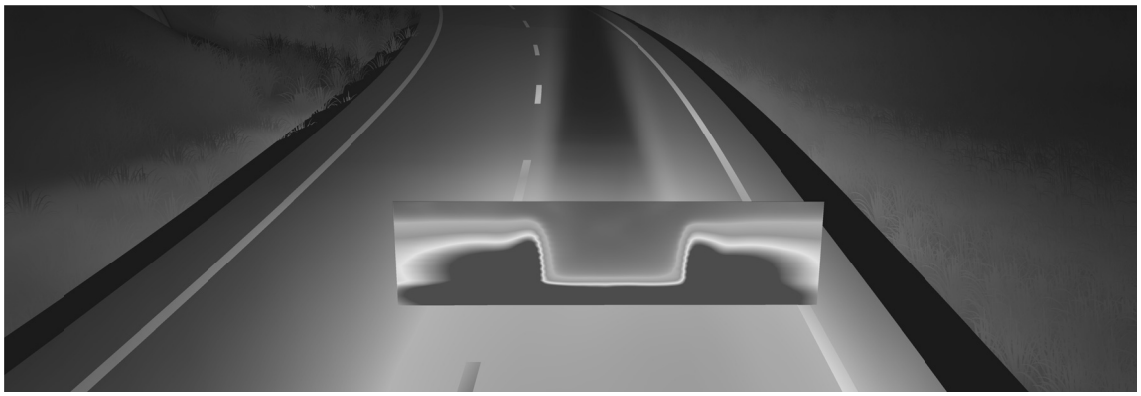


Bild 10: Visualisierung des Schattenbereichs für blendfreies Fahren der Scheinwerfer mit GFHB-Funktion zur Anpassung und Optimierung der Ausrichtung auf vorausfahrende Fahrzeuge oder Gegenverkehr. (Quelle: Heinz Nixdorf Institut)

6 Resümee und Ausblick

Der Beitrag stellt das Virtual Prototyping eines lichtbasierten Fahrerassistenzsystems zum blendfreien Fahren mit Fernlicht anhand eines Driver-in-the-Loop-Ansatzes auf der Basis des rekonfigurierbaren Fahrsimulators des Forschungsprojektes TRAFFIS vor. Der Driver-in-the-Loop-Ansatz wurde ermöglicht durch die qualitativ hochwertige Darstellung des nächtlichen Fahrszenarios im Simulator, das realitätsnahe HDR-Rendering der komplexen und sehr kontrastreichen Scheinwerferlichtbilder und die Integration realistischer Versuchsstrecken. Diese hochwertige Visualisierung unterstützt nicht nur die Untersuchung des Zusammenwirkens von Fahrer und Assistenzsystem während der simulierten Fahrt, sondern ermöglicht erst die Optimierung des Algorithmus zur Ansteuerung der Scheinwerfer in dem angebundenen Steuergerät des lichtbasierten Fahrerassistenzsystems.

Danksagung

Diese Arbeit ist Teil des Forschungsprojekts „Test und Trainingsumgebung für fortgeschrittene Fahrerassistenzsysteme“ (TRAFFIS) und wird gefördert durch den Regionalen Entwicklungsfond (ERDF) der Europäische Union und dem Ministerium für Wirtschaft, Energie, Industrie, Handel und Handwerk des Bundeslandes Nordrheinwestfalen im Rahmen des „Ziel2“-Programms.

Wir danken unseren Projektpartnern von Varroc Lighting Systems, dSPACE und dem Institut für Logistik & Verkehr OWL für die gute Zusammenarbeit im Rahmen des Projektes TRAFFIS.

Literatur

[AS14] AMELUNXEN, H.; SCHMID, C.: Driver-in-the-Loop Simulation for Advanced Driver Assistance Systems. 12. International Symposium on Advanced Vehicle Control (AVEC), Tokyo, Japan, 22.-26. September 2014.

[Ash02] ASHIKHMIN, M.: A Tone Mapping Algorithm for High Contrast Images. Proceedings of Thirteenth Eurographics Workshop on Rendering, 2002, S. 145–155.

[BB13] BERSSENBRÜGGE, J.; BONNER, E.: GPU-Based Local Tone Mapping in the Context of Virtual Night Driving. In: Journal of Computing and Information Science in Engineering, Band 13(2), 2 Park Avenue, New York, NY 10016, Jun. 2013, ASME

[BBG06] BERSSENBRÜGGE, J.; BAUCH, J.; GAUSEMEIER, J.: A Virtual Reality-based Night Drive Simulator for the Evaluation of a Predictive Advanced Front Lighting System. Proceedings of the ASME IDETC & CIE, Philadelphia, PY, USA, 2006.

[Ber05] BERSSENBRÜGGE, J.: Virtual Nightdrive – Ein Verfahren zur Darstellung der komplexen Lichtverteilungen moderner Scheinwerfersysteme im Rahmen einer virtuellen Nachtfahrt. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 177, 2005.

[BKG10] BERSSENBRÜGGE, J.; KREFT, S.; GAUSEMEIER, J.: Virtual Prototyping of an Advanced Leveling Light System Using a Virtual Reality-Based Night Drive Simulator. Journal of Computing and Information Science in Engineering, Vol 10(2), June 2010.

[BSK+14] BERSSENBRÜGGE, J.; STÖCKLEIN, J.; KOZA, A.; GRÄßLER, I.: Procedural Generation of Vegetation for a Virtual Test Track. In: Proceedings of the ASME IDETC & CIE, Buffalo, NY, USA, 17.-20. Aug. 2014.

[Has14] HASSAN, B.: A Design Framework for Developing a Reconfigurable Driving Simulator. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 333, 2014.

[ISK05] IBRAHIM, F.; SCHMIDT, C.; KLEIN, M.: Predictive Advanced Front Lighting System, Proceedings of the ISAL 2005 Symposium, Darmstadt University of Technology, Darmstadt, Germany, 2005.

[KMS05] KRAWCZYK, G.; MYSZKOWSKI, K.; SEIDEL, H.-P.: Perceptual Effects in Real-time Tone Mapping. Proceedings of the 21st Spring Conference on Computer Graphics (SCCG), ACM, 2005, S. 195-202.

[Kre12] KREFT, S.: Systematik zur effizienten Bildung von geospezifischer Umgebungsmodelle für Fahrsimulationen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 305, 2012.

[LS02] LÖWENAU, J. P.; STROBL, M. H.: Advanced Lighting Simulation (ALS) for the Evaluation of the BMW System Adaptive Light Control (ALC), Technical Paper SAE – 2002- 01-1988, Society of Automotive Engineers International, Warrendale, PA, USA, 2002.

[Ren01-ol] RENAULT TECHNICAL CENTER FOR SIMULATION (HRSG.): Driving Simulation & Virtual Reality – Dynamics Driving Simulation. Unter: http://www.experts.renault.com/kemeny/projects/dynamic_driving_simulator/index.html, last checked: Nov. 2014

[RSS+02] REINHARD, E.; STARK, M.; SHERLEY, P.; FERWERDA, J.: Photographic Tone Reproduction for Digital Images. Proceedings of ACM SIGGRAPH, Vol. 21, ACM Press/Addison-Wesley Publishing Co., 2002, S. 267-276.

[RWP+06] REINHARD, E.; WARD, G.; PATTANAIK, S.; DEBEVEC, P.: High Dynamic Range Imaging - Acquisition, Display and Image-Based Lighting. Morgan Kaufmann, 2006.

[WPK+02] WEBER, T.; PLATTFAUT, C.; KLEINKES, M., BERSSENBRÜGGE, J.: Virtual Nightdrive. In: Proceedings of the Driving Simulation Conference; Paris, Frankreich, 2002.

Autoren

Dr.-Ing. Jan Berssenbrügge ist wissenschaftlicher Mitarbeiter in der Fachgruppe Produktentstehung am Heinz Nixdorf Institut der Universität Paderborn. Seine Forschungsschwerpunkte liegen in den Bereichen Virtual & Augmented Reality, Virtual Engineering sowie Virtual Prototyping und Simulation. Herr Berssenbrügge studierte Informatik an der Universität Paderborn und promovierte am Heinz Nixdorf Institut mit einem Thema zum Virtual Prototyping von Scheinwerfersystemen.

Dipl.-Ing. Christoph Schmidt ist technischer Spezialist für Lichtalgorithmen und simulationsbasierte Softwareentwicklung bei der VARROC Lighting Systems GmbH.

Prof. Dr.-Ing. Ansgar Trächtler hat an der Universität Karlsruhe (TH) Elektrotechnik studiert und 1991 am Institut für Regelungs- und Steuerungssysteme bei Prof. Föllinger promoviert. Von 1992 bis 1998 war er als wissenschaftlicher Assistent am Institut für Meß- und Regelungstechnik (Prof. Mesch) der Universität Karlsruhe tätig, habilitierte sich 2000 und erhielt die Venia Legendi für das Fach „Meß- und Regelungstechnik“.

Für seine wissenschaftlichen Arbeiten erhielt er 1998 den Eugen-Hartmann-Preis der GMA und den Messtechnik-Preis des AHMT (Arbeitskreis der Hochschullehrer für Messtechnik e.V). Von 1998 bis 2004 war er bei der Robert Bosch GmbH tätig und leitete zuletzt den Bereich Vorausentwicklung Fahrwerksysteme. Seit November 2004 ist er Professor und Leiter des Lehrstuhls für Regelungstechnik und Mechatronik der Universität Paderborn. Außerhalb der Universität engagiert sich Prof. Trächtler in der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und in der International Federation of Automatic Control (IFAC).

Entwicklung eines Closed-Loop Testsystems für fortgeschrittene Fahrerassistenzsysteme

Dr.-Ing. Hendrik Amelunxen
dSPACE GmbH
Rathenaustraße 26, 33102 Paderborn
E-Mail: hamelunxen@dspace.de

Zusammenfassung

Für die Hardware-in-the-loop (HIL) Simulation mit fortgeschrittenen Fahrerassistenzsystemen (Advanced Driver Assistance Systems, ADAS), bspw. Spurhalteassistenten, adaptive Scheinwerfersteuergeräte oder Car-to-X-Anwendungen, werden besondere Anforderungen an die Umgebungs- und die Verkehrssimulation gestellt. Im Rahmen des *TRAFFIS*-Forschungsprojektes wurde ein Testsystem für die closed-loop Simulation mit ADAS Steuergeräten aufgesetzt, das zusätzlich mit Hilfe eines Fahrsimulators den Menschen in die Regelschleife mit einbindet. In dieser Arbeit werden zunächst die innerhalb des *TRAFFIS*-Projektes durchgeführten modellseitigen Weiterentwicklungen vorgestellt, die durch die Anforderungen moderner ADA Systeme für die HIL Simulation erforderlich sind. Weiterhin wird das mit einem beispielhaftem ADAS Steuergerät für Scheinwerferanwendungen aufgebaute closed-loop ADAS Testsystem und die Möglichkeiten, die sich dadurch für die Entwicklung und den Test von ADAS ergeben, aufgezeigt.

Schlüsselworte

HIL Simulation, ADAS, Umgebungsmodellierung

Development of a Closed-Loop Test System for Advanced Driver Assistance Systems

Abstract

Hardware-in-the-loop (HIL) simulation of advanced driver assistance systems (ADAS), like lane keeping assists, adaptive front lighting systems or car-to-x applications, demand for specific characteristics and features of the environment and traffic simulation. In the *TRAFFIS* research project a test system for the closed-loop simulation with ADAS controllers has been developed, which also includes the human driver into the control loop by using a complex driving simulator. This paper presents the model modifications developed to comply with the requirements of latest ADA systems for HIL simulations. Furthermore the closed-loop test system with an exemplary ADAS controller for adaptive front lighting applications, and the possibilities for development and test of ADA systems, which evolve from such a tool, are presented.

Keywords

HIL Simulation, ADAS, Environment model

1 Einleitung

Im Entwicklungsprozess von Fahrzeugregelsystemen ist die Hardware-in-the-Loop (HIL) Simulation, bei der das reale elektronische Steuergerät zur Funktionsabsicherung in einer simulierten Umgebung in geschlossener Regelschleife betrieben wird, eine seit Jahren etablierte Technik. Allerdings stößt sie bei den zunehmend im Fahrzeug Einzug haltenden Fahrerassistenzsystemen (Advanced Driver Assistance Systems, ADAS) an Grenzen, da hier neben den rein funktionalen Tests der Steuergerätesoftware auch die Wirkung auf den Fahrer und die Interaktion mit ihm ein bedeutendes Bewertungskriterium darstellt. Als beispielhafte Systeme seien hier Spurhalteassistenten, adaptive Scheinwerfersteuergeräte und Car-to-X-Anwendungen genannt. Bisher werden diese Funktionalitäten zum allergrößten Teil noch im Fahrversuch auf einer Teststrecke und im realen Straßenverkehr erprobt, so dass Simulationstechniken an dieser Stelle ein bedeutendes Einsparpotenzial eröffnen. Hierzu ist es allerdings notwendig, die Interaktion des Fahrers mit den Bedien- und Anzeigeelementen, sowie den kinästhetischen Eindruck unter Verwendung des realen ADAS-Steuergeräts zu berücksichtigen.

Das Heinz Nixdorf Institut der Universität Paderborn hat in Zusammenarbeit mit der dSPACE GmbH, der Varroc Lighting Systems GmbH und weiteren Partnern das Forschungsprojekt *TRAFFIS* (Test- und Trainingsumgebung für fortgeschrittene Fahrerassistenzsysteme, [TRA14-ol]) ins Leben gerufen, bei dem ein Hauptaspekt die Entwicklung einer Simulationsumgebung darstellt, um damit die Möglichkeiten hinsichtlich Entwicklung und Test von ADAS Steuergeräten zu untersuchen.

In dieser Arbeit werden die innerhalb des *TRAFFIS*-Projektes für dieses Closed-Loop Testsystem durchgeführten Entwicklungen vorgestellt, die sich durch die Anforderungen von ADAS-Steuergeräten für die Funktionsabsicherung ergeben. Weiterhin wird der Testaufbau mit einem in geschlossener Regelschleife betriebenen beispielhaften ADAS Steuergerät für Scheinwerfer-Fahrerassistenzfunktionen mit den für diese spezielle Anwendung zusätzlich notwendigen Modellerweiterungen beschrieben.

2 Übersicht

2.1 TRAFFIS-Projekt

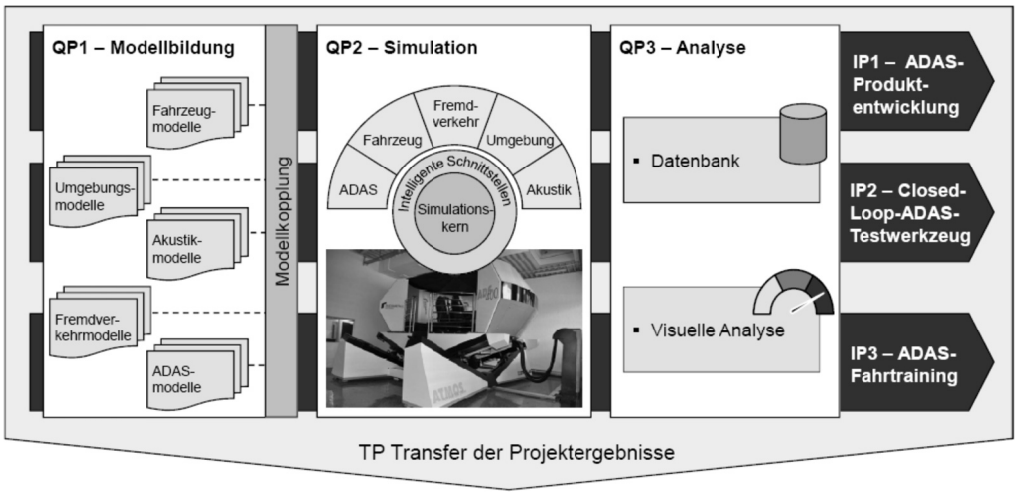


Bild 1: TRAFFIS-Projektstruktur

Innerhalb des *TRAFFIS*-Projektes wurde eine Test- und Trainingsumgebung entwickelt, um auf der einen Seite neue Werkzeuge für die Entwicklung und den Test fortgeschrittener Fahrerassistenzsysteme zu schaffen, auf der anderen Seite soll dieses System dazu genutzt werden, realen Fahrern, bspw. Berufskraftfahrern, die Wirkungsweise und den Nutzen dieser neuartigen Assistenzfunktionen simulativ und effektiv zu vermitteln, [HG13]. Der Kern des Projektes ist ein leistungsfähiger Fahrsimulator mit Bewegungsplattform, der innerhalb des Projektes zur Test- und Trainingsumgebung entwickelt und angewandt wird. Für die Entwicklung sind drei Querschnittsprojekte (QP) vorgehalten, (siehe Bild 1). Innerhalb der Innovationsprojekte (IP) wird das System von den verschiedenen Projektpartnern angewandt und die Projektergebnisse werden im Rahmen eines Transferprojektes (TP) einem breiten Publikum zugänglich gemacht.

Die in dieser Arbeit vorgestellten Ergebnisse zu den Entwicklungen der Simulationsumgebung sind dem Querschnittsprojekt QP1 *Modellbildung* zugeordnet, der Aufbau und die Anwendung des Closed-Loop Testwerkzeugs mit einem beispielhaften ADA System für Scheinwerferassistenzfunktionen beschreiben das Innovationsprojekt IP2 *Closed-Loop-ADAS-Testwerkzeug*.

2.2 Systemübersicht

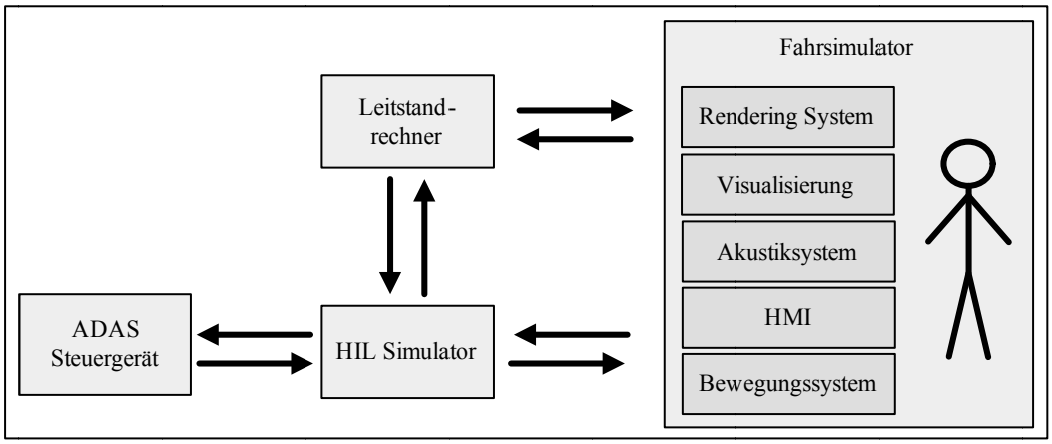


Bild 2: Funktionsstruktur des Closed-Loop Testwerkzeugs

Bild 2 fasst die Funktionsstruktur des Closed-Loop Testsystems zusammen. Der Fahrsimulator bildet die Schnittstelle mit dem Menschen und ist in der Lage, die Sinne des Fahrers visuell, akustisch, haptisch und kinästhetisch zu stimulieren und somit ein realistisches Fahrgefühl zu vermitteln, [ABS14].

Der HIL Simulator nimmt innerhalb des Testwerkzeugs eine zentrale Stellung ein, da dieser sowohl für das angeschlossene ADAS Steuergerät als auch für den Fahrsimulator die benötigten Größen in Echtzeit bereitstellt. Die dSPACE Automotive Simulation Models (ASM) werden für die Berechnung der Fahrdynamik des Ego-Fahrzeugs verwendet und liefern dessen aktuelle Zustände. Diese Zustände können allerdings aufgrund der Kinematik des Bewegungssystems nicht 1:1 umgesetzt werden. Deshalb werden zusätzlich Motion-Cueing Algorithmen gelöst, die die Sollgrößen für die Bewegungssystemaktorik berechnen, um die aktuellen Fahrzustände mit der vorhandenen Kinematik möglichst realistisch abzubilden, [AQT12]. Weiterhin werden Umgebungsverkehrsmodelle der ASM verwendet, um das Bewegungsverhalten weiterer Verkehrsteilnehmer zu berechnen. Die Zustände des Ego-Fahrzeugs und der weiteren Verkehrsteilnehmer werden durch das Rendering- und Projektionssystem umgesetzt, um dem Menschen in dem Fahrsimulator die Fahrsituation zu vermitteln. Die Fahrereingaben zur Steuerung des Fahrzeugs werden durch Sensoren im Fahrsimulator erfasst und in der Simulation entsprechend berücksichtigt.

Der HIL Simulator liefert weiterhin die für ein reales ADAS Steuergerät benötigten Sensordaten in Echtzeit, nimmt die von dem Steuergerät erzeugten Aktorsignale auf und lässt die Wirkung auf das Ego-Fahrzeug in die Simulation mit einfließen. Durch die Kombination des Streckenmodells mit einem spezifischen IO-Modell zur Steuerung der Kommunikation zwischen Simulator und Steuergerät wird die Regelschleife geschlossen und das Steuergerät closed-loop betrieben.

Der Leitstandrechner dient zum einen als Eingriffspunkt zur Steuerung der Simulation. So werden hiermit beispielsweise Ego-Fahrzeug Parameter und Simulationsszenarien

erstellt, verwaltet und auf das Echtzeitmodell geladen, die Simulationen werden gestartet oder angehalten. Zum anderen werden mit dem Leitstandrechner Messungen durchgeführt und Datenauswertung betrieben.

3 Modellentwicklungen für ADAS-Testszenarien

Die ASM Fahrdynamik- und Umgebungsmodelle von dSPACE werden in dem HIL-Simulator für die Generierung der Echtzeit-Simulationsdaten verwendet. Für das im Rahmen des *TRAFFIS*-Projektes aufgebaute ADAS Closed-Loop Testsystem wurden Modellerweiterungen durchgeführt, um ADAS-relevante Testszenarien durchführen zu können.

3.1 Straßennetzwerk

Das Hauptanwendungsgebiet für herkömmliche Fahrdynamikmodelle liegt in der Funktionsabsicherung von Fahrdynamikregelsystemen (ESP, EPS, ASR etc.) mittels HIL-Simulation, ein in der Automobilindustrie anerkanntes Standardverfahren, bei dem das reale Steuergerät und das simulierte Streckenmodell (in diesem Fall das Fahrzeug) in geschlossener Regelschleife in Echtzeit betrieben wird. Anhand geeigneter Fahrmanöver (Stationäre Kreisfahrt, Sinuslenken, µ-split Bremsung etc.) wird die Funktionalität des Steuergeräts unter variierenden Umgebungsbedingungen und mit verschiedenen Fahrzeugkonfigurationen simulativ abgesichert. Für die HIL-Simulation mit den zunehmend in Fahrzeugen Einzug haltenden ADA Systemen, beispielsweise Fahrspurassistenten, Notbremsassistenten, Car-to-X Systemen, werden allerdings andere Anforderungen an das Umgebungsmodell gestellt, da hier nicht mehr das reine Fahrverhalten eines Fahrzeugs simuliert werden muss, sondern zunehmend komplexe Szenarien relevant sind, in denen weitere Verkehrsteilnehmer in unterschiedlichen Situationen involviert sind. Aus diesem Grund wurde das Umgebungsmodell für Fahrdynamikanwendungen derart modifiziert, dass auch mehrere Straßen und mehrere Kreuzungen definiert und in der Simulation verwendet werden können, um so den Übergang vom Fahrdynamik-Anwendungsfall (Manöver auf einer Teststrecke) hin zu ADAS-Anwendungsfällen (urbane Verkehrsszenarien in einem komplexen Straßennetzwerk) zu vollziehen.

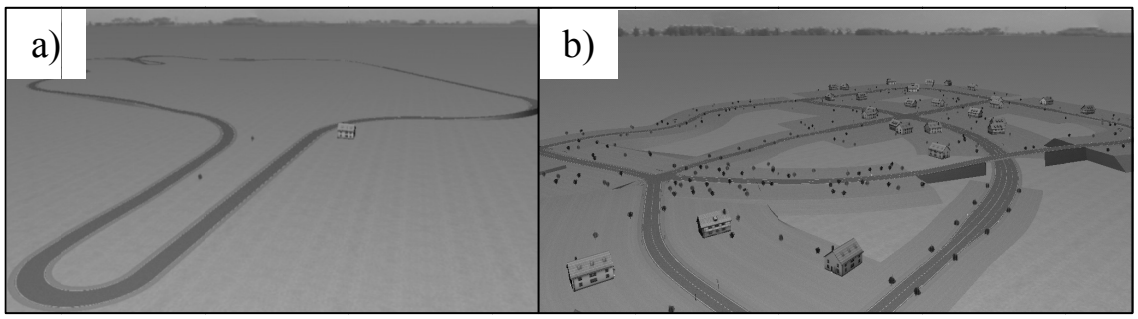


Bild 3: a) Umgebungsmodell für Fahrdynamikanwendungen, b) ADAS-Anwendungen

Bild 3 verdeutlicht den Unterschied zwischen dem ursprünglichen Umgebungsmodell für Fahrdynamikanwendungen (a) und ADAS-Anwendungen (b). Für den Fahrdynamikfall ist die Nachbildung von Teststrecken mit Definition von Reibbereichen, Höhen- und Querneigungsprofilen etc. notwendig. Für das Umgebungsmodell für ADAS Anwendungen wurden diese Definitionsmöglichkeiten für Straßen beibehalten; zusätzlich wurde die Möglichkeit geschaffen, eine Mehrzahl von Straßen im Umgebungsmodell zu definieren, und diese über generische Modellierungsobjekte für Kreuzungen zu Straßennetzwerken zu verknüpfen. Auf diese Weise lassen sich neben Standardkreuzungen (Einmündungen, Viererkreuzungen) auch Kreuzungen mit einer beliebigen Anzahl von Einfahrtstraßen, komplexe Kreuzungen mit Abbiegegassen, Kreisverkehre, Autobahnauffahrten etc. modellieren, siehe Bild 4.

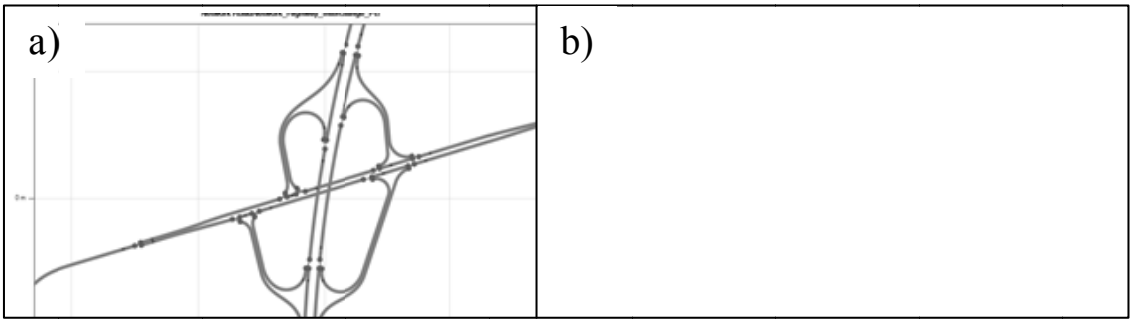


Bild 4: a) Struktur einer Autobahnauffahrt durch Kombination von Straßen- und Kreuzungsobjekten, b) Darstellung in der Animation

Durch die Erweiterung des Umgebungsmodells ist es möglich, Verkehrsszenarien an kritischen Stellen, beispielsweise unübersichtlichen Kreuzungen, Autobahnauffahrten und -abfahrten etc., zu simulieren, also genau dort, wo ADA Systeme zum Einsatz kommen.

Für eine Fahrsimulation auf realen Straßennetzwerken kommt eine in [Kre12] entwickelte und in [Kre13] zusammengefasste Systematik zum Einsatz, die aus gegenwärtig verfügbaren Geodaten effizient ein Umgebungsmodell für die Visualisierung erzeugen kann. Die dabei erzeugte Straßennetzwerkbeschreibung im weitverbreiteten OpenDrive-Format kann durch eine entwickelte Import-Funktionalität der Simulation verfügbar gemacht werden. Auf diese Weise kann mit geringem Aufwand ein Graphikmodell einer realen Umgebung für das Visualisierungssystem des Fahrsimulators generiert und durch die standardisierte Beschreibung der Netzwerkdaten ein konsistentes Logikmodell für die Simulation verwendet werden.

3.2 Autonomer Fremdverkehr

Für die Simulation der Fremdfahrzeugbewegungen wird eine Traffic Erweiterung für die ASM Vehicle Dynamics verwendet. Der Hauptanwendungsfall dieser Modellkombination ist die HIL Simulation, für die exakt definierte, reproduzierbare Verkehrsszena-

rien benötigt werden, um mögliche Steuergerätfehlfunktionen zu erkennen und nachstellen zu können. Dafür werden vielfältige Möglichkeiten bereitgestellt, um sowohl zeit- als auch ereignisbasierte Verkehrsszenarien zu erstellen. Das Bewegungsverhalten der Fremdfahrzeuge wird auf festen Routen im Straßennetzwerk durch Vorgabe Ihrer Position, Geschwindigkeit oder Beschleunigung (absolut oder relativ zu Bezugsobjekten) festgelegt. Die Fremdfahrzeuge verfügen weder über Sensorfunktionalitäten noch über physikalische Eigenschaften, so dass zum einen ein gegenseitiges Erkennen der Fahrzeuge nicht möglich ist, zum anderen können unrealistische Bewegungsmuster definiert werden. Da solch eine a priori Definition des Fremdverkehrbewegungsverhaltens für reguläre HIL Simulationen zwar erforderlich, für die Einbindung eines realen Fahrers in die Regelschleife allerdings ungeeignet ist, wurde ein Verkehrsfahrermodell entwickelt, das den Fremdfahrzeugen ermöglicht, sich auf vorgegebenen Routen entsprechend der Verkehrssituation und der Vorfahrtregelungen eigenständig und plausibel zu bewegen (siehe Bild 5). Die Grundlagen für das Verkehrsfahrermodell wurden in [Abd12] erarbeitet.



Bild 5: Selbstständiges Verhalten der Fremdfahrzeuge an einer beispielhaften, schildergeregelten Kreuzung, unter Berücksichtigung der Vorfahrts- und Verkehrssituation

In dem Verkehrsfahrermodell sind die Fremdfahrzeuge mit Sensorfunktionalitäten ausgestattet, um sowohl andere Verkehrsteilnehmer als auch vorfahrtsregelnde Verkehrszeichen zu erkennen. Weiterhin sind den Fahrzeugen physikalische Eigenschaften zugeordnet, so dass ein realistisches Beschleunigungs- und Verzögerungsverhalten der Fahrzeuge realisiert werden kann. Auf Basis der zusätzlichen Sensorik und der physikalischen Eigenschaften wurden Regeln implementiert, so dass die Fahrzeuge auf Straßen nicht kollidieren (Auffahren auf langsamere Fahrzeuge), und sich an Einmündungen und Kreuzungen, unter Beachtung der vorherrschenden Vorfahrts- und Verkehrssituation, eigenständig plausibel bewegen.

4 Closed-Loop Testsystem mit lichtbasiertem ADAS

In dem *TRAFFIS*-Projekt wird zum Aufbau des Closed-Loop Testsystems als beispielhaftes ADA System ein Steuergerät für Scheinwerferassistenzfunktionalitäten (Headlamp Control Module, HCM) des Projektpartners Varroc Lighting Systems GmbH eingesetzt. Das HCM wird mittels HIL Simulation in geschlossener Regelschleife betrieben. Durch die Ankopplung an den Fahrsimulator und die Umsetzung der Steuergerätstellsignale und Visualisierung der daraus resultierenden Ausleuchtung der Straße, lassen sich die Wirkungsweise und die Qualität der Steuergerätealgorithmen am Simulator beurteilen.

4.1 HCM Funktionalitäten

Das HCM fusioniert zwei auf unterschiedlichen Sensorprinzipien basierenden Scheinwerfer-Fahrerassistenzfunktionen: Das Predictive Adaptive Front Lighting System (PAFS) nutzt Karteninformationen, um das Licht entsprechend der Umgebung anzupassen (urbaner Bereich, Autobahn, Landstraße, etc.) und den vorausliegenden Straßenverlauf vor Kurveneinfahrt auszuleuchten. Die Glare-Free High Beam-Funktionalität (GFHB) basiert auf vorverarbeiteten Daten eines Frontkamerasystems, das andere Verkehrsteilnehmer im Bereich vor dem Fahrzeug detektiert. Anhand der Kamerasignale wird im Fernlicht ein dynamisch adaptierbarer Schattenbereich erzeugt und mit den Bewegungen der Fahrzeuge mitgeführt, so dass andere Verkehrsteilnehmer trotz Fernlicht nicht geblendet werden, [AS14].

4.2 HIL-Simulations Layout

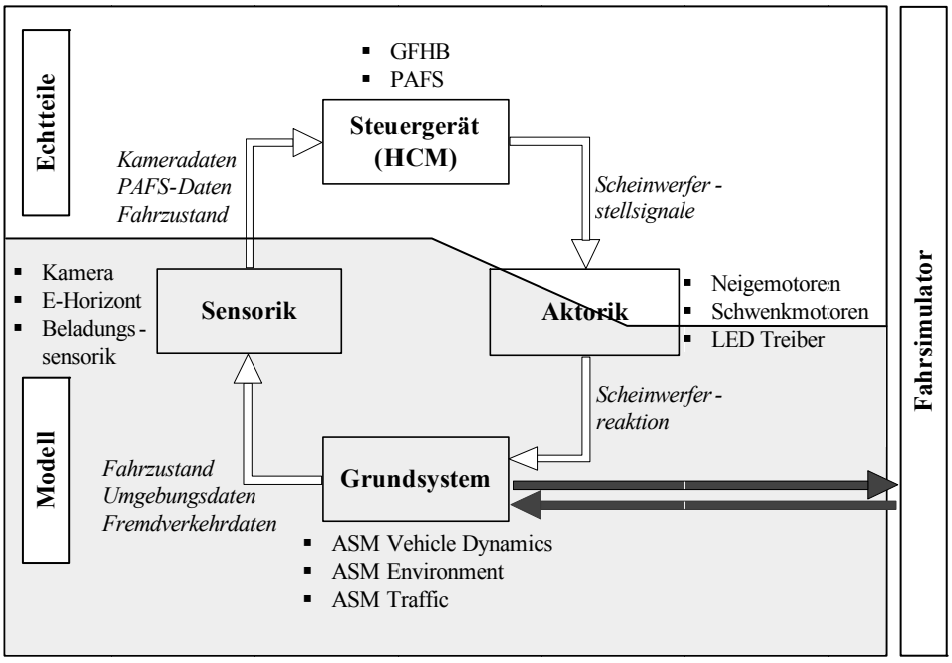


Bild 6: Übersicht HIL Simulation

Die Regelschleife der im *TRAFFIS*-Projekt aufgesetzten HIL Simulation für das HCM samt Unterscheidung zwischen realen und simulierten Komponenten ist in Bild 6

dargestellt. Das HCM sendet Stellsignale an die Scheinwerferaktoren, wobei in diesem Simulationsaufbau die Schwenk- und die Neigemotoren des Projektionssystems als Echtteile ausgeführt sind. Das Grundsystem, dargestellt durch Echtzeitmodelle zur Beschreibung der Fahrdynamik, der Umgebung und des Fremdverkehrs, erhält die Reaktionen der Scheinwerferaktoren und dient als Interface zu der Bewegungsplattform, dem Renderingsystem und dem HMI des Fahrsimulators. Die Kamera, der elektronische Horizont und die analogen Sensoren für die Messung des Beladungszustandes werden simuliert und liefern die vom HCM benötigten Eingangssignale in Echtzeit.

4.3 Sensormodellierung

Die GFHB Funktionalität des HCM benötigt für die Erzeugung des dynamischen Blendbereichs im Fernlicht speziell aufbereitete Sensorsignale, die in einem realen Fahrzeug von einer, den Frontbereich des Fahrzeugs überwachenden Kamera mit entsprechender Bildverarbeitung generiert werden. Dabei werden die geometrischen Werte (Abstände und Winkel) eines sogenannten „U-Frames“ berechnet, der die vor dem Fahrzeug erkannten Fremdfahrzeuge umschließt (siehe Bild 7b).

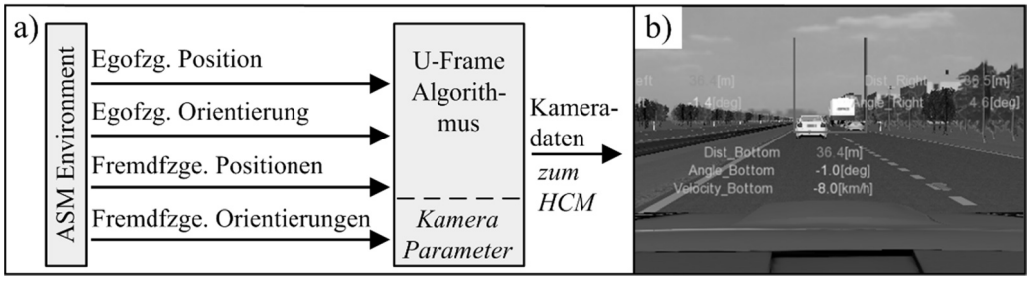


Bild 7: a) Struktur der Kamerasimulation, b) animierter U-Frame

Für die Simulation der Kamerasensorik wurde ein U-Frame Algorithmus entwickelt, der anhand der Position und der Orientierung, sowohl des Ego-Fahrzeugs als auch der in der Nähe befindlichen Fremdfahrzeuge, den blendfreien U-Frame-Bereich berechnet. Die dazu notwendigen Informationen sind in den Echtzeitmodellen hinterlegt und so während der Simulation zur Laufzeit verfügbar. Der U-Frame Algorithmus ist durch Modifizieren der Kameraparameter, wie beispielsweise der Position der Kamera im Fahrzeug, des horizontalen und des vertikalen Öffnungswinkel, etc. anpassbar.

Neben den Kameradaten für das GFHB benötigt das HCM E-Horizontdaten für die prädiktive Lichtadaptierung PAFS. Die Position des Ego-Fahrzeugs und der Straßenverlauf in Form eines Most Likely Path (MPP) werden im Umgebungsmodell berechnet und als GPS-Informationen zur Verfügung gestellt (siehe Bild 8).

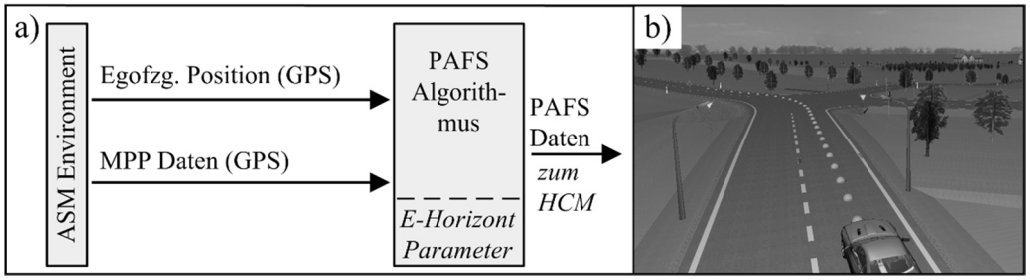


Bild 8: a) Struktur der E-Horizont Simulation, b) animierter MLP

Die Informationen zu der Fahrzeugposition und dem MLP werden neben weiteren Fahrzeuggrößen wie Geschwindigkeit, Gierrate, Lenkradwinkel, etc. einem PAFS Algorithmus zur Verfügung gestellt, der darauf basierend die Sollgrößen für die Scheinwerferaktoren berechnet und an das Steuergerät weiterleitet. Die Hauptfunktionalität des Steuergeräts besteht somit darin, die Informationen von der Kamera, von dem elektronischen Horizont und von dem Fahrzustand des Ego-Fahrzeugs zu kombinieren, um eine optimale Ausleuchtung der Straße bei einer gleichzeitigen Vermeidung der Blendung weiterer Verkehrsteilnehmer zu erreichen.

4.4 Simulationsergebnisse

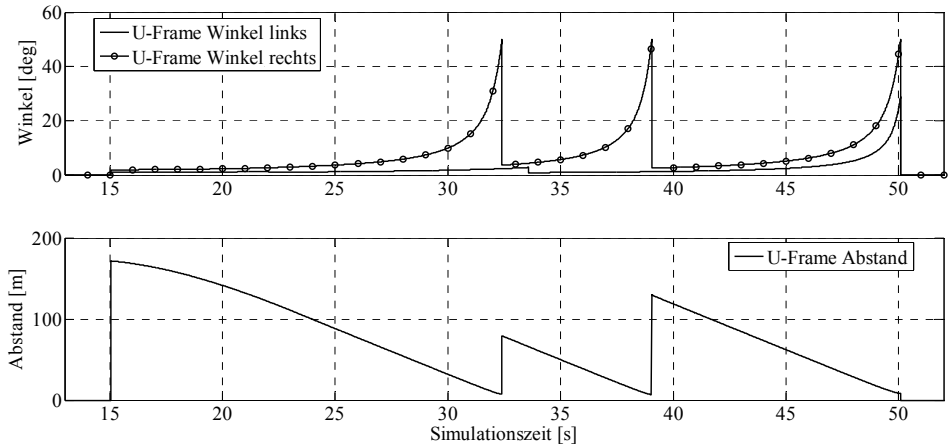


Bild 9: Ergebnisse der Kamerasimulation bei einem Überholmanöver

Zur Verdeutlichung der U-Frame Charakteristik sind in Bild 9 Simulationsergebnisse der Winkel und Abstände des U-Frames während eines Überholmanövers dargestellt. In diesem Simulationsszenario überholt das Ego-Fahrzeug drei auf der rechten Fahrspur fahrende Fremdfahrzeuge auf einem geraden Autobahnabschnitt. Alle Fahrzeuge fahren mit einer konstanten Geschwindigkeit und Spurwechsel finden nicht statt. Der obere Graph in Bild 9 zeigt die Winkel (positiv: rechtsseitig des Ego-Fahrzeugs; negativ: linksseitig des Ego-Fahrzeugs) von der Kameraposition im Ego-Fahrzeug zur linken und zur rechten Begrenzung des U-Frames in einem ego-fahrzeugfesten Koordinatensystem während des Überholmanövers. Dadurch dass die Fremdfahrzeuge an der rech-

ten Seite des Ego-Fahrzeugs überholt werden, bleiben die U-Frame Winkel während dieses Szenarios stets positiv. Zu den Zeitpunkten t1 = 32s, t2 = 39s und t3 = 50s ist das Ego-Fahrzeug so nah an einem Fremdfahrzeug, dass dieses aus dem Sensorbereich der Kamera verschwindet und der U-Frame dadurch sprunghaft Größen ändert. In dem dritten Abschnitt (delta_t = 39s-50s) ist nur ein Fremdfahrzeug innerhalb des Erfassungsbereiches der Kamera, so dass hier der Winkelbereich zwischen linker und rechter U-Frame Begrenzung nicht mehr so groß ist wie vorher, wo stets mindestens zwei verschiedene Fahrzeuge den U-Frame aufgespannt haben.

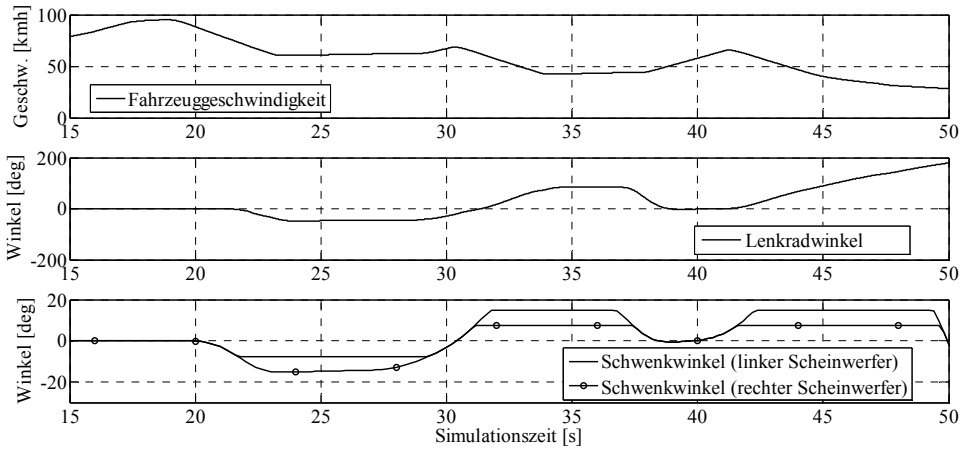


Bild 10: Ergebnisse der E-Horizontsimulation

Bild 10 zeigt beispielhafte Simulationsergebnisse einer Fahrt entlang einer kurvigen Straße, um den im Modell verwendeten PAFS Algorithmus zu verdeutlichen. Der obere Graph zeigt die Ego-Fahrzeuggeschwindigkeit, der Lenkradwinkel des Fahrers und die Schwenkwinkel als Sollgrößen für die Scheinwerferaktoren sind im mittleren und im unteren Graphen dargestellt. Beim Vergleich der Schwenkwinkel mit dem Lenkradwinkel wird deutlich, dass die Schwenkwinkel für die Scheinwerfer gesetzt werden, bevor der Fahrer das Lenkrad betätigt, die Scheinwerfer also prädiktiv dem vorausliegenden Straßenverlauf angepasst werden.

Die HIL Simulation des ADAS Steuergeräts stellt zwar einen wesentlichen Teil des Closed-Loop Testsystems dar, allerdings ist die Kopplung mit dem Fahrsimulator, und, insbesondere bei dem verwendeten lichtbasiertem Assistenzsystem, die Visualisierung ein weiterer wichtiger Aspekt zur Beurteilung der Steuergerätfunktionalität.

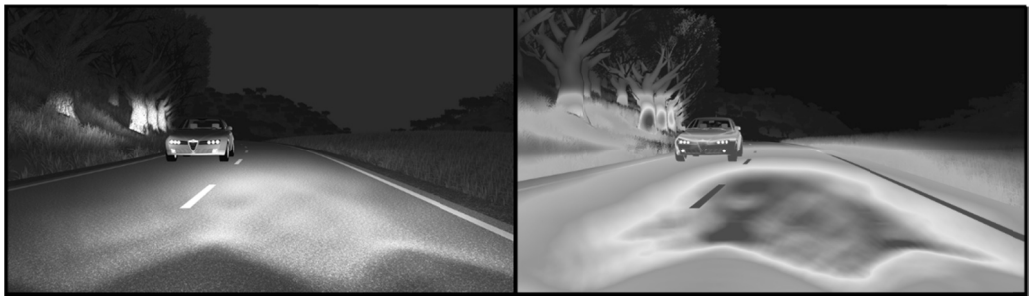


Bild 11: Beispiel für ein Nachtfahr-Szenario

Die Leuchtcharakteristika der Scheinwerfer sind als Modelle in dem Visualisierungssystem hinterlegt, so dass mit den vom Steuergerät erzeugten Stellsignalen die Ausleuchtung der Straße in Echtzeit berechnet und angezeigt werden kann, Bild 11 zeigt eine beispielhafte Nachtfahrszene, dargestellt in Original- und Falschfarben. Durch die realistische Darstellung der Ausleuchtung lassen sich die Steuergerätefunktionalitäten simulativ bewerten. Für weiterführende Informationen zur Scheinwerfervisualisierung bei Nachtfahrten sei an dieser Stelle auf [Ber05] und [BB12] verwiesen.

5 Resümee und Ausblick

Das im Rahmen des *TRAFFIS*-Projekts aufgebaute Closed-Loop Testsystem bietet vielseitige Möglichkeiten, um zukünftige ADA-Systeme während der Entwicklung und dem Test zu unterstützen. Da im Gegensatz zu herkömmlichen Fahrdynamikregelsystemen bei ADAS nicht mehr das reine Fahrverhalten, sondern zunehmend komplexe Szenarien relevant sind, in denen weitere Verkehrsteilnehmer in unterschiedlichen Situationen involviert sind, werden für die simulative Bewertung der ADAS-Funktionen zusätzliche Anforderungen an die Umgebungs- und Verkehrsmodelle gestellt. Dazu wurden im Rahmen des *TRAFFIS*-Projektes die eingesetzten Echtzeitmodelle derart funktional erweitert, um ADAS relevante Szenarien simulieren zu können.

Die Möglichkeiten, die solch ein Closed-Loop Testsystem für die Entwicklung und den Test von ADAS bietet, wurden anhand der aufgebauten Simulationsumgebung mit einem lichtbasierten Fahrerassistenzsystem aufgezeigt. Durch die Kombination des closed-loop Betriebs des Steuergeräts mit dem Fahrsimulator und dem Visualisierungssystem, lassen sich nicht nur die Algorithmen zur Ausleuchtung der Straße und die mögliche Blendung anderer Verkehrsteilnehmer beurteilen, sondern der reale Fahrer im Simulator erhält durch die visuelle, akustische, haptische und kinästhetische Stimulation zusätzlich einen realistischen Fahreindruck und kann auf diese Weise ein ADA-System im Labor „erfahren“.

EUROPÄISCHE UNION
Investition in unsere Zukunft
Europäischer Fonds
für regionale Entwicklung

TRAFFIS ist ein Forschungsprojekt aus dem EFRE kofinanzierten operationellen Programm für NRW im Ziel „Regionale Wettbewerbsfähigkeit und Beschäftigung“ 2007-2013.

Literatur

[Abd12] ABDELSABOUR, A.: Conception and Implementation of a Traffic-proof Driver Model under Consideration of Intersection Situations. Master Thesis, Fakultät für Maschinenbau, Produktentstehung (PE), Universität Paderborn, 2012.

[AQT12] AL QUAISI, I.; TRÄCHTLER, A.: Constrained Linear Quadratic Optimal Controller for Motion Control of ATMOS Driving Simulator. In: Proceedings of Driving Simulation Conference – DSC 2012, Paris, France, Sept. 2012.

[ABS14] AMELUNXEN, H.; BERSSENBRÜGGE, J.; SCHMIDT, C.: Driver-in-the-Loop-Simulation – DiL zur Entwicklung von Fahrerassistenzsystemen: weit mehr als „nur“ HiL. In: Automobil Elektronik 03/2014, S.56-59.

[AS14] AMELUNXEN, H.; SCHMIDT, C.: Driver-in-the-Loop-Simulation for Advanced Driver Assistance Systems. In: Proceedings of AVEC14, 12th International Symposium on Advanced Vehicle Control, Tokyo, Japan, 2014.

[Ber05] BERSSENBRÜGGE, J.: Virtual Night Drive – A Method for Displaying the Complex Light Distribution Characteristics of Modern Headlight Systems Within a Simulated Night Drive. Dissertation, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 177, 2005.

[BB12] BERSSENBRÜGGE, J.; BONNER, E.: GPU-Based Local Tone Mapping in the Context of Virtual Night Driving. In: Journal of Computing and Information Science in Engineering, Vol. 13(2), 2 Park Avenue, New York, NY 10016, Jun. 2013.

[HG13] HASSAN, B.; GAUSEMEIER, J.: Concept for a Task-Specific Reconfigurable Driving Simulator. In: Proceedings of SIMUL 2013, the Fifth International Conference on Advances in System Simulation, Venice, Italy, 27. Oct. - 1. Nov. 2013.

[Kre12] KREFT, S.: Systematik zur effizienten Bildung geospezifischer Umgebungsmodelle für Fahrsimulationen. Dissertation, Universität Paderborn, HNI-Verlagsschriftenreihe, Paderborn, Band 305, Jan. 2012.

[Kre13] KREFT, S.: Systematische Integration von Geodaten zur Bildung geospezifischer Umgebungsmodelle für Fahrsimulationen. In: 11. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung, HNI-Verlagsschriftenreihe, Paderborn, Band 311, Apr. 2013.

[TRA14-ol] *TRAFFIS*: www.traffis.de

Autor

Dr.-Ing. Hendrik Amelunxen studierte Maschinenbau an der Universität Paderborn. Von 2007 bis 2011 war er von der dSPACE GmbH geförderter Stipendiat der International Graduate School Dynamic Intelligent Systems und am Lehrstuhl für Regelungstechnik und Mechatronik tätig. Seit 2011 ist er Modellierungsexperte im Bereich der Hardware-in-the-Loop Simulation bei der dSPACE GmbH.

Entwicklung eines prädiktiven Motion-Cueing-Verfahrens für den ATMOS-Fahrsimulator

Dipl.-Ing. Daniel Zimmermann, M. Sc. Andreas Kohlstedt, Dr.-Ing. Sandra Gausemeier, Prof. Dr.-Ing. Ansgar Trächtler
Universität Paderborn
Fürstenallee 11, 33102 Paderborn
Tel. +49 (0) 5251 / 60 63 21, Fax. +49 (0) 5251 / 60 62 97
E-Mail: Daniel.Zimmermann@uni-paderborn.de

Zusammenfassung

Dieser Beitrag beschreibt ein neuartiges Motion-Cueing-Verfahren für einen Fahrsimulator, der zentraler Bestandteil einer Entwicklungsumgebung für fortgeschrittene Fahrerassistenzsysteme (ADAS) ist. Motion Cueing bezeichnet den Berechnungsvorgang, in dem die Beschleunigungen, die auf ein reales Fahrzeug einwirken würden, in Soll-Trajektorien der Komponenten der Bewegungsplattform für die Simulation transformiert werden. Einleitend erfolgt eine technische Beschreibung des ATMOS-Fahrsimulators und der Architektur der rekonfigurierbaren Simulationsumgebung, in die er eingebettet ist. Anschließend werden aktuelle Motion-Cueing-Algorithmen diskutiert und Möglichkeiten zur Verbesserung identifiziert. Darauf aufbauend wird das im Rahmen der vorliegenden Arbeit entwickelte modellprädiktive Motion-Cueing sowie seine Implementierung in die bestehende Simulationsumgebung ausführlich beschrieben. Zur Bewertung des neuartigen Verfahrens wurden Simulationen durchgeführt, die die Leistungsfähigkeit des Verfahrens mit der von etablierten Motion-Cueing-Verfahren vergleichen. Zusätzlich wurden Messdaten aus realen Testdaten zur Validierung herangezogen. Die Resultate werden im Beitrag vorgestellt und diskutiert, bevor abschließend ein Resümee gezogen und ein Ausblick auf weiterführende Arbeiten gegeben werden.

Schlüsselworte

Fahrsimulator, Motion-Cueing, Modellbasierte prädiktive Regelung, Prüfstandsautomatisierung

Development of a predictive motion cueing algorithm for the ATMOS driving simulator

Abstract

This paper describes the development of a novel motion cueing algorithm for a driving simulator which is the central part of an environment for development of advanced driver assistance Systems (ADAS). Motion cueing terms the calculation procedure to transform accelerations acting on drivers in real vehicles into reference trajectories for the components of the motion systems that is used in simulation. Introductory a technical description of the ATMOS driving simulator and the architecture of its reconfigurable simulation environment is given. Afterwards, the state of the art of motion cueing algorithms is presented and discussed in order to identify potential for improvements. As a consequence, the model predictive motion cueing algorithm developed within the presented work and its implementation in the given simulation environment is described. Simulation results are presented in order to evaluate and compare the novel method's performance to that of established methods. Additionally, measurement data from a real test vehicle is shown to complete the validation. The results are illustrated and analyzed in the paper. Finally, a conclusion and a prospect on further work is given.

Keywords

Driving simulator, motion cueing, model predictive Control, test bed automation

1 Der ATMOS-Fahrsimulator

1.1 Aufbau

Ziel der vorliegenden Arbeit ist die Entwicklung eines neuartigen Motion-Cueing-Verfahren für den ATMOS-Fahrsimulator am Heinz Nixdorf Institut. Wie in Bild 1 links zu sehen ist, besteht dieser aus einer Fahrzeugkabine (Mockup), einer Projektionskanzel und einem Bewegungssystem. Im Inneren der Kanzel wird die Szenerie auf einer 270° umspannenden Rundprojektion nahtlos dargestellt, die das Blickfeld des Fahrers vollständig abdeckt. Zusätzlich wird das realitätsnahe visuelle Empfinden durch die Darstellung der korrespondierenden Bilder im Innen- und den Außenspiegeln gefördert, die durch Video-Tracking der Kopfposition laufend der Blickrichtung des Fahrers angepasst werden. Beim Mockup handelt es sich um ein mit Feedback-Aktorik ausgestattetes Fahrzeugchassis. Der Fahrer verfügt hier über reale Steuerungselemente. Neben der visuellen Information erhält er haptisches Feedback durch Lenkrad und Pedalerie und kann die Geschwindigkeit von den Instrumenten ablesen, um die Fahrsituation einzuschätzen.

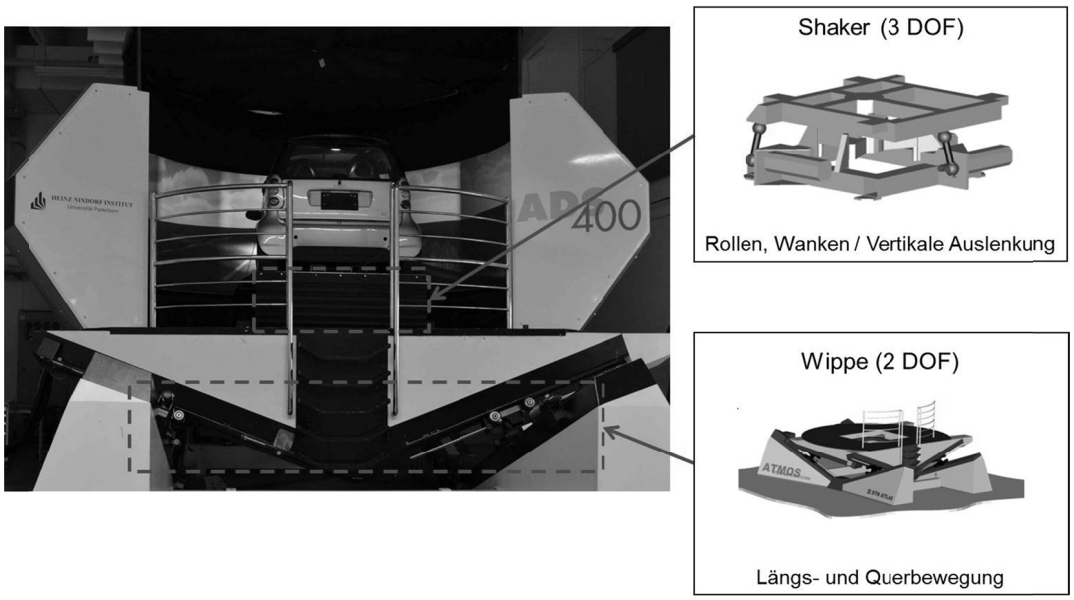


Bild 1: Bewegungssystem des ATMOS-Fahrsimulators

Das Mockup ist auf einer Shakerplattform montiert. Sie ist eine von zwei unabhängigen Komponenten des Bewegungssystems, die im rechten Teil von Bild 1 skizziert sind. Der Shaker ist mit drei Kurbeltrieben ausgestattet und kann das Mockup um die Längs- und Querachse neigen sowie in vertikaler Richtung auslenken. Während die durch den Shaker bereitgestellten Freiheitsgrade das Mockup relativ zur Projektion bewegen, kann die Kanzel zusätzlich noch im Ganzen geneigt werden. Dazu dient die zweite Komponente des Bewegungssystems, bei der vier Linearantriebe die beiden Teile einer Wippenkonstruktion gleichzeitig und in beliebiger Kombination längs und quer zueinander verschieben können. Wie die insgesamt fünf Freiheitsgrade des Bewegungssystems

während der Simulation eingesetzt werden, wird im weiteren Verlauf des Beitrags detailliert beschrieben.

Die Struktur der für den Simulatorbetrieb erforderliche Infomationsverarbeitung zeigt Bild 2 Im Kern wird eine dSPACE-Echtzeithardware eingesetzt, auf der die sich durch die Fahrereingaben und Umgebungsbedingungen einstellenden Fahrzeugbewegungen modellbasiert berechnet werden. Diese wiederum werden in weiteren Berechnungsschritten durch das Motion Cueing, eine inverse Kinematik und die Motorregelungen in Sollwerte für die sieben Frequenzumrichter des Bewegungssystems transformiert. Die Kommunikation zwischen der Fahrzeugkabine und der Echtzeithardware wurde im vorliegenden Fall durch eine CAN-Schnittstelle realisiert. Neben den Fahrereingaben können so weitere Sensorsignale aus dem Simulator empfangen und umgekehrt Steuersignale für die Feedback-Aktorik und die Instrumentenanzeige im Mockup gesendet werden.

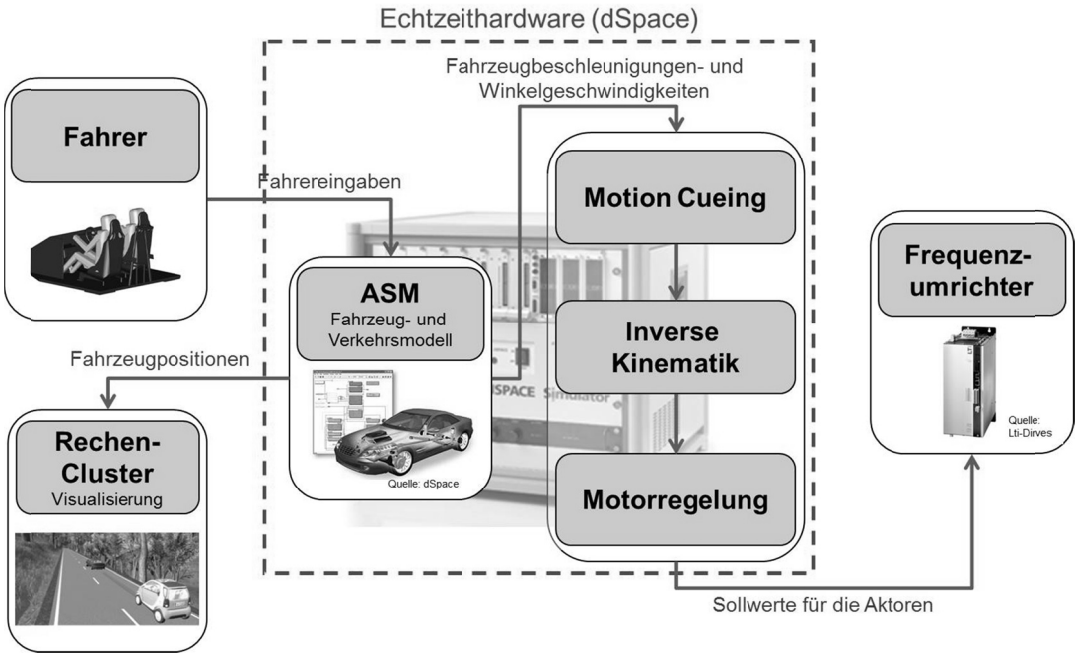


Bild 2: Signalstruktur der Simulationsumgebung

Die hier beschriebenen Strukturen wurden im Rahmen eines öffentlich geförderten Projektes aufgebaut, bei dem neben zwei Fachgruppen des HNI noch vier weitere Projektpartner aus der Industrie eine Test- und Trainingsumgebung für ADAS entwickelten. Ein wichtiges Kriterium dabei war die Rekonfigurierbarkeit. Dies beinhaltet auch die Möglichkeit, verschiedene Simulatoren einzusetzen oder die Fahrzeugkabinen auf dem ATMOS-Simulator auszutauschen. Durch die informationstechnische Anbindung mittels CAN-Schnittstelle kann beispielsweise das in Bild 1 gezeigte Kleinstwagen-Chassis gegen eine LKW-Kabine ausgetauscht werden. Die Realisierung der dazu erforderlichen Anpassungen in der Informationsverarbeitung (beispielsweise der Austausch des Fahrzeugmodells) wird in den folgenden Ausführungen beschrieben.

1.2 Modulare, rekonfigurierbare Simulationsumgebung

Ziel der Arbeiten war unter anderem, eine in hohem Maße rekonfigurierbare Simulationsumgebung zu schaffen. Zu diesem Zweck wurde eine weitgehend modulare Systemstruktur konzipiert [HBA+13] und implementiert. Sie besteht aus drei zusammenhängenden Softwarewerkzeugen, die den täglichen Umgang mit der modularen Simulationsumgebung erleichtern.

Mithilfe des „TRAFFIS Konfigurator“ erfolgt die domänenübergreifende Spezifikation des zu betreibenden Fahrsimulators. Der Nutzer wählt unter einer Menge von Lösungselementen für jedes Systemmodul diejenige aus, die für den anstehenden Simulatorbetrieb gewünscht sind. So kann beispielsweise das während des Simulationsablaufs zur Berechnung der Bewegung der Bewegungsplattform herangezogene Motion-Cueing-Verfahren ausgewählt werden. Auch der Austausch von physikalischen Komponenten, zum Beispiel des Visualisierungssystems oder der Fahrerkabine, sind in der Software vorgesehen. Bei der Auswahl der Teilmodelle wird deren Kompatibilität durch das Programm geprüft und so nicht mögliche Simulatorkonfigurationen ausgeschlossen. Sobald der Nutzer ein realisierbares Gesamtsystem zusammengestellt hat, erzeugt der Konfigurator eine XML-Datei, die alle relevanten Informationen über das System enthält. Die Zusammenstellung der Konfiguration kann dabei auf jedem beliebigen Arbeitsplatzrechner erfolgen.

Die erzeugte Konfigurationsdatei kann dann auf dem Leitstandrechner des Fahrsimulators mit dem zweiten Werkzeug, dem „TRAFFIS Assistant“, eingelesen werden. Er bietet dem Nutzer komfortable Möglichkeiten zum Betrieb des Simulators. Durch implementierte Softwareschnittstellen übernimmt er die automatisierte Überführung des Simulators in den betriebsbereiten Zustand auf Basis der gegebenen Konfiguration. Beispielsweise wird die Visualisierungsumgebung gestartet und die relevante Software auf das Echtzeitsystem geladen und bei Bedarf neu erzeugt. Ebenso bereitgestellt werden elementare, alltägliche Funktionen wie das Starten und Stoppen einer Fahrsimulation sowie das Ein- und Ausschalten der Bewegungsplattform. Aber auch die Überwachung des Gesamtsystems ist durch das Programm möglich. Das Programm stellt außerdem die Möglichkeit der Sprachkommunikation mit dem entfernt in der Fahrerkabine befindlichen Probanden bereit.

Auf dem Echtzeitrechner, der unter anderem die Berechnung der Plattformbewegung auf Grundlage eines Fahrzeugmodells und des Motion-Cueings vornimmt, läuft schließlich die dritte Komponente des Softwarepakets, der „TRAFFIS Moderator“. Er verwaltet die während der Simulation vorliegenden Signale und verteilt sie auf die Teilmodelle. Seine wesentliche Aufgabe besteht im Signalrouting.

2 Motion-Cueing-Verfahren

Es gibt eine ganze Reihe entwickelter Motion-Cueing-Verfahren. Ziel dieser Algorithmen ist es, die Bewegung des simulierten Fahrzeugs in eine Bewegung des Fahrsimulators umzusetzen, so dass das Beschleunigungsempfinden der Insassen möglichst realistisch ist. Da der Arbeitsraum des Simulators im Vergleich zum Bewegungsraum des Fahrzeugs deutlich eingeschränkt ist, ist die Entwicklung von Motion-Cueing-Verfahren keine triviale Aufgabe. Für die Auslegung derartiger Verfahren können unterschiedliche regelungstechnische Ansätze genutzt werden.

2.1 Der klassische Motion-Cueing-Algorithmus

Eines der ältesten und am meisten etablierten Verfahren ist der klassische Motion-Cueing-Algorithmus [GR97]. Bei diesem Algorithmus werden üblicherweise die eingehenden Fahrzeugbeschleunigungen in Längs-und Querrichtung zunächst herunterskaliert (siehe Bild 3). Anschließend werden diese Signale über entsprechende Filter in einen hochfrequenten und in einen niederfrequenten Teil aufgetrennt. Der hochfrequente Beschleunigungsanteil wird über eine Doppelintegration direkt in ein Wegsignal umgewandelt. Der niederfrequente Teil wird über die Tilt-Coordination in eine Neigung umgerechnet. Dabei macht man sich zunutze, dass ein Insasse im Fahrsimulator nicht zwischen einer lang anhaltenden Beschleunigung und einer Neigung unterscheiden kann (bis zu einer Neigung von etwa 20-30 Grad)[Mit83]. Die so berechnete Neigung wird zu dem Winkel, der aus der Fahrzeugwinkelgeschwindigkeit resultiert, hinzuaddiert. Nachgeschaltete Washout-Filter sorgen dafür, dass der Simulator permanent in die Nullposition zurückdriftet.

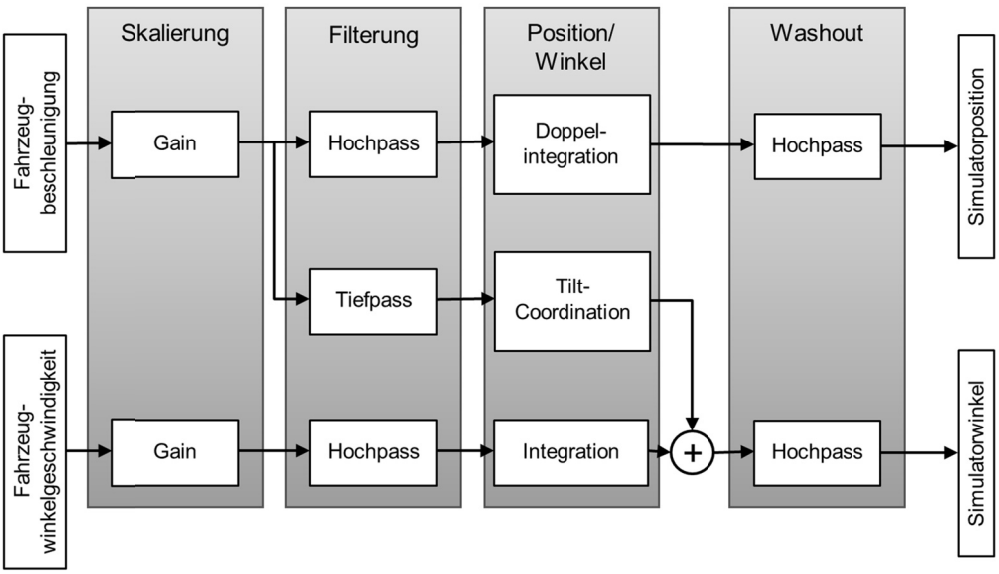


Bild 3: Klassisches Motion-Cueing-Verfahren

Grund für den weit verbreiteten Einsatz des klassischen Motion-Cueing-Verfahrens sind vor allem die einfache Implementierung und Auslegung der Parameter. Jedoch kann der klassische Algorithmus nicht unterschiedliche Fahrmanöver optimal abbilden. Vielmehr muss ein Kompromiss derart gefunden werden, damit der Simulator auch bei hohen Beschleunigungen den Arbeitsraum nicht verlässt. Das führt jedoch zwangsläufig dazu, dass Fahrmanöver mit geringeren Beschleunigungen nicht optimal abgebildet werden. Einen Ausweg bieten hier fortgeschrittene Motion-Cueing-Verfahren wie beispielsweise adaptive Motion-Cueing-Algorithmen ([PDM75], [RN85]) oder die modellbasierte prädiktive Regelung.

2.2 Modellbasierte prädiktive Regelung

Eine weitere Klasse von Motion-Cueing-Algorithmen ist die modellbasierte prädiktive Regelung (MPC) [DRK+09]. Dabei wird in jedem Zeitschritt eine Kostenfunktion über einen definierten Prädiktionszeitraum minimiert. Im einfachsten Fall beinhaltet diese Kostenfunktion die Abweichung zwischen den Beschleunigungen des Fahrzeugaufbaus und der Simulatorkabine jeweils an der Position des Fahrers. Diese Kostenfunktion kann darüber hinaus durch menschliche Wahrnehmungsmodelle erweitert werden [AT12]. Ein Vorteil von MPC-Verfahren ist die direkte Berücksichtigung von Simulatorbeschränkungen, so dass der Bewegungsraum besser ausgenutzt werden kann als z.B. beim klassischen Verfahren. Für die Implementierung des MPC-Verfahrens in diesem Beitrag wird die in Bild 4 angegebene Struktur verwendet. Demzufolge wirkt dieses nur auf die hochfrequenten Beschleunigungsanteile. Niederfrequente Beschleunigungen werden wie bei dem klassischen Verfahren mit Hilfe der Tilt-Coordination umgesetzt. Da das MPC-Verfahren für die Einhaltung der Simulatorgrenzen sorgt, wird hier auf das Washout-Filter verzichtet.

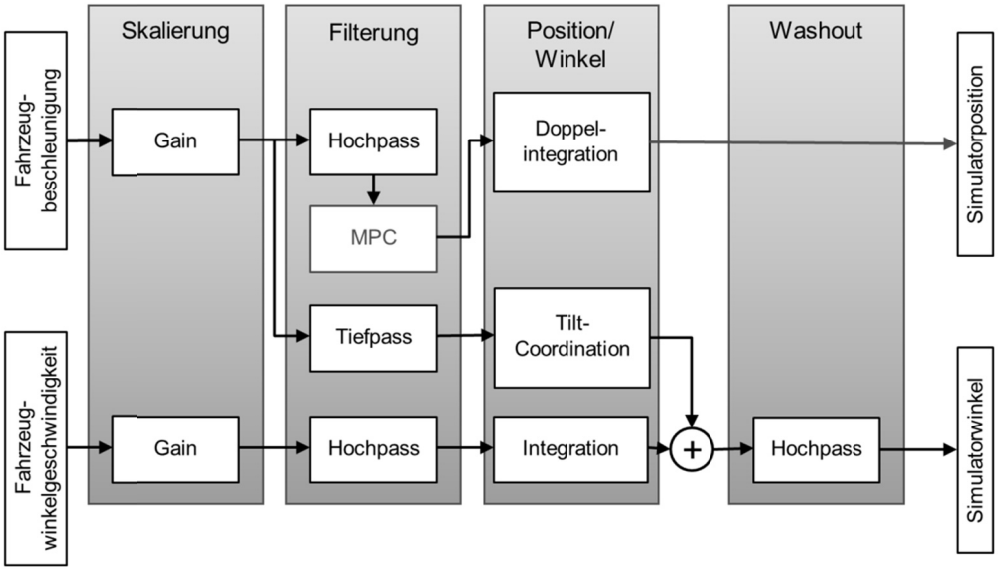


Bild 4: Modellbasierte prädiktive Regelung

2.3 Prädiktion

Während für die Nutzung von MPC-Verfahren als Motion-Cueing-Algorithmus oft der aktuelle Beschleunigungs-Referenzwert über den gesamten Prädiktionszeitraum konstant gehalten wird, kann durch die Berücksichtigung zukünftiger Referenzwerte die Regelung verbessert werden. Das Einfließen zukünftiger Referenzwerte in das Motion-Cueing-Verfahren wird im Folgenden als Previewing bezeichnet. Die Ermittlung dieser Referenzwerte bzw. die Vorhersage von Beschleunigungs-Sollwerten ist keine triviale Aufgabe. Dazu muss unter anderem das Verhalten des Fahrers im Simulator prädiziert werden. Da es unmöglich ist, das Verhalten des Fahrers ideal vorherzusagen, ist es demzufolge auch nicht möglich, exakte Beschleunigungs-Referenzwerte zu generieren. Trotzdem liefern auch nichtideale Referenzwerte eine Verbesserung des Motion-Cueing-Verfahrens gegenüber MPC-Verfahren ohne Previewing, d.h. mit konstant angenommenen Referenzwerten [BBM13]. Die Implementierung einer Previewing-Strategie ist Teil zukünftiger Arbeiten. Neben einem Fahrermodell werden dazu auch ein Fahrzeugmodell und Streckeninformationen in die Berechnung einfließen. Für den Vergleich der Motion-Cueing-Verfahren in diesem Beitrag werden jedoch zuvor generierte Referenzwerte genutzt, d.h. das Previewing wird als ideal angenommen.

3 Implementierung

Wie beschrieben sollen die dargestellten Motion-Cueing-Verfahren exemplarisch miteinander verglichen werden. Dazu zählen zum einen das klassische Verfahren, zum zweiten das MPC-Verfahren ohne Previewing (d.h. Referenzwert wird konstant gehalten) und zum dritten das MPC-Verfahren mit Previewing. Für den Test wurde ein kurzes Beschleunigungsmanöver gewählt (siehe Bild 5).

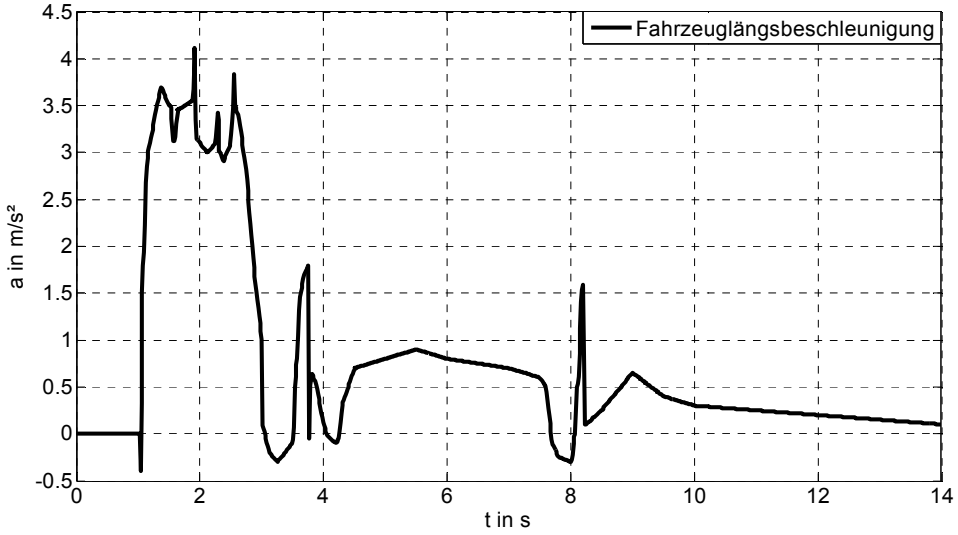


Bild 5: Beschleunigungsvorgang des gewählten Manövers

Die Motion-Cueing-Algorithmen wurden entsprechend der Bewegungsgrenzen des ATMOS-Fahrsimulators (siehe Tabelle 1).

Tabelle 1: Bewegungsgrenzen des ATMOS-Fahrsimulators

Wippen:

	Position / Winkel	Geschwindigkeit	Beschleunigung
Longitudinal	±733 (mm)/ ±13.5°	1.3 m/s	±3 (m/s²)
Lateral	±522 (mm)/ ±10°	1 m/s	±2.2 (m/s²)

Shaker:

	Position / Winkel	Geschwindigkeit	Beschleunigung
Vertikal	±72.5 (mm)	±0.6 (m/s)	±4 (m/s²)
Nicken	±7 (°)	±50 (°/s)	±360 (°/s²)
Wanken	±7 (°)	±50 (°/s)	±360 (°/s²)

Für das klassische Verfahren ergeben sich die in Bild 6 dargestellten Parameter. Die Auswahl der festgelegten Parameter garantiert, dass der Simulator für den dargestellten Beschleunigungsvorgang die Bewegungsgrenzen möglichst gut ausnutzt. Für das MPC-Verfahren wurden identische Parameter genutzt, jedoch fällt hier der Hochpass des Washouts heraus.

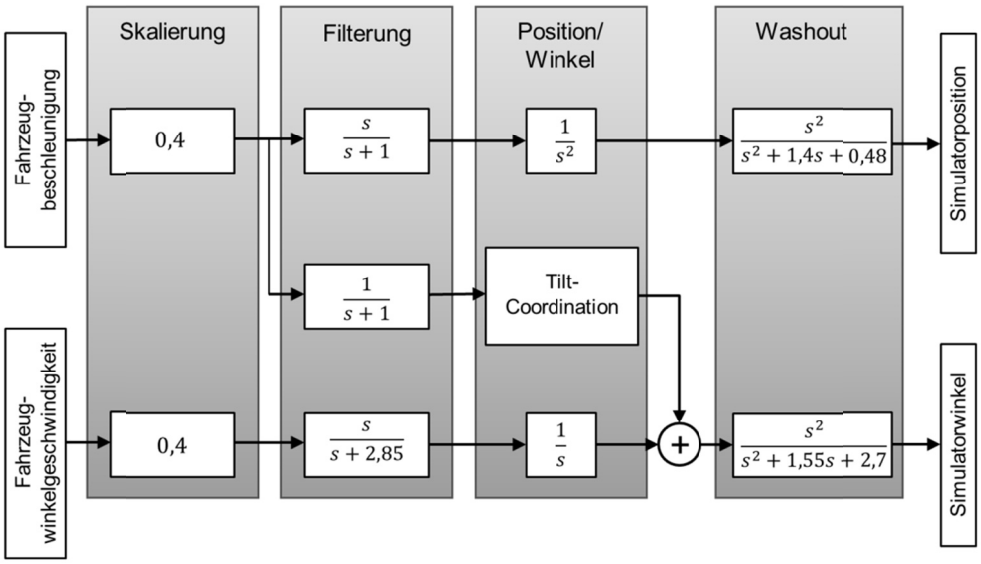


Bild 6: Parameterauswahl

Um die drei Verfahren miteinander vergleichen zu können muss eine Bewertungs- oder Gütefunktion G gefunden werden. Im ausgewählten Test soll dafür das Integral der quadratischen Abweichung zwischen Simulator- und Fahrzeugbeschleunigung genutzt werden:

In Tabelle 2 sind die Simulationsergebnisse dargestellt. Für die beiden MPC-Verfahren wurden dabei der Prädiktionshorizont und der Stellhorizont variiert. Der Zeitschritt für die Optimierung wurde hingegen mit *10 ms* konstant gehalten. Unterstrichene Werte

$$G = \int_{t=0}^{t=t_e} \left(a_{Fahrzeug} - a_{Simulator}\right)^2 dt$$

bedeuten, dass hier das jeweilige MPC-Verfahren besser abschneidet als das klassische Verfahren. Wie zu erkennen, ist die Wahl der Parameter für die beiden MPC-Verfahren entscheidend. Bei einem Prädiktionshorizont von 50 schneiden beide MPC-Verfahren schlechter ab als das klassische. Bei einem Prädiktionshorizont von 100 sind die MPC-Verfahren besser. Für einen noch größeren Prädiktionshorizont schneidet das MPC-Verfahren ohne Previewing wiederum schlechter ab. Das liegt daran, dass hier die die Referenzwerte zu lange konstant gehalten werden. Bei einem MPC-Verfahren bedeutet also nicht, dass ein höherer Prädiktionshorizont zwangsläufig auch zu einem besseren Motion-Cueing-Verfahren führt.

Tabelle 2: Ermittelte Gütemaße

Motion-Cueing-Verfahren			Parameter für MPC		
Klassisches Verfahren	**MPC ohne Previewing**	**MPC mit Previewing**	Prädiktions-horizont	Stell-horizont	Zeitschritt
0,7417	**1,482**	**1,472**	*50*	*20*	*10 ms*
	0,6873	**0,6002**	*100*	*50*	
	0,6836	**0,5976**	*100*	*100*	
	1,021	**0,2302**	*200*	*100*	
	0,9869	**0,2295**	200	200	

Bei dem MPC-Verfahren mit Previewing führt ein höherer Prädiktionshorizont abermals zu einer deutlichen Verbesserung. Für eine Implementierung müssen hier zwei Sekunden prädiziert werden. Außerdem ist zu bedenken, dass das Previewing hier als ideal angenommen wird. Dennoch lässt sich schlussfolgern, dass MPC-Verfahren insbesondere mit Previewing Vorteile gegenüber dem klassischen Verfahren liefern können. In Bild 7 sind die Beschleunigungen dargestellt. Das MPC-Verfahren mit Previewing bildet die Fahrzeuglängsbeschleunigung am besten ab. Hier ist auch gut zu erkennen, dass das MPC-Verfahren schon vor dem eigentlichen Beschleunigungsvorgang reagiert.

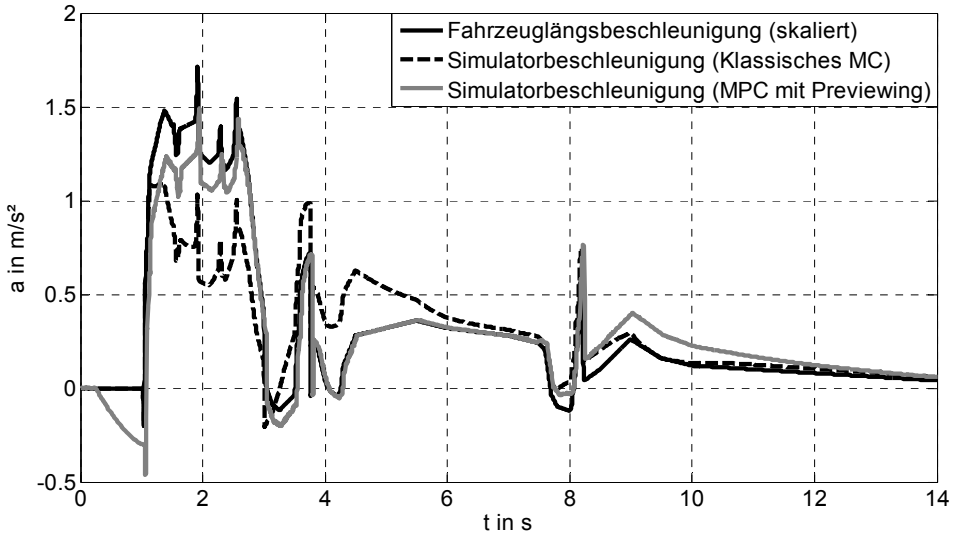


Bild 7: Vergleich der Beschleunigungen für einen Prädiktionshorizont $n_p = 200$, einen Stellhorizont $n_c = 200$ und einer Schrittweite $t_s = 10\ ms$

4 Resümee und Ausblick

Eine Komponente zukünftiger Arbeiten ist wie beschrieben die Entwicklung einer Strategie für die Prädiktion des Fahrerverhaltens bzw. von Beschleunigungs-Referenzwerten. Dazu muss auch ein geeignetes Fahrzeugmodell entwickelt werden sowie eine Schnittstelle zwischen dem Motion-Cueing-Verfahren und dem Streckenmodell. Außerdem werden alternative MPC-Strukturen umgesetzt und getestet.

Literatur

[AT12] AL QAISI, I.; TRÄCHTLER, A.: Constrained Linear Quadratic Optimal Controller for Motion Control of ATMOS Driving Simulator. In Driving Simulation Conference 2012 Europe – DSC 2012, 2012.

[BBM13] BEGHI, A.; BRUSCHETTA, M.; MARAN, F.: A Real-Time Implementation of an MPC-Based Motion Cueing Strategy with Time-Varying Prediction: 2013 IEEE International Conference on Systems, Man and Cybernetics (SMC 2013), 2013; S. 4149-4154.

[DRK+09] DAGDELEN, M. ET AL.: Model-based predictive motion cueing strategy for vehicle driving simulators. In Control Engineering Practice, 2009, 17; S. 995-1003.

[GR97] GRANT, P. R.; REID, L. D.: Motion washout filter tuning: Rules and requirements. In Journal of Aircraft, 1997, 34; S. 145-151.

[HBA+13] HASSAN, B. ET AL.: Reconfigurable Driving Simulator for Testing and Training of Advanced Driver Assistance Systems: Proceedings of ISAM (IEEE International Symposium on Assembly and Task Planning), Xi'an, China, 2013.

[Mit83] MITTELSTÄDT, H.: A new solution to the problem of the subjective vertical. In Naturwissenschaften, 1983, 70; S. 272-281.

[PDM75] PARRISH, R. V.; DIEUDONNE, J. E.; MARTIN JR; DENNIS J.: Coordinated adaptive washout for motion simulators. In Journal of Aircraft, 1975, 12; S. 44-50.

[RN85] REID, L. D.; NAHON, M. A.: Flight Simulation Motion-base Drive Algorithimns: Part 1-Developing and Testing the Equations. Institute for Aerospace Studies, Toronto University, 1985.

Autoren

Dipl.-Ing. Daniel Zimmermann studierte Mechatronik an der TU Dresden und ist seit Januar 2013 wissenschaftlicher Mitarbeiter am Lehrstuhl für Regelungstechnik und Mechatronik. Neben der Inbetriebnahme und Regelung des ATMOS-Fahrsimulators beschäftigt er sich mit der Weiterentwicklung von prädiktiven Motion-Cueing-Algorithmen.

M. Sc. Andreas Kohlstedt studierte Wirtschaftsingenieurwesen an der Universität Paderborn und ist seit Februar 2013 wissenschaftlicher Mitarbeiter am Lehrstuhl für Regelungstechnik und Mechatronik. Er hat vor allem an der Realisierung der rekonfigurierbaren Simulationsumgebung mitgewirkt.

Dr.-Ing. Sandra Gausemeier studierte Wirtschaftsingenieurwesen an der Universität Paderborn. Seit 2007 ist sie wissenschaftliche Mitarbeiterin des Lehrstuhls für Regelungstechnik und Mechatronik, promovierte dort im Jahre 2013 und ist im Forschungsbereich Fahrerassistenzsysteme tätig.

Prof. Dr.-Ing. Ansgar Trächtler hat an der Universität Karlsruhe (TH) Elektrotechnik studiert und 1991 am Institut für Regelungs- und Steuerungssysteme bei Prof. Föllinger promoviert. Von 1992 bis 1998 war er als wissenschaftlicher Assistent am Institut für Meß- und Regelungstechnik (Prof. Mesch) der Universität Karlsruhe tätig, habilitierte sich 2000 und erhielt die Venia Legendi für das Fach „Meß- und Regelungstechnik“. Für seine wissenschaftlichen Arbeiten erhielt er 1998 den Eugen-Hartmann-Preis der GMA und den Messtechnik-Preis des AHMT (Arbeitskreis der Hochschullehrer für Messtechnik e.V). Von 1998 bis 2004 war er bei der Robert Bosch GmbH tätig und leitete zuletzt den Bereich Vorausentwicklung Fahrwerksysteme. Seit November 2004 ist er Professor und Leiter des Lehrstuhls für Regelungstechnik und Mechatronik der Universität Paderborn. Außerhalb der Universität engagiert sich Prof. Trächtler in der VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik (GMA) und in der International Federation of Automatic Control (IFAC).

Das Heinz Nixdorf Institut – Interdisziplinäres Forschungszentrum für Informatik und Technik

Das Heinz Nixdorf Institut ist ein Forschungszentrum der Universität Paderborn. Es entstand 1987 aus der Initiative und mit Förderung von Heinz Nixdorf. Damit wollte er Ingenieurwissenschaften und Informatik zusammenführen, um wesentliche Impulse für neue Produkte und Dienstleistungen zu erzeugen. Dies schließt auch die Wechselwirkungen mit dem gesellschaftlichen Umfeld ein.

Die Forschungsarbeit orientiert sich an dem Programm „Dynamik, Mobilität, Vernetzung: Eine neue Schule des Entwurfs der technischen Systeme von morgen“. In der Lehre engagiert sich das Heinz Nixdorf Institut in Studiengängen der Informatik, der Ingenieurwissenschaften und der Wirtschaftswissenschaften.

Heute wirken am Heinz Nixdorf Institut acht Professoren mit insgesamt 200 Mitarbeiterinnen und Mitarbeitern. Etwa ein Viertel der Forschungsprojekte der Universität Paderborn entfallen auf das Heinz Nixdorf Institut und pro Jahr promovieren hier etwa 30 Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler.

Heinz Nixdorf Institute – Interdisciplinary Research Centre for Computer Science and Technology

The Heinz Nixdorf Institute is a research centre within the University of Paderborn. It was founded in 1987 initiated and supported by Heinz Nixdorf. By doing so he wanted to create a symbiosis of computer science and engineering in order to provide critical impetus for new products and services. This includes interactions with the social environment.

Our research is aligned with the program “Dynamics, Mobility, Integration: En-route to the technical systems of tomorrow.” In training and education the Heinz Nixdorf Institute is involved in many programs of study at the University of Paderborn. The superior goal in education and training is to communicate competencies that are critical in tomorrows economy.

Today eight Professors and 200 researchers work at the Heinz Nixdorf Institute. The Heinz Nixdorf Institute accounts for approximately a quarter of the research projects of the University of Paderborn and per year approximately 30 young researchers receive a doctorate.

Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

Bd. 314 VON DETTEN, M.: Reengineering of Component-Based Software Systems in the Presence of Design Deficiencies. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 314, Paderborn, 2013 – ISBN 978-3-942647-33-5

Bd. 315 MONTEALEGRE AGRAMONT, N. A.: Immunorepairing of Hardware Systems. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 315, Paderborn, 2013 – ISBN 978-3-942647-34-2

Bd. 316 DANGELMAIER, W.; KLAAS, A.; LAROQUE, C.: Simulation in Produktion und Logistik 2013. HNI-Verlagsschriftenreihe, Band 316, Paderborn, 2013 – ISBN 978-3-942647-35-9

Bd. 317 PRIESTERJAHN, C.: Analyzing Self-healing Operations in Mechatronic Systems. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 317, Paderborn, 2013 – ISBN 978-3-942647-36-6

Bd. 318 GAUSEMEIER, J. (Hrsg.): Vorausschau und Technologieplanung. 9. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 5. und 6. Dezember 2013, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, HNI-Verlagsschriftenreihe, Band 318, Paderborn, 2013 – ISBN 978-3-942647-37-3

Bd. 319 GAUSEMEIER, S.: Ein Fahrerassistenzsystem zur prädiktiven Planung energie- und zeitoptimaler Geschwindigkeitsprofile mittels Mehrzieloptimierung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 319, Paderborn, 2013 – ISBN 978-3-942647-38-0

Bd. 320 GEISLER, J.: Selbstoptimierende Spurführung für ein neuartiges Schienenfahrzeug. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 320, Paderborn, 2013 – ISBN 978-3-942647-39-7

Bd. 321 MÜNCH, E.: Selbstoptimierung verteilter mechatronischer Systeme auf Basis paretooptimaler Systemkonfigurationen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 321, Paderborn, 2014 – ISBN 978-3-942647-40-3

Bd. 322 RENKEN, H.: Acceleration of Material Flow Simulations - Using Model Coarsening by Token Sampling and Online Error Estimation and Accumulation Controlling. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlags-schriftenreihe, Band 322, Paderborn, 2014 – ISBN 978-3-942647-41-0

Bd. 323 KAGANOVA, E.: Robust solution to the CLSP and the DLSP with uncertain demand and online information base. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlags-schriftenreihe, Band 323, Paderborn, 2014 – ISBN 978-3-942647-42-7

Bd. 324 LEHNER, M.: Verfahren zur Entwicklung geschäftsmodell-orientierter Diversifikationsstrategien. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 324, Paderborn, 2014 – ISBN 978-3-942647-43-4

Bd. 325 BRANDIS, R.: Systematik für die integrative Konzipierung der Montage auf Basis der Prinziplösung mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 325, Paderborn, 2014 – ISBN 978-3-942647-44-1

Bd. 326 KÖSTER, O.: Systematik zur Entwicklung von Geschäftsmodellen in der Produktentstehung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 326, Paderborn, 2014 – ISBN 978-3-942647-45-8

Bd. 327 KAISER, L.: Rahmenwerk zur Modellierung einer plausiblen Systemstrukturen mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 327, Paderborn, 2014 – ISBN 978-3-942647-46-5

Bd. 328 KRÜGER, M.: Parametrische Modellordnungsreduktion für hierarchische selbstoptimierende Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 328, Paderborn, 2014 – ISBN 978-3-942647-47-2

Bezugsadresse:
Heinz Nixdorf Institut
Universität Paderborn
Fürstenallee 11
33102 Paderborn

Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

Bd. 329 AMELUNXEN, H.: Fahrdynamikmodelle für Echtzeitsimulationen im komfortrelevanten Frequenzbereich. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 329, Paderborn, 2014 – ISBN 978-3-942647-48-9

Bd. 330 KEIL, R.; SELKE, H. (Hrsg):. 20 Jahre Lernen mit dem World Wide Web. Technik und Bildung im Dialog. HNI-Verlagsschriftenreihe, Band 330, Paderborn, 2014 – ISBN 978-3-942647-49-6

Bd. 331 HARTMANN, P.: Ein Beitrag zur Verhaltensantizipation und -regelung kognitiver mechatronischer Systeme bei langfristiger Planung und Ausführung. Dissertation, Fakultät für Wirtschaftswissenschaften, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 331, Paderborn, 2014 – ISBN 978-3-942647-50-2

Bd. 332 ECHTERHOFF, N.: Systematik zur Planung von Cross-Industry-Innovationen Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 332, Paderborn, 2014 – ISBN 978-3-942647-51-9

Bd. 333 HASSAN, B.: A Design Framework for Developing a Reconfigurable Driving Simulator. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 333, Paderborn, 2014 – ISBN 978-3-942647-52-6

Bd. 334 GAUSEMEIER, J. (Hrsg.): Vorausschau und Technologieplanung. 10. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 20. und 21. November 2014, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, HNI-Verlagsschriftenreihe, Band 334, Paderborn, 2014 – ISBN 978-3-942647-53-3

Bd. 335 RIEKE, J.: Model Consistency Management for Systems Engineering. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 335, Paderborn, 2014 – ISBN 978-3-942647-54-0

Bd. 336 HAGENKÖTTER, S .: Adaptive prozessintegrierte Qualitätsüberwachung von Ultraschalldrahtbondprozessen. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 336, Paderborn, 2014 – ISBN 978-3-942647-55-7

Bd. 337 PEITZ, C.: Systematik zur Entwicklung einer produktlebenszyklusorientierten Geschäftsmodell-Roadmap. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 337, Paderborn, 2015 – ISBN 978-3-942647-56-4

Bd. 338 WANG, R.: Integrated Planar Antenna Designs and Technologies for Millimeter-Wave Applications. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 338, Paderborn, 2015 – ISBN 978-3-942647-57-1

Bd. 339 MAO, Y.: 245 GHz Subharmonic Receivers For Gas Spectroscopy in SiGe BiCMOS Technology. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 339, Paderborn, 2015 – ISBN 978-3-942647-58-8

Bd. 340 DOROCIAK, R.: Systematik zur frühzeitigen Absicherung der Sicherheit und Zuverlässigkeit fortschrittlicher mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 340, Paderborn, 2015 – ISBN 978-3-942647-59-5

Bd. 341 BAUER, F.: Planungswerkzeug zur wissensbasierten Produktionssystemkonzipierung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 341, Paderborn, 2015 – ISBN 978-3-942647-60-1

Bd. 342 GAUSEMEIER, J.; GRAFE, M.; MEYER AUF DER HEIDE, F. (Hrsg.): 12. Paderborner Workshop Augmented & Virtual Reality in der Produktentstehung. HNI-Verlagsschriftenreihe, Band 342, Paderborn, 2015 – ISBN 978-3-942647-61-8

Bd. 343 GAUSEMEIER, J.; DUMITRESCU, R.; RAMMIG, F.; SCHÄFER, W.; TRÄCHTLER, A. (Hrsg.): 10. Paderborner Workshop Entwurf mechatronischer Systeme. HNI-Verlagsschriftenreihe, Band 343, Paderborn, 2015 – ISBN 978-3-942647-62-5

Bezugsadresse:
Heinz Nixdorf Institut
Universität Paderborn
Fürstenallee 11
33102 Paderborn