

Methoden zur Reduktion der Messlatenz von GOBO-Projektor-basierten 3D-Sensoren

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

vorgelegt der
Fakultät für Maschinenbau der
Technischen Universität Ilmenau

von
Master of Science Patrick Dietrich

1. Gutachter: Univ.-Prof. Dr. rer. nat. Gunther Notni
2. Gutachter: Prof. Dr.-Ing. Eduard Reithmeier
3. Gutachter: Univ.-Prof. Dr.-Ing. Patrick Mäder

Tag der Einreichung: 14.09.2021

Tag der wissenschaftlichen Aussprache: 16.06.2022

Dieses Werk ist lizenziert unter einer [Creative Commons „Namensnennung 4.0 International“](#) Lizenz.



URN: urn:nbn:de:gbv:ilm1-2022000263
DOI: 10.22032/dbt.53040

Kurzzusammenfassung

Genauere, optische 3D-Messverfahren werden vielfältig in der Industrie, der Medizin und der Wissenschaft eingesetzt. Bei etlichen dieser Anwendungen ist eine schnelle Reaktion auf Veränderungen in der Messszene erforderlich. Dies ist z.B. der Fall, wenn eine Maschine aus Sicherheitsgründen abgeschaltet oder angehalten werden muss oder eine direkte Rückmeldung an einen Menschen gegeben werden soll. Steht das 3D-Ergebnis der Messung nach hinreichend kurzer Zeit ab Beginn der Messung zur Verfügung, kann die Reaktion basierend auf diesem Ergebnis angestoßen werden.

Das Prinzip des GOBO-Projektor-basierten, aktiven Stereo-Sensors hat sich als genaues, optisches 3D-Messverfahren etabliert. Bei diesem Verfahren wird ein sich zeitlich änderndes, aperiodisches Streifenmuster auf das Messobjekt projiziert, während zwei kalibrierte Kameras jeweils eine Bildsequenz synchron aufnehmen. Innerhalb dieser Bildsequenzen werden anschließend die Abbilder von Objektpunkten, welche in beiden Kameras sichtbar sind, einander zugeordnet. Für jedes solche Paar wird dann die 3D-Koordinate des zugehörigen Objektpunktes trianguliert.

Das Verfahren erlaubt auch 3D-Aufnahmen mit speziellen Anforderungen, die von anderen, genauen 3D-Sensorprinzipien nur schwer erreicht werden. Dazu gehört die Messung mit speziellen Lichtwellenlängen, wie z.B. dem Nah-Infrarotbereich, womit blendfreie Vermessung ermöglicht wird, oder die Erfassung sehr schneller Prozesse, wie die Messung von Airbag-Entfaltungen.

Bisher war es jedoch nicht möglich, die 3D-Messergebnisse in so kurzer Zeit (z.B. 100 ms), d.h. mit so kurzer Messlatenz, zur Verfügung zu stellen, dass eine unmittelbare Reaktion auf eine Veränderung der Messszene erfolgen kann. Diese Verkürzung der Messlatenz ist das Ziel dieser Arbeit.

Es werden Methoden beschrieben und untersucht, mit denen die Messlatenz von GOBO-Projektor-basierten, aktiven Stereo-Sensoren auf unter 100 ms verkürzt werden kann. Die Verbesserungen konzentrieren sich auf zwei Bereiche: die schnelle Rekonstruktion des 3D-Modells aus den aufgenommenen Bildsequenzen und die Reduktion der Aufnahmezeit durch Verkürzung der Bildsequenz-Länge. Letztere wird mittels einer Optimierung der Musterprojektion ermöglicht, welche bei kurzen Bildsequenz-Längen eine erhebliche Reduktion unerwünschter Messartefakte bewirkt.

Abschließend werden mehrere Anwendungen gezeigt, die von diesen Verbesserungen profitieren.

Abstract

Accurate, optical 3D measurement techniques are widely used in industry, medicine and science. In some of the applications, a fast response to changes in the measurement scene is required. This is the case, for example, when a machine has to be stopped to avoid collision, or direct feedback to a human being must be provided. If the 3D result of the measurement is available after a sufficiently short time from the start of the measurement, the reaction can be triggered based on this result.

Active stereo 3D sensors based on GOBO projection have been established as an accurate optical 3D measurement method. The sensor projects a time-varying, aperiodic fringe pattern onto the measured object, while two calibrated cameras synchronously record an image sequence, each. After the recording, a 3D reconstruction algorithm searches for pixels pairs, which correspond to the same object point within these two image sequences. For each such pair, the algorithm triangulates the 3D coordinate of the associated object point.

With this method, special requirements can be accounted for which are difficult to achieve by other accurate 3D measurement principles. These include measurements with special light wavelengths, such as the near-infrared range, thus enabling glare-free, i.e. irritation-free measurement, or the measurement of very fast processes, such as the measurement of airbag inflation.

Until now, however, it has not been possible to provide 3D measurement results in such a short time (e.g. 100ms), i.e. with such a short measurement latency, that an immediate reaction to a change in the measurement scene can be made. Therefore, the goal of this dissertation is the reduction of the measurement latency.

In this dissertation, the author describes and investigates methods to shorten the measurement latency of GOBO projector-based active stereo sensors to less than 100ms. The improvements focus on two areas: fast reconstruction of the 3D model from the acquired image sequences, and reduction of the acquisition time by shortening the image sequence length. The latter is achieved by means of an optimization of the pattern projection, which results in a significant reduction of unwanted measurement artifacts for short image sequence lengths.

Finally, the author shows several applications that benefit from these improvements.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter der Abteilung *Bildgebung und Sensorik* am *Fraunhofer Institut für Angewandte Optik und Feinmechanik (IOF)* und war in den letzten Jahren eines der bestimmenden Themen in meinem Leben. Ich möchte mich bei all jenen bedanken, die mich auf meinem Weg zur Promotion unterstützt haben.

Zuallererst danke ich meinen Eltern, ohne die ich niemals so weit gekommen wäre. Sie haben mir ein sorgenfreies und behütetes Aufwachsen ermöglicht und mich stets zum Lernen und Erfahren ermuntert. Ich danke meiner Frau für ihr Verständnis, den Rückhalt, den sie mir gab, und die Zeit, die sie opferte, um mir das Schreiben zu ermöglichen.

Meinem Doktorvater, Prof. Dr. rer. nat. Gunther Notni, Leiter des Fachgebiets *Qualitätssicherung und Industrielle Bildverarbeitung* an der *Technischen Universität Ilmenau* gilt mein besonderer Dank für die Ermöglichung dieser Arbeit, viele wertvolle Ratschläge, die Anregungen und die hilfreichen Diskussionen.

Ich danke außerdem allen Mitgliedern der Promotionskommission, insbesondere den Gutachtern, Prof. Dr.-Ing. Eduard Reithmeier und Prof. Dr.-Ing. Patrick Mäder, dem Vorsitzenden Prof. Dr.-Ing. Stephan Husung, sowie Prof. Dr.-Ing. Eberhard Manske und Prof. Dipl.-Ing. Dr. med. (habil.) Hartmut Witte, ohne die die Promotion nicht möglich gewesen wäre.

Meinen Kollegen Martin Landmann und Dr. Stefan Heist danke ich für die zahlreichen fachlichen Diskussionen, die erheblich zu meinem wissenschaftlich-technischen Verständnis beigetragen haben.

Ich danke dem Leiter der Abteilung *Bildgebung und Sensorik* des Fraunhofer IOF, Dr. Peter Kühmstedt, und dem ehemaligen Leiter der Gruppe *Bildgebende Systeme*, Daniel Höhne, dafür, dass sie mir den Freiraum zur Arbeit an dieser Dissertation geschaffen haben.

PD Dr. Gerd Fabian Volk und Ellen Schötz von der Abteilung HNO des Universitätsklinikums Jena danke ich für die hilfreichen Informationen zum Einsatz des 3D-Sensors bei der Facialis-Parese-Behandlung.

Schließlich danke ich den Kollegen der Abteilung *Bildgebung und Sensorik* des Fraunhofer IOF für die gute Zusammenarbeit und die angenehme Arbeitsatmosphäre, insbesondere Dr. Peter Lutzke, Kevin Srokos, Ingo Gebhart, Roland Ramm, Dr. Christoph Munkelt, Matthias Heinze, Marc Preißler, Dr. Christian Bräuer-Burchardt, Marco Blumenthal, Simon Schindwolf, Kathrin Mäusezahl, Andy Tänzer, Henri Speck und Dr. Marcel Spehr.

Inhaltsverzeichnis

Kurzzusammenfassung	3
Abstract	4
Danksagung	5
Abbildungsverzeichnis	10
Tabellenverzeichnis	12
Verwendete Formelzeichen	13
Abkürzungsverzeichnis	16
1 Einleitung	17
1.1 Motivation	17
1.2 Überblick über existierende 3D-Sensorprinzipien	18
1.2.1 ToF-Kameras	19
1.2.2 LIDAR	19
1.2.3 Shape From Silhouette	19
1.2.4 Lichtschnittverfahren	20
1.2.5 Photogrammetrie	20
1.2.6 Passives Stereo-Sehen	21
1.2.7 Aktives Stereo-Sehen mit statischer Musterprojektion	21
1.2.8 Aktives Stereo-Sehen mit zeitlich veränderlicher Musterprojektion	22
1.2.9 Phasenschiebungs-Profilometrie	22
1.2.10 Aktives Stereo-Sehen mit aperiodischen Streifenmustern	23
1.3 Ziele und Gliederung dieser Arbeit	24
1.4 Kernthesen	25
2 Grundlagen	27
2.1 Stereosehen & Triangulation	27
2.2 Kamerakalibrierung	27
2.2.1 Extrinsische Kalibrierinformation	28
2.2.2 Intrinsische Kalibrierinformation	28
2.2.3 Kalibrierverfahren	29
2.3 Epipolargeometrie & Bildrektifizierung	30
2.4 Disparitäten, Disparitätskarte und Triangulation	32
2.5 3D-Rekonstruktion	33

2.6	GOBO-Projektion	34
2.7	Bildaufnahme und Kameras	35
3	Schnelle 3D-Rekonstruktion	37
3.1	Grundstruktur der 3D-Rekonstruktion	37
3.2	Vergleich mit verwandten Messprinzipien	38
3.3	Korrespondenzsuche mittels Normierter Kreuzkorrelation	40
3.4	Implementierung und Laufzeit des Referenzalgorithmus	41
3.5	Optimierte Korrespondenzsuche	43
3.5.1	Lösungsansatz	43
3.5.2	Konzept: schnelleres Ähnlichkeitsmaß	43
3.5.3	Vorberechnung der Binäreigenschaften	44
3.5.4	Konsistenzcheck und Median-Filter	45
3.5.5	Subpixel-genaue Korrespondenzverfeinerung	45
3.6	Eigenschaften der BF-Ähnlichkeit	46
3.6.1	Invarianz gegenüber Umgebungslicht und richtungsabhängiger Reflektivität	46
3.6.2	Statistischer Zusammenhang zwischen NCC und BF-Ähnlichkeit	47
3.7	Implementierung des schnellen Korrespondenzsuch-Algorithmus	47
4	Projektionsoptimierung	49
4.1	Projektionsmuster, Codewörter, Freiheitsgrade	50
4.2	Autokorrelationsanalyse	50
4.3	Verbesserung der Mustereindeutigkeit durch schnellere GOBO-Rotation	51
4.4	Optimierung der Projektionsmuster	53
4.4.1	Musterselbstähnlichkeit	53
4.4.2	Optimierungskonzept	53
4.4.3	Projektor-Kamera-Simulation	54
4.4.4	Ähnlichkeitseinpreisung	56
4.4.5	Mathematische Beschreibung der Kostenfunktion	58
4.4.6	Ableitung der Kostenfunktion	59
4.4.7	Iterative Anpassung der Streifenbreiten	61
4.5	Software-Tool zur Musteroptimierung	63
4.6	Autokorrelation der optimierten Muster	63
4.7	Zusammenfassung	64
5	Experimente und Untersuchungen	65
5.1	Sensorsystem und Messszenen	65
5.2	GOBO-Räder	66
5.3	Bildrauschen der Kameras	67
5.4	Messartefakte	69
5.5	Einfluss der optimierten Muster auf die Messgenauigkeit	72
5.6	Rechenzeit und Latenz	73
5.7	Erreichte Verbesserungen	73
6	Anwendungen	75
6.1	Einleitung	75
6.2	Sensor für medizinische Untersuchungen der Gesichtsmuskulatur und -nerven	75
6.2.1	Motivation: Krankheitsbild der Facialisparesie	75
6.2.2	3D"-Sensor zur Erfassung des Gesichts	76
6.2.3	Bestimmung von 3D"-Landmarken	77

6.2.4	Behandlung mittels Elektrostimulation	78
6.2.5	Interaktives Trainingssystem	79
6.2.6	Fazit und Ausblick	79
6.3	Sensornetzwerk zur Patientenüberwachung während der Bestrahlungstherapie	81
6.3.1	Motivation	81
6.3.2	Sensornetzwerk	82
6.3.3	Erreichte Verbesserungen	83
6.3.4	Atembewegungsüberwachung	84
6.3.5	Fazit	85
6.4	Weitere Anwendungen	86
6.4.1	Ausstellungsexponat für Mensch-Roboter-Interaktion	86
6.4.2	Sensor für "Crash"-Tests	87
6.4.3	Multimodaler 3D-sensor	90
7	Zusammenfassung und Ausblick	93
	Literatur	97
	Eigene referierte Veröffentlichungen	101
	Eigene Tagungsbeiträge	101
	Patentanmeldung	102
	Weitere Quellen	103

Abbildungsverzeichnis

1.1	Veranschaulichung Time of Flight (ToF)	19
1.2	Veranschaulichung Lichtschnittverfahren	20
1.3	Projiziertes, statisches Infrarot-Muster	21
1.4	Verschobene Sinusmuster für Phasenschiebungs-Profilometrie	22
1.6	GOBO-Rad.	23
1.5	Veranschaulichung des Vergleichs von Helligkeitssequenzen	23
1.7	GOBO-Projektor-Prinzip	24
1.8	Kernthesen dieser Arbeit	26
2.1	Veranschaulichung Triangulation	27
2.2	Veranschaulichung der Bildachsen u und v	27
2.3	Veranschaulichung der extrinsischen Kamera-Transformation	28
2.4	Veranschaulichung des Lochkameramodells	28
2.5	Bildschirmfoto der vom Autor entwickelten Kalibriersoftware.	30
2.6	Unrektifizierte und rektifizierte Kameras	30
2.7	Unrektifizierte und rektifizierte Bilder	31
2.8	Bildschirmfoto: Rektifizierung in der Kalibriersoftware.	32
2.9	Veranschaulichung: Disparität	32
2.10	Veranschaulichung des Zusammenhangs zwischen Tiefen- und Disparitätsintervall	33
2.11	GOBO-Projektor	34
2.12	GOBO-projizierte aperiodische Streifenmuster	35
3.1	Ablauf der 3D-Rekonstruktion.	37
3.2	Veranschaulichung der Dimensionen der rektifizierten Bildstapel.	41
3.3	Softwarearchitektur des 3D-Rekonstruktions-Framework	42
3.4	Rechenzeit mit Referenzalgorithmus in GPU-Implementierung	42
3.5	Aufteilung der 3D-Rekonstruktion in grobe und Subpixel-genaue Korrespondenzsuche.	43
3.6	Erzeugung von Binäreigenschaften mittels Vergleichs von Helligkeitswerten (Methode A) [54]	44
3.7	Erzeugung von Binäreigenschaften mittels Vergleichen von Helligkeitswerten mit dem Mittelwert (Methode B) [54]	44
3.8	Erzeugung von Binäreigenschaften aus Summen von Helligkeitswerten (Methode C) [54]	44
3.9	Anzahl der erzeugbaren Binäreigenschaften nach Erzeugungsmethode [54].	45
3.10	Korrelation zwischen NCC und BF-Ähnlichkeit	47
3.12	Erreichte Verbesserungen	48
3.11	Vergleich der Rechenzeiten	48
4.1	3D-Model mit unerwünschten Messartefakten	49
4.2	Zeitliche Autokorrelation des nicht optimierten GOBO-Musters bei kontinuierlicher Projektion	51
4.3	Veranschaulichung: gepulste Projektion	52
4.4	Zeitliche Autokorrelation des nicht optimierten GOBO-Musters bei gepulster Projektion	52
4.5	Konzept der Musteroptimierung	53

4.7	Symbolhafte Darstellung der Filter-Kernel	54
4.6	Veranschaulichung der 1D-Approximation des GOBO-Musters	54
4.8	Beispiel: Codewortzeugung aus gefiltertem Muster	55
4.9	NCC-Werte für paarweise Vergleiche des Beispielmusters	56
4.10	Gewichtungsfunktion $C(\rho)$	56
4.11	Realistisches Beispiel für NCC-Werte aller paarweisen Codewort-Vergleiche.	57
4.12	Kostenfunktion angewendet auf die NCC-Werte aus Abbildung 4.11; Gesamtkosten: $\bar{c} = 0,056$	57
4.13	Anpassung des Musters gemäß Kostenableitung	61
4.14	Visualisierung eines Optimierungslaufs	62
4.15	Kostenmatrix des optimierten Musters	62
4.16	Zeitliche Autokorrelation der von den Kameras aufgenommenen Helligkeitswerte eines optimierten GOBO-Musters bei kontinuierlicher Projektion	63
4.17	Zeitliche Autokorrelation eines optimierten GOBO-Musters bei gepulster Projektion	63
5.1	Sensor mit kontrastreicher Messszene	65
5.2	Messszene mit hellen Objekten	65
5.3	Sensor-Foto	66
5.4	GOBO-Projektor des Sensors mit abgenommenem Projektorgehäuse.	66
5.5	Empirisch bestimmtes Bildrauschen für Stereo-Kameras	68
5.6	(oben) Signal-zu-Rausch-Verhältnis für eine Bildsequenz der kontrastreichen Messszene. Gemessen mit GOBO K6 im gepulsten Projektionsmodus.	68
5.7	(rechts) Signal-zu-Rausch-Verhältnis auf dem Ebenennormal (siehe Abbildung 5.2) für alle untersuchten GOBO-Räder.	68
5.8	Verteilung der Messartefakte über die Messszene	70
5.9	3D-Modelle	71
5.10	Vergleich der Projektionsvarianten: Messartefakte mit BiCoS-Algorithmus	71
5.11	Vergleich der Projektionsvarianten: Messartefakte mit Referenzalgorithmus	71
5.12	Ebenenabweichung für verschiedene Messkonfigurationen	72
5.13	Messlatenz für verschiedene Bildsequenzlängen	73
5.14	Erreichte Verbesserungen	74
6.1	3D-Sensor für Gesichtsmessung	76
6.2	gemessene 3D-Gesichtsmodelle	77
6.3	Erkennung von 3D-Landmarken	77
6.4	Augenflächenberechnung mit Landmarken	78
6.5	3D-Untersuchung der Mimischen Muskulatur	78
6.6	Erreichte Verbesserungen	80
6.7	Patientenüberwachung bei der Strahlentherapie mittels dreier 3D-Sensoren	81
6.8	Verschachtelte Bildintegration	82
6.9	Datenfluss innerhalb des Sensornetzwerks	83
6.10	Testaufbau Atemüberwachung	84
6.11	Ergebnisse des Atemüberwachungstests	84
6.12	Erreichte Verbesserungen	85
6.13	Exponat zur Mensch-Roboter-Interaktion	86
6.14	Frontalansicht des Crash-Sensors	87
6.15	3D-Modelle eines Fahrzeug-Crash-Tests	88
6.16	Kalibrier-Target für Thermografie- und Stereo-Kameras	88
6.17	3D-Messung eines Fahrer-Airbags während der Entfaltung mit Temperaturwerten	89
6.19	Kalibriertarget	90

6.18 Frontansicht Multimodalsensor	90
6.20 3D-Mapping der multimodalen Kamerainformation	91

Tabellenverzeichnis

3.1 Unterschiede zwischen der Korrespondenzsuche bei der Phasenschiebungs- Profilometrie und beim GOBO-basierten Sensor-Prinzip.	38
3.2 Unterschiede zwischen passivem Stereo und GOBO-Projektor-basiertem An- satz bei der Pixelkorrespondenzsuche.	39
5.1 Sensoreigenschaften	66
5.2 Liste der GOBO-Räder	67
6.1 Sensoreigenschaften	77
6.2 Technische Daten der 3D-Sensoren	82
6.3 Ergebnisse des Atemüberwachungstests	84

Verwendete Formelzeichen

- $(\hat{X}, \hat{Y}, \hat{Z}, \hat{W})$ 3D-Punkt in homogenen Koordinaten.
- (c_u, c_v) *Kamerahauptpunkt*: Punkt auf dem Kamerasensor in Pixelkoordinaten, an dem die optische Achse des Objektivs auf den Sensor trifft.
- F Filtermatrix, die die Filterung des Musters \vec{p} mit dem Projektor-Kamera-Filter $F_{\sigma,b}$ beschreibt.
- $J_{F_{\sigma,b}}$ Matrix aller Partieller Ableitungen (Jakobimatrix) des Projektor-Kamera-Filters $F_{\sigma,b}(\vec{p})$
- $J_{S_{i,j}}$ Matrix aller Partieller Ableitungen (Jakobimatrix) der Samplingfunktion $S_{i,j}(\vec{f})$
- R Rotationsmatrix
- $\nabla \bar{c}$ Gradient der Gesamtkosten \bar{c} , d.h. Vektor der Länge L mit Richtungsableitungen nach den Werten $p_1, \dots, p_L = \vec{p}$
- \bar{c} Zu minimierender Gesamtkostenwert des Musters. Berechnet als Mittelwert der Kosten C aller Codewort-Vergleiche
- \bar{x} Mittelwert der Helligkeitssequenz \vec{x} . $\bar{x} = \frac{1}{N} \sum_{j=1}^N x_j$
- \bar{y} Mittelwert der Helligkeitssequenz \vec{y} . $\bar{y} = \frac{1}{N} \sum_{j=1}^N y_j$
- $\rho(\vec{x}, \vec{y})$ Normierte Kreuzkorrelation (NCC) der Helligkeitssequenz \vec{x} mit der Helligkeitssequenz \vec{y}
- σ Grad der simulierten Objektivunschärfe: Standardabweichung des Gaußfilters in simulierten Musterpixeln.
- σ_x Standardabweichung der Helligkeitssequenz \vec{x} . $\sigma_x = \sqrt{\frac{1}{N} \sum_{j=1}^N (x_j - \bar{x})^2}$
- σ_y Standardabweichung der Helligkeitssequenz \vec{y} . $\sigma_y = \sqrt{\frac{1}{N} \sum_{j=1}^N (y_j - \bar{y})^2}$
- \vec{a} Filterkoeffizienten der Projektor-Kamera-Filters $F_{\sigma,b}$. $\vec{a} = a_1, \dots, a_L$
- \vec{f} Pixelwerte des simulierten Streifenmusters, die mit dem Projektor-Kamera-Filter gefiltert wurden. $\vec{f} = f_1, \dots, f_L$
- \vec{p} **Kapitel 2:** Punkt in Weltkoordinaten $\vec{p} = (X, Y, Z)$
Kapitel 4: Binäre Pixelwerte des simulierten Streifenmusters. $\vec{p} = p_1, \dots, p_L$
- \vec{p}' Punkt in Kamerakoordinaten $\vec{p}' = (X', Y', Z')$

\vec{p}_0	Binäre Pixelwerte des initialen, binären Streifenmusters für die Optimierungsrechnung.
\vec{p}_i	Binäre Pixelwerte des binären Streifenmusters nach dem i -ten Optimierungsschritt.
\vec{t}	Position der Kamera in Weltkoordinaten
\vec{w}_i, \vec{w}_j	Helligkeitssequenzen/Codewörter. $\vec{w}_i = f_i, f_{i+r}, f_{i+2r}, \dots, f_{i+Nr}$
\vec{x}	Sequenz von Helligkeitswerten der Länge N . $\vec{x} = x_1, \dots, x_N$
\vec{y}	Sequenz von Helligkeitswerten der Länge N . $\vec{y} = y_1, \dots, y_N$
b	Breite des Boxcar-Filters in simulierten Musterpixeln, der die Bewegungsunschärfe durch GOBO-Rotation während der Lichtintegration der Kameras simuliert.
$C = C(\rho(\vec{x}, \vec{y}))$	Kosten-Gewichtungsfunktion. Einzelkosten des Vergleichs des Codewortes \vec{x} mit \vec{y} .
D	abzusuchendes Disparitätsintervall
d	Disparität. $d = u_l - u_r$
f	<i>Kamerakonstante</i> : Abstand des Kamerazentrums vom Sensor in Pixelgrößen.
$F_{\sigma, b}$	Projektor-Kamera-Simulationsfilter mit dem die Projektion des Musters und die Aufnahme mit den Kameras simuliert wird.
g	Nicht lineare Verzeichnungsfunktion. Simuliert als Teil der Kameraabbildungsfunktion die Objektivverzeichnung.
L	Länge des simulierten Streifenmusters in Pixeln
N	Bildsequenzlänge = Länge der Helligkeitssequenzen bzw. der Codewörter
n_{x_i}	Normierter Helligkeitswert berechnet aus dem Helligkeitswert x_i der Helligkeitssequenz \vec{x} . $n_{x_i} = \frac{x_i - \bar{x}}{\sigma_x}$
n_{y_i}	Normierter Helligkeitswert berechnet aus dem Helligkeitswert y_i der Helligkeitssequenz \vec{y} . $n_{y_i} = \frac{y_i - \bar{y}}{\sigma_y}$
P_i	i -tes Pixel auf der Bildzeile im rektifizierten, rechten Bildstapel
R	Faktor, um den die Reflektivität Richtung linker Kamera größer ist, als Richtung rechter Kamera.
r	Rotationsweite des Musters in simulierten Musterpixeln vom Beginn einer Bildaufnahme bis zum Beginn der folgenden.
$S_{i,j}(\vec{f})$	Samplingfunktion, die aus dem gefilterten Streifenmuster \vec{f} die beiden Codewörter \vec{w}_i und \vec{w}_j zusammensetzt. $S_{i,j}(\vec{f}) = (\vec{w}_i, \vec{w}_j)$
T	Abstand der Kamerazentren

t	Tiefe des Messvolumens
U	Umgebungslichtkomponente, von der Kamera zusätzlich zum gestreuten Projektionslicht aufgenommenes Umgebungslicht.
u	horizontale Pixelkoordinate
U_l	Anzahl der Pixelspalten des linken rektifizierten Kamerabildes = Breite des linken rektifizierten Kamerabildes in Pixeln.
u_l	horizontale Pixelkoordinate im Bild der linken (rektifizierten) Kamera
U_r	Anzahl der Pixelspalten des rechten rektifizierten Kamerabildes = Breite des rechten rektifizierten Kamerabildes in Pixeln.
u_r	horizontale Pixelkoordinate im Bild der rechten (rektifizierten) Kamera
V	Anzahl der Pixelzeilen der rektifizierten Kamerabilder
v	vertikale Pixelkoordinate
X	Raumkoordinate in Weltkoordinaten (zeigt vom Sensor aus nach rechts)
X'	Raumkoordinate in Kamerakoordinate (zeigt von der Kamera aus nach rechts)
Y	Raumkoordinate in Weltkoordinaten (zeigt vom Sensor aus nach unten)
Y'	Raumkoordinate in Kamerakoordinate (zeigt von der Kamera aus nach unten)
Z	Raumkoordinate in Weltkoordinaten (zeigt vom Sensor aus nach vorne)
Z'	Raumkoordinate in Kamerakoordinate (zeigt von der Kamera aus nach vorne)

Abkürzungen

BF Binary Feature/Binäreigenschaft. 44

BiCoS Binary Correspondence Search. 41, 42, 47

BRDF Bidirectional Reflection Distribution Function. 35

CPU Central Processing Unit/Hauptprozessor. 41

FPGA Field Programmable Gate Array. 21, 43

GOBO GOes Before Optics. 23, 34

GPU Graphic Processing Unit/Grafikprozessor. 25, 41

LIDAR light detection and ranging/laser imaging, detection, and ranging. 19

LWIR Long Wave Infrared/langwelliges Infrarotlicht (Wellenlängen: 3 μm bis 15 μm). 92

NCC Normalized Cross Correlation/Normierte Kreuzkorrelation. 40

NIR Nah-Infrarot-Licht (Wellenlängen: 750 nm bis 1000 nm). 34

PC Personal Computer. 25

SAD Summe Absoluter Differenzen. 39

SSD Sum of Square Differences/Summe der Quadratischen Differenzen. 39, 42

SWIR Short Wave Infrared/kurzwelliges Infrarotlicht (Wellenlängen: 1 μm bis 3 μm). 92

ToF Time of Flight. 19, 22

1.1 Motivation

In unterschiedlichen Gesellschaftsbereichen gibt es eine Vielzahl an Anwendungsfällen für die 3D-Vermessung von Personen und Objekten. Etliche dieser Anwendungsfälle erfordern zusätzlich, dass das digitale 3D-Abbild innerhalb kurzer Zeit, d.h. mit niedriger Messlatenz erzeugt wird. Dies ist überall da erforderlich, wo eine schnelle Reaktion auf die gemessene Situation erfolgen muss.

Ein wichtiger Bereich ist hierbei die Medizin. Ein Beispiel ist die Überwachung der Patientenlage während der Strahlentherapie. Bei dieser Anwendung muss sichergestellt werden, dass die Bestrahlung der korrekten Körperposition erfolgt. Jegliche Bewegung des Patienten muss mit möglichst kurzer Messlatenz erkannt werden. Als Reaktion kann die Bestrahlungsanlage nachgeführt oder angehalten werden.

Genauere 3D-Modelle von Händen können zur Steuerung von Geräten mittels Gesten genutzt werden, wenn sie mit hinreichend kurzer Messlatenz zur Verfügung stehen. Eine solche Gesteninteraktion hat den Vorteil, dass keine zusätzlichen Geräte, wie Datenhandschuhe, benötigt werden. Gestensteuerung dieser Art kann z.B. für die Steuerung von Schnittbilddarstellungen von Computer-Tomografie-Datensätzen für Infotainment- oder Navigationssysteme im Fahrzeug eingesetzt werden. Die Gestensteuerung kann dabei der Interaktion mit realen Gegenständen nachempfunden werden und kann somit sehr intuitiv gestaltet werden.

Ein weiterer Anwendungsbereich ist die Sicherheitstechnik. An Sicherheitstoren in Flughäfen werden Gesichter automatisiert analysiert und mit den Bildern in Pässen abgeglichen. Rein 2D-basierte Systeme können jedoch durch das Vorhalten von Fotos getäuscht werden. Die 3D-Information kann so biometrische Daten mit erhöhter Sicherheit liefern, weil die Gesichtsform miterfasst wird. Um einen hohen Personendurchsatz zu gewährleisten, muss das 3D-Ergebnis in kurzer Zeit verfügbar sein.

In der industriellen Produktion werden 3D-Verfahren zur Erfassung der exakten Geometrie von produzierten Teilen eingesetzt, um deren Qualität durch Vergleich mit zugrundeliegenden CAD-Modellen zu bestimmen. In der Vergangenheit war es nur möglich, einen kleinen Bruchteil der Teile stichprobenhaft auf diese Weise zu überprüfen. Schnelle 3D-Messverfahren bieten hier erstmals die Möglichkeit einer 100 %-Überprüfung beim Durchlauf der Teile.

Viele Prozesse in der Produktion werden heute mittels Industrie-Robotern durchgeführt. Diese können große Lasten schnell und präzise bewegen und positionieren. Damit stellen sie allerdings auch eine Gefahr für menschliche Arbeiter dar. Die häufigste Lösung für dieses Problem ist, Roboter-Arbeitsräume für Menschen zu sperren. Die Roboter werden vollständig gestoppt, wenn Menschen sich in deren Reichweite aufhalten. Eine Alternative stellen sogenannte *Cobots* dar: Sie sollen das Zusammenarbeiten von Menschen und Robotern in gemeinsamen Räumen und Bereichen ermöglichen. Allerdings dürfen Cobots

Definition 1.1.1

Die Messlatenz ist die Dauer zwischen dem Beginn der Messung und dem Bereitstellen des 3D-Modells.

nicht mit der gleichen hohen Geschwindigkeit wie Industrie-Roboter arbeiten und verlangsamen daher auch den Produktionsprozess. Zudem sind sie nicht zum Heben großer Lasten, wie z.B. Karosserieteilen, geeignet. Der Einsatz latenzarmer 3D-Messtechnik kann hier erhebliche Vorteile bringen: Die Position von Menschen, welche sich im Arbeitsraum aufhalten, kann kontinuierlich erfasst werden, und Roboter können ihre Geschwindigkeit je nach Abstand zu ihnen reduzieren oder bestimmte Bereiche umfahren. Dies erlaubt eine konstruktive Zusammenarbeit zwischen Menschen und Robotern, ohne dass auf die hohe Hebeleistung oder Geschwindigkeit großer Industrie-Roboter verzichtet werden muss.

Beim Fügen von Stahlteilen werden Teile oft wegen geringer Stückzahlen oder der hohen Bauteilkomplexität von Hand geschweißt. Deshalb sind zeit- und kostenintensive Prüfungen und zum Teil auch aufwändige Nachbearbeitungen erforderlich. Zudem wird die Anzahl der individualisierten Kleinteile, wie z.B. Auspuffanlagen, immer größer. Wünschenswert wäre also eine CAD-gestützte Robotisierung der Schweißprozesse. Um diese bewerkstelligen zu können, müssen die Schweißteile, insbesondere deren zu fügende Kanten und ihre genaue Lage im Raum, sehr genau bekannt sein. Die dazu notwendigen Daten kann ein latenzarmer 3D-Sensor liefern. Außerdem können damit Schweißstellen beim Schweißen kontinuierlich überwacht werden, sodass Fehler frühzeitig noch im Prozess erkannt und korrigiert werden können.

Für all diese Anwendungen aus der Industrie, der Medizin, der Mensch-Maschine-Interaktion und bei vielen weiteren muss also die Messlatenz kurz sein. Je nach konkreter Anwendung könnte die geforderte maximale Messlatenz z.B. bei 40 ms oder bei 300 ms liegen. Dies kann auch von der Reaktionszeit der auf die 3D-Messung folgenden Prozesse abhängen, z.B. wie schnell eine Anlage abgeschaltet oder nachgeführt werden kann.

Zudem ist es bei vielen Anwendungen auch notwendig, die Oberfläche des Messobjektes mit hoher Genauigkeit zu erfassen. Anwendungen, bei denen Menschen direkt vermessen werden, haben zudem oft die Anforderung der Irritationsfreiheit, d.h. die vermessenen Personen sollen von der Messung nicht irritiert, also insbesondere nicht geblendet werden.

Ein 3D-Sensor-System, welches diese Anforderungen erfüllt, kann in einigen Fällen eine bestimmte Anwendung erst ermöglichen, z.B. bei der Mensch-Maschine-Interaktion, in anderen die Sicherheit deutlich erhöhen, z.B. bei der Patientenüberwachung. In wieder anderen kann es helfen, die Kosten erheblich zu senken, indem es eine Automatisierung erst ermöglicht, oder die Qualität von Produkten erhöhen, wie im Beispiel der 100 %-inline-Produktkontrolle.

1.2 Überblick über existierende 3D-Sensorprinzipien

Es existiert eine Vielzahl an 3D-Sensorprinzipien, von denen einige hier vorgestellt werden und kurz bezüglich ihrer Tauglichkeit für die latenzarme, genaue Messung beschrieben werden.

1.2.1 ToF-Kameras

Time-of-Flight-(ToF)-Kameras [1, 2] senden eine Serie von Lichtpulsen aus und messen dann pixelweise die Zeitverzögerung, mit der das vom Messobjekt gestreute Licht wieder beim Sensor eintrifft. Mithilfe der bekannten Lichtgeschwindigkeit lässt sich daraus direkt die Entfernung zum Messobjekt berechnen. Abbildung 1.1 veranschaulicht dies.

ToF-Kameras können niedrige Messlatenzen von < 10 ms und hohe Messfrequenzen (> 100 Hz) erreichen. Allerdings ist die räumliche Auflösung sowohl lateral (< 0.1 Megapixel) als auch in der Tiefe (> 1 cm) begrenzt.

Bekannte Beispiele für eine ToF-Kameras sind die Microsoft-Kinect-Modelle ab ca. 2013 („Kinect for Xbox One“, „Kinect 2 for Windows“ und „Azure Kinect“).

Der Einsatz von ToF-Kameras zur Registrierung der Körperposition von Patienten in der Strahlentherapie wurde experimentell untersucht. [4] Dabei waren Messlatenzen von 65 ms möglich. Die Untersuchung konzentrierte sich auf die Erkennung der relativen Position und Rotation eines rigiden Gipspatientenmodells.

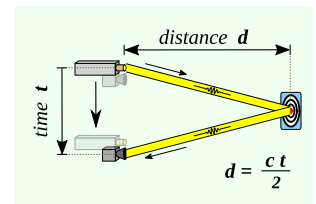


Abb. 1.1: Veranschaulichung Time of Flight (ToF): Die Entfernung d ergibt sich aus der Laufzeit t und der Lichtgeschwindigkeit c . [3]

1.2.2 LIDAR

LIDAR („light detection and ranging“ oder „laser imaging, detection, and ranging“)[5] ist eine 3D-Scanning-Methode, bei der mittels eines Laserstrahls ein Messobjekt sequenziell abgetastet wird. Bei jeder Abtastung wird die Entfernung des Objektpunktes, auf den der Laser trifft, aus der Laufzeit des Lichts ermittelt. Dies ist analog zur Messung bei ToF-Kameras. LIDAR wird unter anderem als Sensortechnik für autonome Fahrzeuge[6] und Flugdrohnen[7] sowie für die großflächige landschaftliche Vermessung sowohl vom Boden als auch vom Flugzeug aus verwendet[8, 9].

Ein wesentlicher Vorteil dieser Technik ist die hohe Reichweite, die bis zu mehreren Kilometern betragen kann. Ein Nachteil ist, dass zu keinem Zeitpunkt ein vollständiges Tiefenbild des Messobjektes entsteht. Stattdessen erfolgt die Aufnahme immer punktweise sequenziell. Latenzarme Messungen unter 100 ms können mit dieser Technik nur durchgeführt werden, wenn die Gesamtmesspunktanzahl klein gehalten wird (Größenordnung 1000 bis 10000 Punkte), z.B. nur linienhaft oder von sehr kleinen Flächen oder mit geringer Auflösung (> 10 cm). Messlatenzen können für die Aufnahmen geringer Punktmengen im Bereich weniger Millisekunden bis hin zu vielen Minuten für hoch aufgelöste Landschaftsmodelle liegen. Bei Aufnahmen mit niedriger Auflösung und Objekten mit hohem Detailgrad entsteht das Problem der Unterabtastung, wodurch erhebliche Artefakte in der aufgenommenen Objektgeometrie entstehen können.

1.2.3 Shape From Silhouette

Die Erzeugung von 3D-Modellen aus Silhouetten, die in den Bildern einer oder mehrerer Kameras erkannt wurden, wird als *Shape From Silhouette* bezeichnet. Wird zur Datenrepräsentation ein Voxel-Volumen genutzt, wird dies auch **Voxel Carving** genannt. Ein Review der Methode gibt es in [10].

Eine Möglichkeit, wie diese Methode eingesetzt wird, ist, ein Messobjekt vor einer Kamera zu rotieren und so Bilder aus mehreren Blickrichtungen auf das Objekt aufzunehmen. Dies wird z.B. erfolgreich eingesetzt, um 3D-Pflanzenmodelle zu erstellen [11, 12].

Alternativ kann das gleiche Sensorprinzip mit mehreren Kameras eingesetzt werden, die gleichzeitig jeweils eine Silhouette erkennen. In [13] wurde diese Methode verwendet, um reale Schauspieler in Echtzeit mit niedriger Auflösung (64^3 Voxel) in 3D zu erfassen und in virtuelle Szenen einzufügen.

In der Mehrkamera-Variante erlaubt das Messprinzip, bei niedrigen Auflösungen (> 5 cm) die Belegung eines Volumens mit kurzen Latenzen von unter 40 ms zu erfassen. Eine genaue Erfassung der Oberflächengeometrie kann damit aber nicht erfolgen.

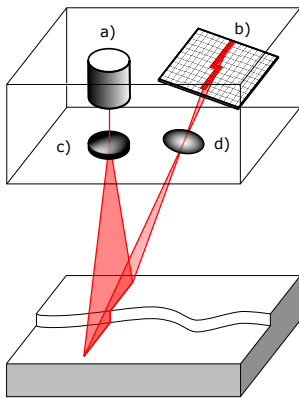


Abb. 1.2: Veranschaulichung Lichtschnittverfahren. a) Laserquelle, b) Kamerasensor, c) Laseroptik d) Kameraoptik[14]

1.2.4 Lichtschnittverfahren

Beim Lichtschnittverfahren wird ein Laserstrahl auf das Messobjekt projiziert und von einer Kamera aus einem festgelegten Winkel beobachtet. Abhängig von der Objektgeometrie verformt sich die Laserlinie aus der Perspektive der Kamera. Aus der Position der Laserlinie in den Kamerabildern kann die 3D-Geometrie des Messobjekts rekonstruiert werden. Die Laserlinie wird sequenziell über das Messobjekt geführt, wodurch ein vollständiger 3D-Scan entsteht [15]. In Abbildung 1.2 wird dies veranschaulicht.

Dieses Verfahren ermöglicht die genaue Erfassung von bis zu 30 Millionen 3D-Punkten pro Sekunde[15]. Allerdings wird nie zeitgleich ein komplettes 3D-Modell aufgenommen, sondern die Aufnahme erfolgt immer linienweise sequenziell. Dies kann bei bewegten Objekten zu Deformationsartefakten im 3D-Modell führen. Für die Erfassung einzelner Linien kann eine Messlatenz von unter 1 ms erreicht werden.

1.2.5 Photogrammetrie

Bei der Photogrammetrie handelt es sich um eines der ältesten optischen 3D-Messverfahren überhaupt: Erste Versuche fanden bereits in den 1860er Jahren statt[16]. Sie ist ein passives Verfahren, bei dem ein Messobjekt von mehreren Positionen aus mit einer oder mehreren Kameras fotografiert wird [17]. Im Nachgang werden in den Bildern Objektpunkte erkannt und einander zugeordnet. Dann wird zunächst eine Startlösung für die Kamerapositionen ermittelt, wobei hierbei Zusatzinformationen, z.B. aus Lage- und Beschleunigungssensoren, genutzt werden können. Mittels dieser Startlösung werden mit einem *Bündelblockausgleich*[18] genannten Verfahren sowohl die exakten Kamerapositionen als auch die Positionen der Objektpunkte ermittelt. Danach können verschiedene Verfahren angewendet werden, um ein dichtes 3D-Modell zu erhalten.

Ein wesentlicher Nachteil der Photogrammetrie ist, dass die verwendeten Algorithmen sehr rechenaufwändig sind (Größenordnung der Rechenzeiten auf PC-Hardware: Minuten bis Stunden). Deshalb ist dieses Verfahren nicht für die Anwendung der Echtzeit-Vermessung geeignet.

1.2.6 Passives Stereo-Sehen

Passives Stereo-Sehen[19–21] kann als Spezialfall der Photogrammetrie betrachtet werden. Der wesentliche Unterschied ist, dass zwei Kameras zum Einsatz kommen, deren relative Positionen zueinander bei der Messung schon bekannt sind. Aus diesem Grund können deutlich schnellere Auswerte-Algorithmen zum Einsatz kommen. (Siehe auch Abschnitt 3.2.) Ein Überblick über solche Algorithmen ist in [22] zu finden.

In [23] wurde gezeigt, dass mit einem optimierten Auswerte-Algorithmus und einer FPGA-Implementierung 30 3D-Modelle pro Sekunde bei 0,3 MPx Bildauflösung berechnet werden können, was auf eine Rechenzeit von ≤ 33 ms hindeutet. Dazu addiert sich die Bildaufnahmelatenz, die bei 30 Hz Bildeinzugsrate maximal 33 ms, bei guten Lichtverhältnissen, d.h. kürzeren Belichtungszeiten, auch deutlich weniger, z.B. 5 ms, beträgt.

Allerdings ist das Verfahren auf eine deutlich erkennbare Objekttextur oder -struktur angewiesen. Diese ist bei vielen industriell erzeugten Objekten nicht gegeben, da deren Oberflächen oft (zumindest stückweise) homogen sind. Das Gleiche gilt für die Vermessung von menschlicher Haut und etliche in Kleidung verarbeitete Stoffe.

1.2.7 Aktives Stereo-Sehen mit statischer Musterprojektion

Das Problem der fehlenden Objekttextur wird beim aktiven Stereo-Sehen mit statischer Musterprojektion durch den Einsatz eines Projektors gelöst, der ein zeitlich unverändertes Muster auf das Messobjekt projiziert [25, 26]. So können auch untexturierte Objekte vermessen werden. Beispiele für solche Sensoren sind der erste Sensor aus der *Kinect*-Produktreihe von Microsoft [27] und der *TrueDepth*-Sensor des *iPhone X* [28]. Diese projizieren ein zeitlich statisches Punktmuster in die Messszene (siehe Abbildung 1.3). Der Projektor ersetzt hierbei eine der Kameras, d.h. es wird trianguliert zwischen Projektor und einer Kamera. Ein weiteres Beispiel wird in [29] beschrieben, hierbei wird ein farbiges Streifenmuster projiziert.

Damit lassen sich Latenzen unter 50 ms erreichen, räumliche Auflösungen von unter 1 mm sind zwar potenziell möglich, es werden aber bei gleicher Kamera-Sensorauflösung nicht die räumlichen Auflösungen erreicht, die mit den in den Abschnitten 1.2.9 und 1.2.10 beschriebenen Multi-Shot-Verfahren möglich sind.

Wie alle anderen 3D-Sensorprinzipien mit aktiver Musterprojektion hat dieses Verfahren den Nachteil, dass zusätzliches Umgebungslicht störend wirken kann, wenn es das projizierte Licht überstrahlt. Im Extremfall kann dies die Messung unmöglich machen. So sind diese Verfahren meist nicht geeignet, um im prallen Sonnenlicht zu messen.

Ein ähnliches Problem stellen sehr dunkle oder spiegelnde Objekte dar. Diese streuen das projizierte Muster nicht oder in unzureichender Helligkeit zurück zu den Kameras. Wenn bei der Entwicklung des Sensors klar ist, welche Art Objekte vermessen werden sollen, kann dies ausgeglichen werden, indem spezielle Projektionslichtwellenlängen genutzt werden, in denen die Objekte eine für die Messung vorteilhafte Lichtstreuung aufweisen.



Abb. 1.3: Projiziertes, statisches Infrarot-Muster der *Microsoft Kinect*[24]

1.2.8 Aktives Stereo-Sehen mit zeitlich veränderlicher Musterprojektion

Anstatt ein einzelnes statisches Muster zu projizieren, können auch mehrere zeitlich aufeinanderfolgende, unterschiedliche Muster projiziert werden. Eine Klassifikation dieser Methoden ist in [30] zu finden. Ein neueres Review mit Fokus auf Echtzeit-Messung findet sich in [31].

Die zeitliche Änderung des Musters ermöglicht es, die Position des jeweiligen Projektor-Pixels in die projizierte Helligkeitssequenz zu kodieren. Aus diesem Grund ist es nicht mehr zwingend notwendig, eine Umgebung um das Pixel zu betrachten, um dieses eindeutig zu identifizieren. Eine genauere Analyse dieses Sachverhalts gibt es in Abschnitt 3.2.

In den folgenden Abschnitten 1.2.9 und 1.2.10 werden zwei wichtige Methoden, die auf diesem Prinzip basieren, beschrieben.

1.2.9 Phasenschiebungs-Profilometrie

Eines der verbreitetsten 3D-Sensorkonzepte ist die Phasenschiebungs-Profilometrie (weitere häufige Bezeichnungen: "phase-shift profilometry", "digital fringe projection with phase-shifting", "phase-shift interferometry")[32, 33], ein Verfahren, das mit mehreren aufeinanderfolgenden Aufnahmen und kodiertem Licht arbeitet.

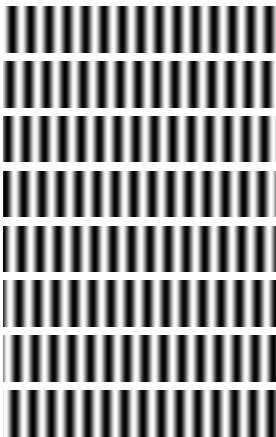


Abb. 1.4: Verschobene Sinusmuster für Phasenschiebungs-Profilometrie. Die Phasenwinkel der acht Muster sind jeweils um $2\pi/8$ weitergeschoben.

Eine Sequenz von Musterbildern wird mittels eines meist digitalen Projektors auf das Messobjekt projiziert. Es handelt sich dabei um (periodische) sinusförmige Streifenmuster, die bei den aufeinanderfolgenden Projektionen um jeweils einen festen Phasenwinkel verschoben werden [33]. Abbildung 1.4 zeigt eine solche Mustersequenz der Länge $N = 8$, wobei die Phase jeweils um $2\pi/8$ weitergeschoben ist. Eine oder mehrere Kameras machen jeweils eine Aufnahme des Messobjektes für jedes projizierte Musterbild. Die so erzeugte Sequenz aus Kamerabildern enthält also für jedes Pixel des Kamerasensors eine Sequenz von Helligkeitswerten.

Aus einer solchen Helligkeitssequenz kann direkt der Phasenwinkel ϕ berechnet werden, der die Position desjenigen Projektorpixels kodiert, welcher auf die aufgenommene Position des Messobjektes projiziert wurde. Für eine so gefundene Korrespondenz aus Kamera- und Projektor-Pixelkoordinaten wird mittels Triangulation zwischen Kamera und Projektor (oder alternativ zwischen mehreren Kameras) eine 3D-Koordinate berechnet.

Zu beachten ist dabei, dass eine aus ausschließlich Sinusmustern bestehende Musterbild-Sequenz das Projektorpixel lediglich innerhalb einer Wellenlänge des Musters kodiert. Aus diesem Grund ist es notwendig, entweder weitere Muster zu projizieren, die eine eindeutige Dekodierung über das gesamte Projektionsbild ermöglichen (z.B. Gray-Code-Sequenzen), oder es kommt alternativ ein Algorithmus zur Verstetigung des Phasenwinkels ϕ zum Einsatz, der Annahmen über die Objektgeometrie macht (z.B. Stetigkeit der Oberfläche)[34]. In [35] wurde gezeigt, dass ein grob aufgelöstes Tiefenmodell, welches mittels einer ToF-Kamera (siehe Abschnitt 1.2.1) aufgenommen wurde, genutzt werden kann, um eine Verstetigung des Phasenwinkels zu berechnen. Algorithmen zur Verarbeitung von Phasenschiebungs-Mustern sind ein Gebiet mit hoher Forschungsaktivität [36–38].

Die Phasenschiebungs-Profilometrie gilt als sehr genau. Tiefenaufösungen von besser als 10^{-4} mal der Diagonale des Messfeldes können in gut kalibrierten Systemen erreicht werden. Z.B. wurden in [39] $10\ \mu\text{m}$ in einem Messfeld mit einer 600 mm Diagonalen erreicht, also eine Tiefenauflösung von $1,67 \times 10^{-5}$ der Diagonalen.

Phasenschiebungs-Profilometrie erfordert aber einen sehr gut steuerbaren und kalibrierten Projektor. (Ein praktisches Kalibrierverfahren für Musterprojektoren wird in [40] vorgestellt.) Für Einsatzbereiche, für die keine kommerziellen Digitalprojektoren existieren, z.B. spezielle Wellenlängenbereiche oder extrem schnelle Projektion im Bereich über 1 kHz, macht dies das Verfahren meist unpraktikabel.

1.2.10 Aktives Stereo-Sehen mit aperiodischen Streifenmustern

Bei diesem Verfahren werden aperiodische Streifenmuster auf das Messobjekt projiziert, während zwei Messkameras synchron Bilder des Messobjekts aufnehmen [41–43]. Genau wie bei der Phasenschiebungs-Profilometrie entsteht so eine Helligkeitssequenz für jedes Kamerapixel, die als ein Code-Wort interpretiert werden kann, welches eine Position auf dem Messobjekt kodiert.

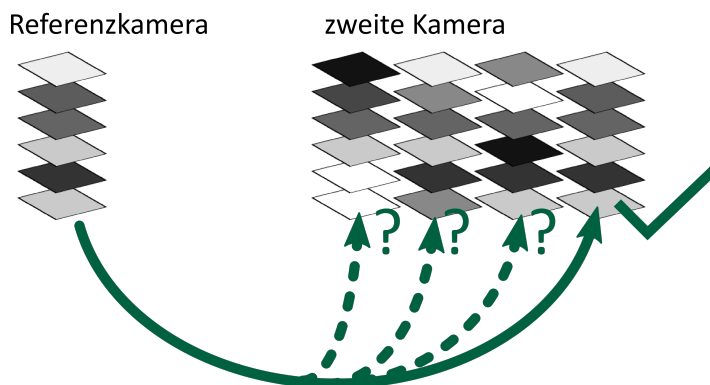


Abb. 1.5: Veranschaulichung des Vergleichs von Helligkeitssequenzen: Die Helligkeitssequenz (hier: Sequenzlänge $N = 6$) eines Pixels der Referenzkamera wird mit der vieler Kandidatenpixel in der zweiten Kamera verglichen, um eine Pixelkorrespondenz zu finden. Die rechte Helligkeitssequenz ist die ähnlichste und wird deshalb ausgewählt.

Im Gegensatz zur Phasenschiebungs-Profilometrie werden jedoch nicht die Projektorpixel-Koordinaten direkt aus den Helligkeitssequenzen, bzw. Codewörtern, berechnet, sondern es wird für jedes Codewort das ähnlichste Codewort in der anderen Kamera gesucht, um ein korrespondierendes Pixel-Paar zwischen den beiden Kameras zu finden. Für jedes so gefundene Paar kann dann eine 3D-Koordinate trianguliert werden.

Dieses Verfahren hat gegenüber der Phasenschiebungs-Profilometrie den Vorteil, dass es nicht erforderlich ist, die projizierten Muster bzw. Codewörter a priori zu kennen, was den Einsatz wesentlich einfacherer Projektoren ermöglicht. Dies ist insbesondere bei der Projektion in nicht sichtbaren Spektralbereichen vorteilhaft. So wurden unter anderem Projektoren, die mit thermischer Strahlung arbeiten, erfolgreich für die Vermessung ansonsten transparenter Materialien eingesetzt [44–49].

Am Fraunhofer IOF hat sich hierfür die Projektion mittels GOBO-Projektor (GOBO = GOes Before Optics) bewährt. Dabei handelt es sich um einen Projektor mit rotierendem Dia-Rad (GOBO). Das Prinzip ist in Abbildung 1.7 dargestellt. Das Dia-Rad (siehe Abbildung 1.6) enthält rein binäre, also abwechselnd durchsichtige und opake, aperiodische Streifen, d.h. Streifen mit

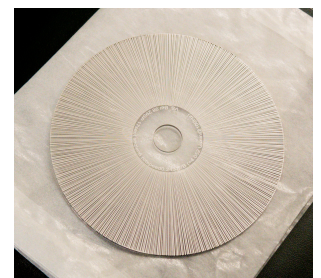
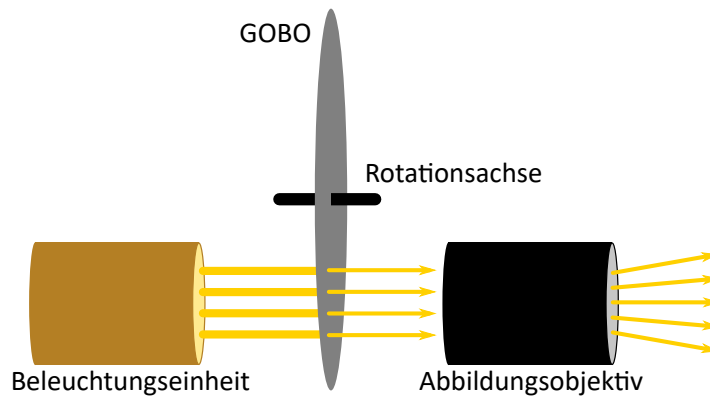


Abb. 1.6: GOBO-Rad.

unterschiedlichen Breiten. Durch Objektiv- und Bewegungsunschärfe entstehen so in den Kamerabildern aperiodische, quasi-sinusförmige Streifen [50]. Die Bewegungsunschärfe entsteht, weil die Kameras Licht integrieren, während das GOBO-Rad sich dreht, also während die Streifen weiterbewegt werden.

Abb. 1.7: GOBO-Projektor-Prinzip: Das Projektionslicht wird teilweise durch das Muster auf dem Dia-Rad (GOBO) geblockt, welches um seine Achse rotiert. Das nicht blockierte Licht wird mittels des Projektionsobjektivs auf die Messszene projiziert.



Diese Technik hat sich als gut funktionierend erwiesen, insbesondere für Anwendungen, bei denen kommerzielle Projektoren nicht verfügbar sind (z.B. erweiterte Spektralbereiche oder hohe Kamerabildraten von 10 kHz und höher). So wurden unter anderem Hochgeschwindigkeitsmesssysteme zur Messung schneller, menschlicher Körperbewegungen [43] oder Airbag-Entfaltungen [51, 52] realisiert.

Das Verfahren hat jedoch auch einen Nachteil bei der 3D-Rekonstruktionsrechnung: Während bei der Phasenschiebungs-Profilometrie eine Pixelkorrespondenz durch Dekodieren der Helligkeitswerte eines einzelnen Pixels gefunden werden kann, müssen bei dem GOBO-basierten System Pixelkorrespondenzen zwischen den Kameras durch die Suche nach der besten Übereinstimmung in einer größeren Anzahl von Kandidatenpixeln gesucht werden. Dies ist deutlich rechenaufwändiger.

1.3 Ziele und Gliederung dieser Arbeit

Ziel dieser Arbeit ist es, Methoden zu entwickeln, um die genaue, dreidimensionale Vermessung der Oberfläche realer Objekte und Personen so zu beschleunigen, dass eine Reaktion auf das Messergebnis so erfolgen kann, dass eine direkte Einflussnahme auf die gemessene Situation möglich wird. Als Zielmarke wird hierbei eine Latenz von 100 ms ab Beginn der Messung bis zur Bereitstellung des 3D-Modells angestrebt, welche für viele der oben beschriebenen Anwendungen ausreichend ist.

Der Hauptteil dieser Arbeit beschäftigt sich mit der Verringerung der Messlatenz von GOBO-Projektor-basierten 3D-Messsystemen. Deren Messlatenz (siehe Definition 1.1.1) hat zwei wesentliche Komponenten:

Definition 1.3.1

*Die **Aufnahmelatenz** ist die Dauer für die Aufnahme der Bildsequenzen.*

Definition 1.3.2

*Die **Rekonstruktionslatenz** ist die Dauer der 3D-Rekonstruktionsrechnung.*

Diese Arbeit ist folgendermaßen gegliedert:

Kapitel 2 beschreibt einige Grundlagen, die zum Verständnis der späteren Kapitel wichtig sind. In Kapitel 3 wird erläutert, wie die 3D-Rekonstruktionsrechnung beschleunigt werden kann. Dazu werden folgende zwei wesentliche Entwicklungen beschrieben:

1. eine schnelle, parallelisierte Implementierung der 3D-Rekonstruktion
2. ein neuer, deutlich schnellerer 3D-Rekonstruktions-Algorithmus

In Kapitel 4 wird beschrieben, wie mittels Optimierung der Musterprojektion eine erhebliche Reduktion der Bildsequenz-Länge und damit der Aufnahme-Latenz erreicht werden kann. Auch hier gibt es wieder zwei wesentliche Kernerrungenschaften, welche beide zu einer Verbesserung der Mustereindeutigkeit und damit der Verringerung von Messartefakten führen:

1. eine Verbesserung der Ansteuerung der Projektionsbeleuchtung
2. eine Optimierung der Muster auf dem Projektions-DIA (GOBO)

In Kapitel 5 werden mehrere Experimente und Ergebnisse vorgestellt, mit denen die Leistungsfähigkeit eines Messsystems bewertet wird, das die in Kapitel 3 und 4 entwickelten Neuerungen verwendet.

In Kapitel 6 werden zwei konkret umgesetzte Anwendungen latenzarmer 3D-Messtechnik vorgestellt. Dabei handelt es sich um medizinische Anwendungen, die jeweils mit einem GOBO-Projektor-basierten 3D-Sensor-System arbeiten. Außerdem folgen kurze Beschreibungen weiterer Anwendungen, welche zwar nicht direkt auf kurze Messlatenz angewiesen sind, die aber anderweitig von den während der Arbeit an dieser Dissertation gemachten Entwicklungen profitieren, z.B. von dem in Abschnitt 3.4 beschriebenen Software-Framework oder der Verringerung der Messartefakte.

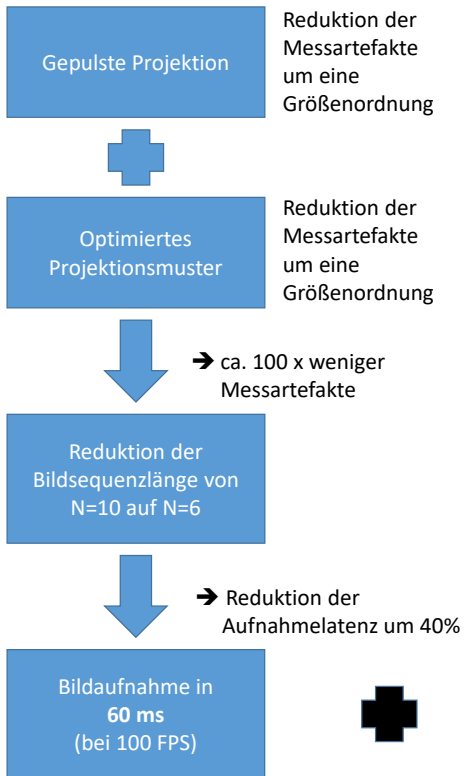
Einige Ergebnisse dieser Arbeit wurden bereits veröffentlicht. Insbesondere in [53–56].

1.4 Kernthesen

In dieser Arbeit werden die folgenden Kernthesen belegt (siehe auch Abbildung 1.8):

1. Mittels einer Parallelisierung und GPU-Implementierung des Rekonstruktions-Algorithmus kann eine **Rekonstruktionslatenz von 35 ms** auf aktueller PC-Hardware erreicht werden.
2. Mittels eines verbesserten Rekonstruktions-Algorithmus ist eine **weitere Beschleunigung auf 17 ms** möglich.
3. Mittels gepulster Musterprojektion kann bei kurzen Bildsequenz-Längen eine **Reduktion der Messartefakte um eine Größenordnung** erreicht werden.
4. Mittels einer Optimierung der Projektionsmuster kann eine **weitere Reduktion der Messartefakte um eine Größenordnung** erzielt werden.
5. Die Anwendung der gepulsten Projektion zusammen mit den optimierten Mustern führt zu einer Reduktion der Messartefakte um insgesamt 2 Größenordnungen und ermöglicht damit eine Reduktion der Bildsequenz-Länge auf $N = 6$ Bildpaare. Bei 100Hz Bildeinzugsrate entspricht dies einer **Aufnahmelatenz von 60 ms**.

Verkürzung der Aufnahmedauer → Kapitel 4



Beschleunigung der 3D-Rekonstruktion → Kapitel 3

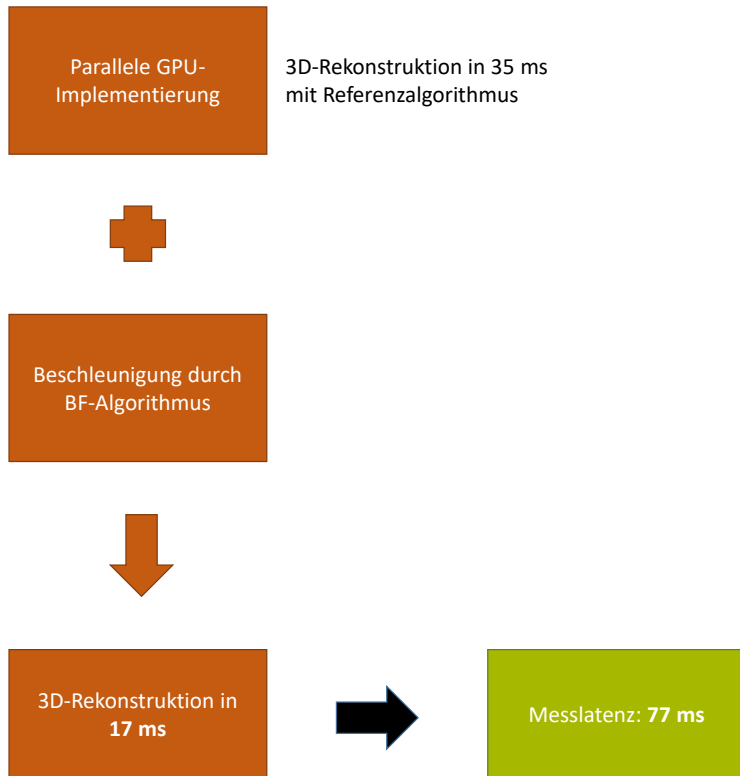


Abb. 1.8: Kernthesen dieser Arbeit

- Zusammengenommen ermöglichen diese Verbesserungen eine **Messlatenz von 77 ms** bei moderater Bildeinzugsrate von 100 Hz.

2.1 Stereosehen & Triangulation

Beim Stereo-Sehen wird ein Messobjekt mittels zweier Kameras von zwei unterschiedlichen Positionen aus betrachtet. Üblicherweise werden die Kameras dabei so angeordnet, dass beide Kameras in etwa den gleichen Abstand zum Messobjekt haben und die optischen Achsen ihrer Objektive einen Winkel kleiner 90° zueinander einnehmen.

In einem Kalibriervorgang werden die Positionen und Orientierungen der Kameras und außerdem ihre intrinsischen Abbildungseigenschaften bestimmt. D.h. es wird einerseits bestimmt, wo sich die Kameras befinden und in welche Richtung sie blicken. Andererseits wird für jedes Pixel auf beiden Kamerasensoren dessen Sichtstrahl in den Raum ermittelt. Details zu diesem Kalibriervorgang folgen in Abschnitt 2.2.

Jeder Punkt auf dem Messobjekt bildet mit den Abbildungszentren der beiden Kameras ein Dreieck im Raum (Abbildung 2.1). Für einen Punkt, der in den Bildern beider Kameras identifiziert wurde, ist es möglich, das Dreieck und damit die Position des Objektpunktes vollständig zu bestimmen. In anderen Worten: Für jedes gefundene Paar an Bildpunkten, jeweils einer in jeder Kamera, kann mittels der Kalibrierinformation eine 3D-Koordinate trianguliert werden.

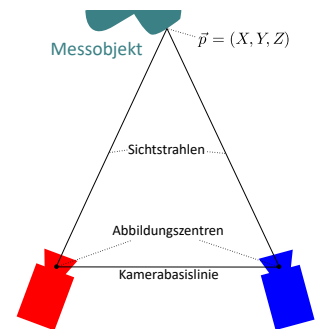


Abb. 2.1: Veranschaulichung der Triangulation: zwei Kameras beobachten einen Punkt P auf einem Messobjekt. Die Sichtstrahlen bilden mit der bekannten Kamerabasislinie ein Dreieck.

2.2 Kamerakalibrierung

Die Abbildung eines 3D-Punktes $\vec{p} = (X, Y, Z)$ auf den Sensor einer Kamera in Pixelkoordinaten (u, v) (siehe Veranschaulichung in Abbildung 2.2) kann durch eine Abbildungsgleichung beschrieben werden. Diese Abbildungsgleichung kann in zwei Teile aufgeteilt werden:

1. Die **extrinsische** Kameraabbildung berücksichtigt Position und Orientierung der Kamera. Sie bildet den Punkt \vec{p} (in Weltkoordinaten) auf einen Punkt \vec{p}' in einem Kamera-lokalen Koordinatensystem ab.
2. Die **intrinsische** Kameraabbildung berücksichtigt die Eigenschaften des Sensors und des Kameraobjektivs und bildet den Punkt \vec{p}' auf die Pixelkoordinate (u, v) ab.

Diese Teilabbildungen sind spezifisch für jede konkrete Kamera, d.h. sie sind für jede konkrete Kamera anders parametrisiert.

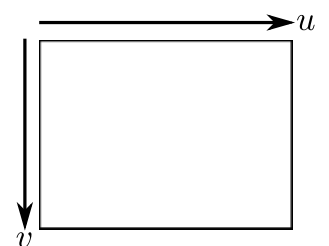


Abb. 2.2: Veranschaulichung der Bildachsen u und v . u repräsentiert horizontale Achse, d.h. die Bildspalte, v die vertikale Achse, also die Bildzeile.

Definition 2.2.1

Die Bestimmung der Kamera-spezifischen Parameter der Kameraabbildungsgleichungen wird als **Kamerakalibrierung** bezeichnet. Diese Parameter werden als **Kalibrierinformation** bezeichnet.

Um mittels eines Kamera-basierten 3D-Sensors die 3D-Geometrie eines Messobjekts aus den Kamerabildern zu rekonstruieren, ist es zwingend erforderlich, die Kamera-Abbildungsgleichungen zu kennen. Deshalb muss eine Kamera-kalibrierung durchgeführt werden. In den folgenden Abschnitten wird die Kamerakalibrierung beschrieben.

2.2.1 Extrinsische Kalibrierinformation

Als *extrinsische* Kalibrierinformation wird die Position und die Orientierung der Kamera relativ zu einem definierten Koordinatensystem bezeichnet. Diese Information hat pro Kamera 6 durch die Kalibrierung zu bestimmende Parameter: einen Punkt $\vec{t} = (t_x, t_y, t_z)$ im Raum, der die Position des Abbildungszentrums beschreibt, und eine Rotation, die sich z.B. mit 3 Eulerwinkeln oder einer Rotationsachse (2 Parameter) und einem Winkel beschreiben lässt.

Die extrinsische Kalibrierinformation beschreibt also eine Transformation von einem Weltkoordinatensystem in ein Koordinatensystem der Kamera. Dies ist in Abbildung 2.3 dargestellt.

Im Fall eines Stereo-Sensors kann eine der beiden Kameras das Koordinatensystem definieren, d.h. ihr Abbildungszentrum liegt im Koordinatenursprung und sie ist genau parallel zu den Koordinatenachsen orientiert. Die zweite Kamera wird dann relativ zu dieser Referenzkamera kalibriert. Für diesen Fall müssen also nur für die zweite Kamera die 6 Parameter bestimmt werden. Mittels der Rotation als Rotationsmatrix R und der Position als Translationsvektor \vec{t} kann ein Punkt \vec{p} folgendermaßen ins Koordinatensystem der jeweiligen Kamera transformiert werden:

$$\vec{p}' = R \cdot (\vec{p} - \vec{t}) \tag{2.1}$$

Die Achsen des lokalen Kamerakoordinatensystems von $\vec{p}' = (X', Y', Z')$ sind so orientiert, dass Z' den Abstand senkrecht zum Sensor vor der Kamera beschreibt. X' zeigt aus Sicht der Kamera nach rechts, Y' nach unten.

2.2.2 Intrinsische Kalibrierinformation

Als *intrinsisch* wird die Kalibrierinformation bezeichnet, die ausgehend vom Abbildungszentrum der Kamera für jedes Kamerapixel einen Sichtstrahl definiert, der auf dieses Pixel abbildet. In der Regel wird dabei aber nicht jedes Pixel durch einen eigenen Parametersatz beschrieben, sondern es wird ein erweitertes Lochkamera-Modell verwendet.

Die Lochkamera hat ein Abbildungszentrum in dem sich alle Sichtstrahlen schneiden. Die Abbildung eines Punktes $\vec{p}' = (X', Y', Z')$ auf die Pixelkoordinaten (u, v) des Kamerasensors einer idealen Lochkamera (siehe Abbildung 2.4) lässt sich mit einer linearen Abbildung beschreiben:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z'} \cdot \begin{pmatrix} X' \\ Y' \end{pmatrix} + \begin{pmatrix} c_u \\ c_v \end{pmatrix} \tag{2.2}$$

(c_u, c_v) wird als **Kamerahauptpunkt** bezeichnet. Der *Kamerahauptpunkt* ist der Punkt auf dem Kamerasensor in Pixelkoordinaten, an dem die optische Achse des Objektivs auf den Sensor trifft. Dieser liegt näherungsweise in der Mitte

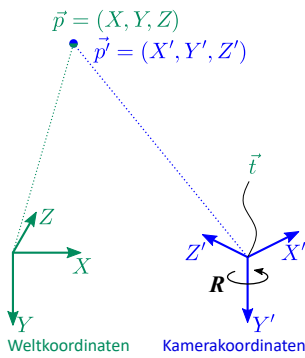


Abb. 2.3: Veranschaulichung der extrinsischen Kamera-Transformation: Die Koordinaten des Punktes \vec{p} im Weltkoordinatensystem (grün) werden transformiert, d.h. verschoben um den Vektor \vec{t} und rotiert mit der Rotation R . Dies resultiert in den Koordinaten $\vec{p}' = (X', Y', Z')$, die denselben Punkt im Kamerakoordinatensystem (blau) repräsentieren.

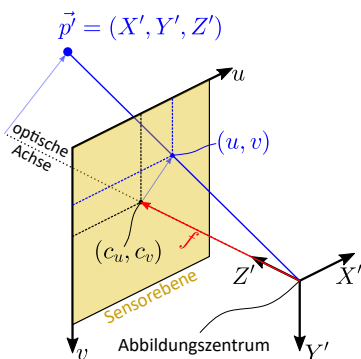


Abb. 2.4: Veranschaulichung des Lochkameramodells: Der Punkt \vec{p}' wird auf die Sensorebene projiziert, die sich im Abstand f zum Abbildungszentrum befindet.

des Sensors. Aufgrund von Fertigungstoleranzen der Kameras ist er bei realen Kameras aber zur tatsächlichen Mitte des Sensors verschoben (üblicherweise um weniger als 100 px). f ist die **Kamerakonstante**. f ist der Abstand des Abbildungszentrums vom Sensor in Pixelgrößen*. Diese Abbildung hat also drei durch die Kalibrierung zu bestimmende Parameter: f , c_u und c_v .

*: Hierbei werden quadratische Pixel vorausgesetzt.

Allerdings produzieren die meisten realen Kameraobjektive eine mehr oder weniger stark ausgeprägte **Bildverzerrung**, die sich in den Kamerabildern als gekrümmte Linien eigentlich gerader Objekte manifestiert. Deshalb wird das Lochkameramodell durch ein nicht-lineares Verzerrungsmodell erweitert. Ein verbreitetes Verzerrungsmodell ist das in [57] oder auch in [58] beschriebene. Dieses wird in Form einer nicht-linearen Verzerrungskorrektur-Funktion g auf den Punkt angewendet. Inklusiv Verzerrungskorrektur kann die intrinsische Abbildungsfunktion des Punktes $\vec{p}' = (X', Y', Z')$ auf die Pixelkoordinaten (u, v) folgendermaßen geschrieben werden:

$$\begin{pmatrix} u \\ v \end{pmatrix} = f \cdot g \left(\frac{1}{Z'} \cdot \begin{pmatrix} X' \\ Y' \end{pmatrix} \right) + \begin{pmatrix} c_u \\ c_v \end{pmatrix} \quad (2.3)$$

Auf die Details der Funktion g wird hier verzichtet, der Leser sei auf [57] und [58] verwiesen. g kann unterschiedlich parametrisiert werden, üblich sind ein bis fünf Verzerrungsparameter. Damit wächst die Anzahl der intrinsischen Kalibrierparameter auf vier bis acht.

Mit den Kameraabbildungsgleichungen 2.1 und 2.3 kann also für jeden 3D-Punkt die Position auf dem Kamerasensor berechnet werden, auf den dieser Punkt bei der Bildaufnahme abgebildet wird.*

*: Zusätzlich ist zu prüfen, ob sich der 3D-Punkt überhaupt vor und nicht hinter der Kamera befindet, d.h. ob $Z' > 0$ ist.

2.2.3 Kalibrierverfahren

Zur Durchführung der Kamerakalibrierung wurde vom Autor dieser Arbeit ein Software-Programm erstellt, in dem das in [58] beschriebene Kalibrierverfahren implementiert wurde. Abbildung 2.5 zeigt ein Bildschirmfoto des Programms. Die Kalibrierung mit diesem Programm erfolgt in fünf Schritten:

1. Aufnahme von Bildern eines planaren Kalibrierobjektes an mehreren Positionen (Die Aufnahme erfolgt gleichzeitig, synchron mit allen zu kalibrierenden Kameras.)
2. Erkennen von bekannten Objektpunkten in den Bildern des Kalibrierobjektes. Die Pixelkoordinaten dieser Objektpunkte zusammen mit deren bekannten 3D-Koordinaten relativ zum Kalibrierobjekt bilden die Eingabe für die folgenden Schritte.
3. Bestimmung der *intrinsischen* Abbildungsparameter aller Kameras
4. Bestimmung der *extrinsischen* Abbildungsparameter aller Kameras relativ zu einer Referenzkamera
5. gemeinsame Optimierung der *intrinsischen* und *extrinsischen* Parameter

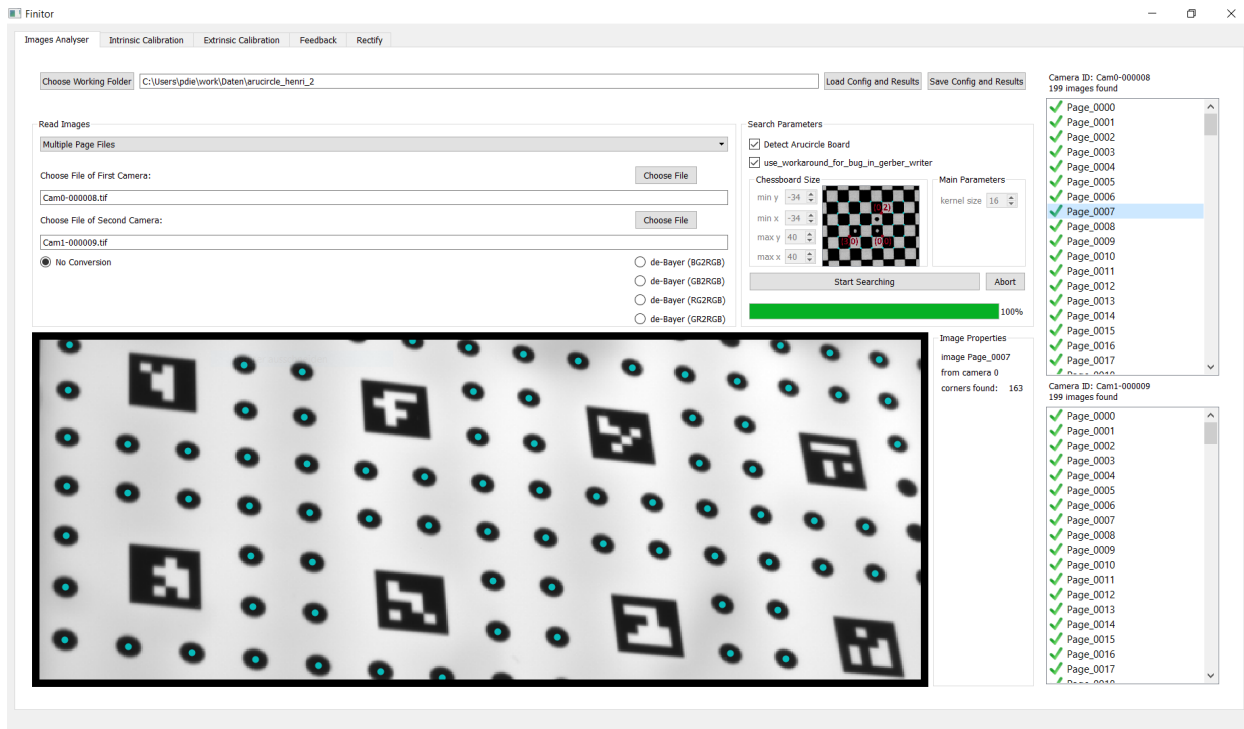


Abb. 2.5: Bildschirmfoto der vom Autor entwickelten Kalibriersoftware. Zu sehen ist das Bilderkennungsfenster. Links unten: Anzeige eines Ausschnitts aus einem Kalibrierbild mit Kalibriermuster und markierten Objektpunkten (türkis). Rechts: zwei Listen mit erfolgreich erkannten Kalibrierbildern.

2.3 Epipolareometrie & Bildrektifizierung

Durch jedes Dreieck, bestehend aus einem Objektpunkt und den Abbildungszentren der beiden Kameras, kann eine Ebene gelegt werden. Diese bildet sich auf die Bilder der Kameras jeweils als eine Gerade ab, da der Schnitt der Ebene mit den Sensorebenen jeweils eine Gerade ist (Abbildung 2.6 oben). Ein Paar solcher korrespondierender Geraden, eine im Bild der linken und eine im Bild der rechten Kamera, wird als Epipolarlinien bezeichnet [59].

Jeder Objektpunkt, der sich auf eine bestimmte Epipolarlinie in der einen Kamera abbildet, muss auf der dazugehörigen Ebene liegen und bildet sich deshalb in der anderen Kamera auf die korrespondierende Epipolarlinie ab. Das bedeutet, dass es ausreicht, für jeden gegebenen Bildpunkt in der einen Kamera den korrespondierenden Bildpunkt in der anderen Kamera auf der dortigen Epipolarlinie zu suchen. Andere Bildbereiche müssen nicht abgesucht werden.

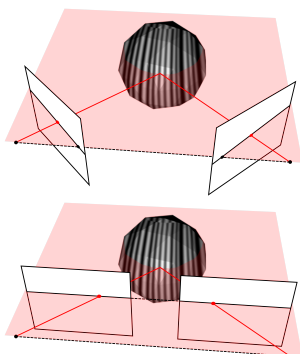


Abb. 2.6: Oben: Dreieck (rot) zwischen Abbildungszentren der Kameras und Objektpunkt in Sichtebene (rosa) und Epipolarlinien (schwarz) für nicht rektifizierte Kameras. Unten: gleiche Situation für rektifizierte Kameras.

Hierbei gibt es einen wichtigen Spezialfall: zwei ideale, verzeichnungsfreie Kameras mit dem gleichen Abbildungsmaßstab, deren optische Achsen exakt parallel stehen, deren Sensorzeilen exakt parallel ausgerichtet sind und deren Verbindungsline zwischen den beiden Abbildungszentren exakt im Rechten Winkel zu den optischen Achsen steht. Für diesen Spezialfall verlaufen die Epipolarlinien parallel zu den Bildzeilen der Kameras, und zwar so, dass ein Punkt, der sich in der einen Kamera auf Bildzeile v abbildet, auch in der anderen Kamera auf Bildzeile v zu finden ist. Jedes Paar zweier gleicher, idealer Kameras kann alleine durch Rotation der Kameras um ihr Abbildungszentrum in diesen Spezialfall überführt werden. In Abbildung 2.6 ist dies exemplarisch dargestellt.

Leider ist dieser Spezialfall in der Praxis kaum zu erreichen, da die Ausrichtung der Kameras sehr exakt erfolgen müsste, wobei auch Toleranzen innerhalb der Kameras zu berücksichtigen wären. Außerdem genügen die Objektive nicht dem hier angenommenen verzeichnungsfreien Ideal. Mittels der Kalibrierinformation ist es jedoch möglich, zwei ideale Kameras zu simulieren, d.h. die Bilder zweier realer Kameras im Nachhinein so zu entzerren, dass sie denen der idealen Kameras des Spezialfalls entsprechen. Diese Entzerrung wird als **Bildrektifizierung** bezeichnet [60]. Abbildung Abb. 2.7 zeigt ein Anschauungsbeispiel mit Originalkamera-Bildern und entsprechenden rektifizierten Bildern.

Um die Rektifizierung durchzuführen, können zwei Rektifizierungsmatrizen genutzt werden, eine je Kamera. Eine Rektifizierungsmatrix ist eine Abbildungsmatrix, welche für jedes Pixel im neuen, rektifizierten (d.h. idealen) Bild eine Koordinate auf eine Position im originalen (d.h. real aufgenommenen) Kamerabild enthält. Um ein rektifiziertes Bild zu erstellen, wird für jedes Idealpixel die zugehörige Koordinate aus der Rektifizierungsmatrix genutzt, um die Bildinformation aus dem originalen Kamerabild auszulesen und in das rektifizierte zu speichern. Da die Koordinaten nicht ganzzahlig sind, wird hierbei zwischen mehreren Bildpixeln des Originalbildes interpoliert, z.B. mittels bilinearer Interpolation der vier umliegenden Pixel.

Zu beachten ist, dass ein Paar Rektifizierungsmatrizen nicht eindeutig ist, da es mehrere Freiheitsgrade bei ihrer Erstellung gibt: eine Rotation des virtuellen Kamerapaares um die Gerade, auf der beide Abbildungszentren liegen, sowie die Skalierung und den Beschnitt der rektifizierten Bilder. Insbesondere haben die rektifizierten Bilder im Allgemeinen nicht mehr die Bilddimensionen der ursprünglichen Kamerabilder. Die Bilder der linken und der rechten rektifizierten Kamera haben die gleiche Anzahl Bildzeilen, können sich aber in der Bildbreite unterscheiden.

Die simulierten, rektifizierten Kameras haben neue Kalibrierinformationen, also neue Parameter der Abbildungsgleichungen. Und zwar hat jede Kamera einen neuen Kamerahauptpunkt (c_u, c_v) und beide die gleiche neue Kamerakonstante f , außerdem sind sie gegenüber den realen Kameras rotiert und haben deshalb jeweils eine neue extrinsische Rotation \mathbf{R} .

Die Berechnung der Rektifizierungsmatrizen aus den Kalibrierdaten wird hier aus Platzgründen nur skizziert:

1. Berechne für die beiden Kameras zwei Rotationsmatrizen \mathbf{R}_l und \mathbf{R}_r , die die Kameras so rotieren, dass ihre optischen Achsen genau parallel sind und sich die Kamerazentren in einer zu den optischen Achsen senkrechten Ebene befinden.
2. Wähle frei für jede der rektifizierten Kameras neue intrinsische Parameter f_n, c_{u_n}, c_{v_n} , wobei beide die gleiche Kamerakonstante f_n haben müssen.*
3. Für jedes Pixel (u_n, v_n) in der neuen, rektifizierten Kamera, wird die Pixelposition (u, v) im Bild der realen Kamera berechnet. Dazu wird für jedes Pixel (u_n, v_n) ein 3D-Punkt $(u_n - c_{u_n}, v_n - c_{v_n}, f_n)$ auf dessen Sichtstrahl gewählt. Dieser 3D-Punkt wird mittels \mathbf{R}_l (bzw. \mathbf{R}_r) ins Koordinatensystem der Originalkamera rotiert (ähnlich wie in Abbildung 2.3 dargestellt) und dann mittels der intrinsischen Kameraabbildungsgleichung (Gleichung 2.3) auf deren Sensor projiziert.
4. Für jedes Pixel (u_n, v_n) wird die Position (u, v) auf dem Kamerasensor der echten Kamera in die Rektifizierungsmatrix eingetragen.

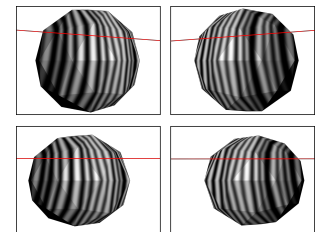


Abb. 2.7: Oben: original Kamerabilder mit schräg verlaufenden Epipolarlinien. Unten: rektifizierte Bilder mit horizontalen Epipolarlinien.

*: In der Praxis werden diese neuen intrinsischen Parameter und die Bildgrößen der rektifizierten Bilder so gewählt, dass der von den echten Kameras aufgenommene Bildausschnitt weitgehend mit dem der rektifizierten übereinstimmt.

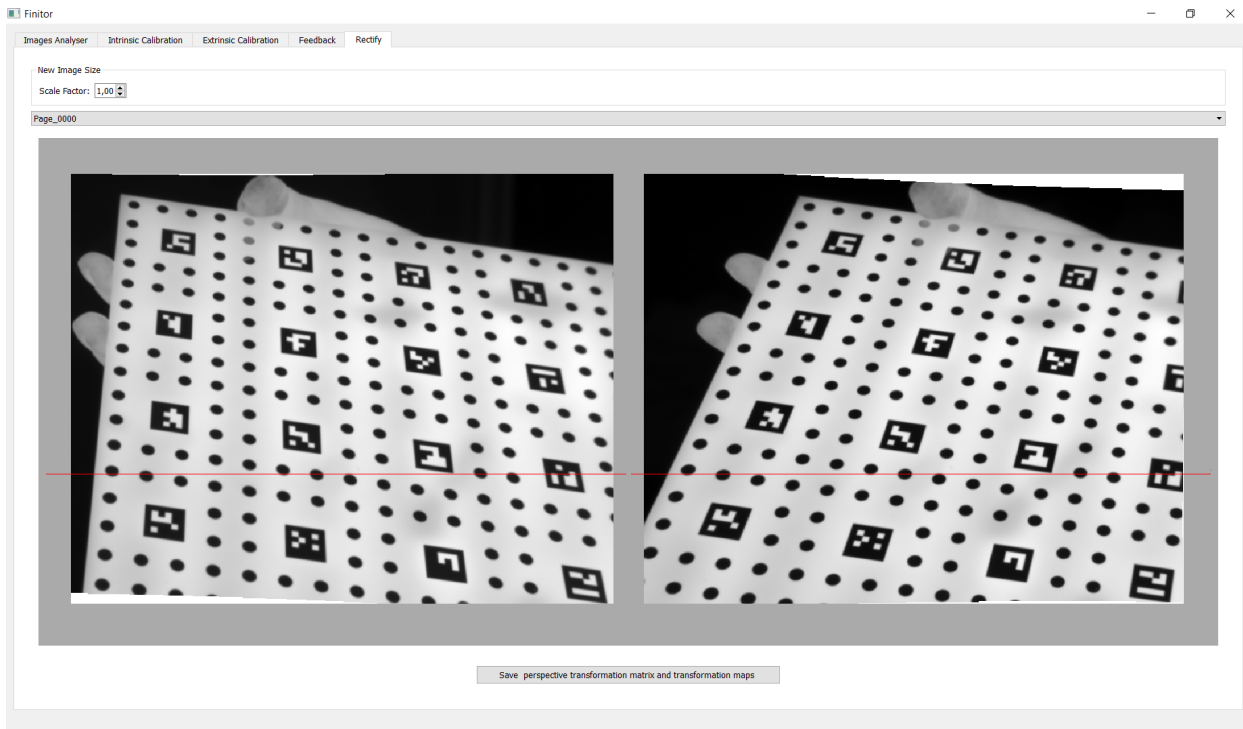


Abb. 2.8: Bildschirmfoto: Rektifizierung in der Kalibriersoftware. Dargestellt ist ein rektifiziertes Bildpaar. Die rote Linie kann mit der Maus vertikal verschoben werden, um zu überprüfen, ob die sichtbaren Objektpunkte in beiden Bildern auf der gleichen Bildzeile liegen.

Die in Abschnitt 2.2.3 beschriebene, vom Autor entwickelte Kalibrier-Software bietet die Möglichkeit, Rektifizierungsmatrizen für ein Kamerapaar zu erzeugen (siehe Bildschirmfoto in Abbildung 2.8).

2.4 Disparitäten, Disparitätskarte und Triangulation

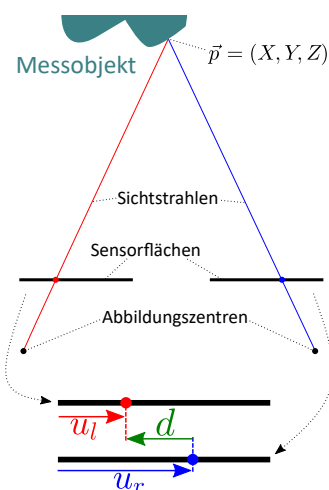


Abb. 2.9: Veranschaulichung der Disparität $d = u_l - u_r$. rot: Sichtstrahl der linken Kamera, blau: Sichtstrahl der rechten Kamera, u_l : Pixel-Spalte im linken Bild, u_r : Pixel-Spalte im rechten Bild

Die Rektifizierung der Kamerabilder hat den Vorteil, dass die Suche korrespondierender Punkte stark vereinfacht wird: Es muss immer nur auf einer Kamerazeile gesucht werden. D.h., um zu einem gegebenen Pixel auf Zeile v im linken rektifizierten Bildstapel das korrespondierende Pixel im rechten rektifizierten Bildstapel zu finden, muss dort lediglich die gleiche Bildzeile v durchsucht werden. Anders ausgedrückt: Jeder Objektpunkt bildet sich in der linken Kamera auf eine Pixelkoordinate (u_l, v) und in der rechten Kamera auf eine Pixelkoordinate (u_r, v) ab. Die Pixelkoordinaten haben die gleiche Bildzeile v , aber unterschiedliche Pixelspalten u_l bzw. u_r .

Definition 2.4.1

u_l und u_r seien Pixelspalten zweier Punkte im linken bzw. rechten rektifizierten Bild, die den gleichen Objektpunkt zeigen. Die Differenz

$$d = u_l - u_r$$

wird als **Pixeldisparität** oder einfach nur **Disparität** bezeichnet. (siehe Veranschaulichung in Abbildung 2.9.)

Für jede gefundene Pixelkorrespondenz wird die Disparität d berechnet. Das Ergebnis ist eine **Disparitätskarte**, ein Bild, welches an jedem Pixel einen Disparitätswert enthält. Wenn, wie hier beschrieben, die linke Kamera als Referenzkamera verwendet wird, also für jedes Pixel der linken ein zugehöriges Pixel der rechten Kamera gesucht wird, so hat die Disparitätskarte genau die Größe eines rektifizierten Bildes der linken Kamera.

Ist der abzusuchende Tiefenbereich im Messvolumen bekannt, so kann auch der abzusuchende Bereich auf der rechten Bildzeile eingeschränkt werden. D.h., dann gibt es eine minimale und maximale Disparität, die dem hinteren und vorderen Rand des Messvolumens entspricht, und es muss nur noch das Disparitätsintervall $D = d_{max} - d_{min}$ abgesucht werden (siehe Abbildung 2.10). Die dazu notwendige Information ist durch die Kalibrierung gegeben.

Für jeden Eintrag in der Disparitätskarte kann ein 3D-Punkt trianguliert werden. Hierfür ist aus der Kalibrierung eine projektive Abbildung bekannt, die von Pixelkoordinaten und Pixeldisparitäten auf 3D-Objektkoordinaten im Messvolumen abbildet. Diese Abbildung wird in Form einer Matrixmultiplikation mit der aus der Kalibrierung bekannten Matrix Q in homogenen Koordinaten[59] durchgeführt.

$$Q = \begin{pmatrix} 1 & 0 & 0 & -c_u \\ 0 & 1 & 0 & -c_v \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T & (c_u - c'_u)/T \end{pmatrix} \quad (2.4)$$

Dabei ist (c_u, c_v) der Kamerahauptpunkt der linken, (c'_u, c'_v) der der rechten rektifizierten Kamera, f die gemeinsame Kamerakonstante beider rektifizierter Kameras und T der metrische Abstand zwischen den Abbildungszentren.

Die Pixelkoordinaten (u_l, v) in der Disparitätskarte zusammen mit dem jeweiligen Disparitätswert d bilden sich auf den zugehörigen 3D-Punkt $\vec{p} = (X, Y, Z)$ ab. In homogenen Koordinaten wird dieser Punkt als $(\hat{X}, \hat{Y}, \hat{Z}, \hat{W})$ geschrieben.

$$Q \begin{pmatrix} u_l \\ v \\ d \\ 1 \end{pmatrix} = \begin{pmatrix} u_l - c_u \\ v - c_v \\ d \cdot f \\ (-d + c_u - c'_u)/T \end{pmatrix} = \begin{pmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \\ \hat{W} \end{pmatrix} \quad (2.5)$$

Die kartesischen Koordinaten (X, Y, Z) können mit folgender Rechnung aus den homogenen Koordinaten $(\hat{X}, \hat{Y}, \hat{Z}, \hat{W})$ berechnet werden. (Für eine detaillierte Beschreibung der homogenen Koordinaten sei auf [59] verwiesen.)

$$\vec{p} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \hat{X}/\hat{W} \\ \hat{Y}/\hat{W} \\ \hat{Z}/\hat{W} \end{pmatrix} \quad (2.6)$$

2.5 3D-Rekonstruktion

Als *3D-Rekonstruktion* wird im Allgemeinen die Berechnung der 3D-Geometrie aus aufgenommenen Sensordaten bezeichnet. Im Kontext dieser Arbeit wird

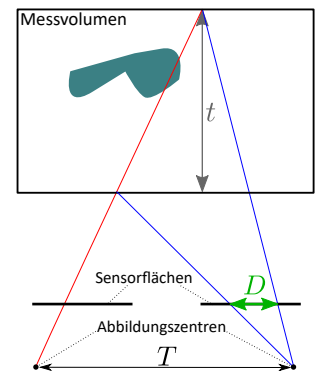


Abb. 2.10: Veranschaulichung des Zusammenhangs zwischen Tiefen- und Disparitätsintervall: Die Tiefenausdehnung des Messvolumens t legt das abzusuchende Disparitätsintervall D fest. Rot: Sichtstrahl der linken Kamera, blau: mögliche korrespondierende Sichtstrahlen der rechten Kamera. T ist der Abstand zwischen den Abbildungszentren.

der Begriff folgendermaßen eingeschränkt verwendet:

Definition 2.5.1

Als **3D-Rekonstruktion** wird die Berechnung von 3D-Punkten aus aufgenommenen Kamerabildern bezeichnet.

Die 3D-Rekonstruktion beinhaltet insbesondere die Suche von korrespondierenden Bildpunkten, d.h. Punkte, die durch Abbildung des gleichen Objektpunktes in jeder Kamera entstehen und die Triangulation des zugehörigen Objektpunktes mit Hilfe der Kalibrierinformation. Die hierfür notwendige Berechnung ist detailliert in Kapitel 3 beschrieben.

2.6 GOBO-Projektion

Um die Bildpunkte der beiden Kameras einander zuordnen zu können, müssen diese auf ihrer jeweiligen Epipolarlinie eindeutig identifiziert werden können. Hierbei gibt es verschiedene Möglichkeiten:

1. passiv: Das Messobjekt muss dazu eine hinreichend texturierte Oberfläche besitzen, um darauf Punkte eindeutig identifizieren zu können.
2. aktiv mit statischem Muster: Hierbei wird die Textur künstlich auf das Messobjekt aufgebracht.
3. aktiv mit dynamischem Muster: Der Projektor projiziert bei jeder konsekutiven Aufnahme ein anderes Muster. Dies führt für jeden Bildpunkt zu einer zeitlichen Helligkeitsänderung. Ist der so entstehende zeitliche Helligkeitsverlauf für den Objektpunkt eindeutig (innerhalb seiner Epipolarlinie), können seine beiden Bildpunkte mittels dieser Helligkeitssequenz einander zugeordnet werden. Damit dies funktioniert, müssen immer mehrere Bildpaare (z.B. 10) aufgenommen werden. Es muss aber nicht, wie im passiven Fall, ein Bildausschnitt um den Objektpunkt betrachtet werden, da keine Textur ausgewertet wird.

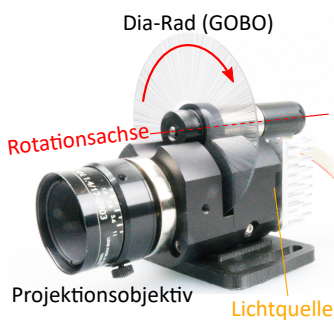


Abb. 2.11: GOBO-Projektor

Am Fraunhofer IOF wurde eine Familie von Projektoren entwickelt, die nach dem Prinzip des Dia-Projektors mit rotierendem Dia (GOBO) funktionieren. [43]. Abbildung Abb. 2.11 zeigt einen konkreten GOBO-Projektor. Die Achse dieses GOBO-Rads ist außerhalb der Projektionsfläche angeordnet. Deshalb rotiert das Muster weitgehend von links nach rechts durch das Projektionsfeld. Radial auf das GOBO-Rad aufgebrachte, undurchsichtige Streifen unterschiedlicher Breite erzeugen ein aperiodisches Streifenmuster. Da das GOBO-Rad sich gleichmäßig, kontinuierlich dreht, insbesondere auch während der Bildintegration der Kameras, wird daraus ein weichgezeichnetes Muster in den Kamerabildern. D.h., aus dem binären (abwechselnd transparenten und opaken) Muster auf dem GOBO-Rad entsteht wegen der Bewegungsunschärfe in den Kamerabildern ein nicht binäres Muster. Abbildung 2.12 zeigt mehrere Aufnahmen von GOBO-projizierten, aperiodischen Streifenmustern bei verschiedenen Rotationsgeschwindigkeiten.

Die einfache Konstruktionsweise des GOBO-Projektor-Prinzips erlaubt es, Projektoren leicht für Spezialanwendungen anzupassen. Insbesondere ist das Prinzip unabhängig von der Lichtwellenlänge verwendbar, z.B. für Nah-Infrarot-Projektion (NIR-Projektion), und benötigt kein kohärentes Licht. Hohe GOBO-Rotationsgeschwindigkeiten in Verbindung mit sehr lichtstarker Beleuchtung ermöglichen außerdem extrem schnelle Bildaufnahmefrequenzen [43, 51].

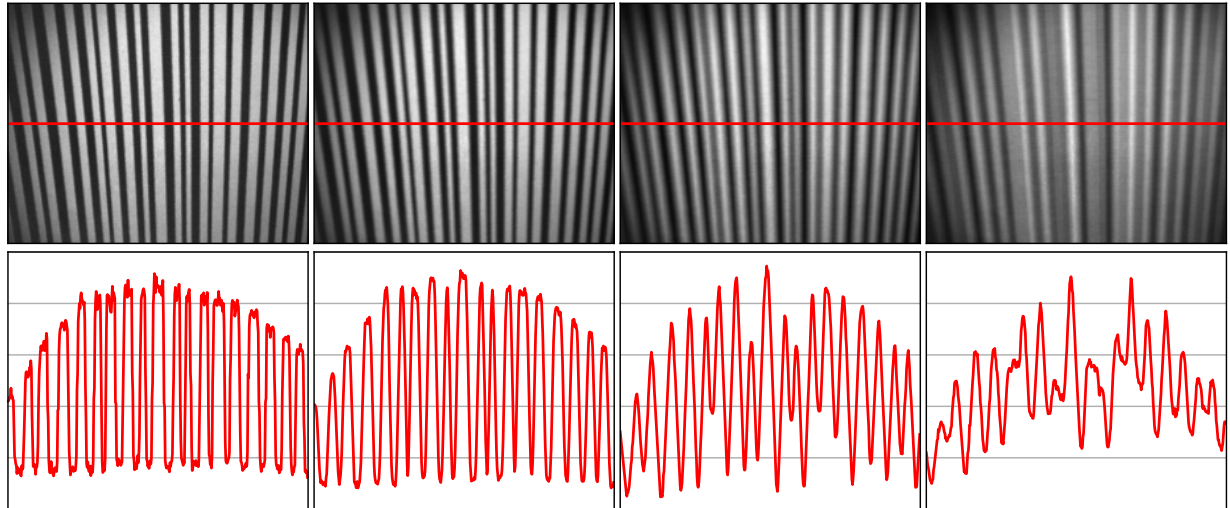


Abb. 2.12: GOBO-projierte aperiodische Streifenmuster. Oben jeweils ein Kamerabild, unten die Helligkeitswerte auf der mittleren Bildzeile. Ganz links: stillstehendes GOBO-Rad, 2. von links: 0.18° , 3. von links: 0.36° , rechts: 0.54° Rotation während der Bildintegration. Bei zu schneller Rotation (ganz rechts) bricht der Kontrast ein, d.h. die Differenz zwischen Tief- und Hochpunkten der Helligkeitskurve.

2.7 Bildaufnahme und Kameras

Das vom Musterprojektor projizierte Licht trifft auf das Messobjekt, wird von diesem gestreut und von den Kameras über ein begrenztes Zeitintervall integriert. Die Oberflächeneigenschaften des Messobjektes spielen bei der Streuung eine zentrale Rolle. Einerseits wird ein Teil des Lichtes absorbiert, andererseits wird das auftreffende Licht im Allgemeinen nicht gleichmäßig in alle Richtungen gestreut, sondern richtungsabhängig stärker oder schwächer. Diese richtungsabhängige Reflektivität lässt sich mittels einer bidirektionalen Reflexions-Verteilungsfunktion (Engl.: bidirectional reflection distribution function, BRDF) modellieren[61]. Für die in dieser Arbeit gemachten Betrachtungen ist es ausreichend, die Objektreflektivität mit einem konstanten Faktor je Kamera zu beschreiben. Das projizierte Licht wird also von einem Punkt auf der Oberfläche unterschiedlich stark zur linken und zur rechten Kamera zurückgestreut.

Zusätzlich zum Projektionslicht nehmen die Kameras Umgebungslicht auf, also Licht von anderen Lichtquellen, das am Messobjekt gestreut wird, sowie einen kleinen Anteil an Licht, das im Objektiv selbst gestreut wird, also nicht direkt vom eigentlich beobachteten Messobjekt kommt. Diese Lichtkomponenten sind für die in Kapitel 3 beschriebenen 3D-Rekonstruktionsverfahren störend, da sie nicht zur Identifikation von Objektpunkten beitragen, und werden deshalb von den Algorithmen entsprechend behandelt.

Damit beide Kameras das gleiche Muster aufnehmen, ist es erforderlich, dass sie genau gleichzeitig Licht integrieren. D.h., die Integrationszeit muss gleichzeitig beginnen und gleichzeitig aufhören. Ist dies nicht der Fall, gibt es einen Zeitversatz zwischen den Bildaufnahmen und die Kameras nehmen unterschiedliche projizierte Muster auf. Im Falle der GOBO-Projektion nimmt die dann später integrierende Kamera ein verschobenes Muster gegenüber der anderen auf. Dies führt bei der Punktkorrespondenzsuche zu Fehlzugeordnungen, die sich letztendlich in geometrischen Fehlern im resultierenden 3D-Modell manifestieren.

Damit eine genau gleichzeitige Bildintegration gewährleistet ist, muss die Integration beider Kameras durch ein **gemeinsames Trigger-Signal** ausgelöst und dann entweder zeitgesteuert oder durch ein zweites Signal beendet werden. Hierbei ist wichtig, dass die Kameras über einen sogenannten **Global Shutter** verfügen, d.h., die Belichtung muss für alle Pixel auf dem Sensor gleichzeitig beginnen und gleichzeitig wieder aufhören. Bei Kameras mit einem sogenannten *Rolling Shutter* ist dies nicht gegeben, dort beginnt die Integration der Pixel nacheinander. In fotografischen Aufnahmen von sich bewegenden Objekten manifestiert sich dies in geometrischen Verzerrungen dieser Objekte [62, 63].

Eine Möglichkeit, solche Rolling-Shutter-Kameras trotzdem für die GOBO-Projektor-basierte 3D-Rekonstruktion zu nutzen, besteht darin, das Projektionslicht synchronisiert mit der Bildintegration der Kameras an- und auszuschalten. Konkret wird die Bildaufnahme gestartet, während das Projektionslicht aus ist. Sobald dann alle Pixel des Sensors Licht integrieren, wird das Projektionslicht eingeschaltet. Und es wird wieder ausgeschaltet, bevor die Licht-Integration des ersten Pixels endet. Dies gewährleistet, dass alle Pixel des Sensors gleichzeitig das Projektionslicht aufnehmen. Lediglich das störende Umgebungslicht unterscheidet sich dann pixelweise. Dieses muss aber ohnehin algorithmisch behandelt werden.

In diesem Kapitel wird eine schnelle Methode zur 3D-Rekonstruktion beschrieben, deren Kern ein neuer Algorithmus für die Suche der Pixelkorrespondenzen zwischen den Kameras ist. Dies ist bei einem GOBO-Projektor-basierten 3D-Sensor der zeitintensivste Teil der Berechnung.

3.1 Grundstruktur der 3D-Rekonstruktion

Es wurden zwei 3D-Rekonstruktions-Algorithmen implementiert und untersucht, von denen der erste als der verbreitete Standardalgorithmus angesehen werden kann (siehe Abschnitt 3.3). Dem Autor ist allerdings keine Veröffentlichung bekannt, in welcher dieser Algorithmus im Detail beschrieben ist. Es ist also davon auszugehen, dass er sich je nach Implementierung mehr oder weniger stark von der hier beschriebenen Variante unterscheidet. Im Folgenden wird er als *Referenzalgorithmus* bezeichnet.

Der zweite, schnellere Rekonstruktions-Algorithmus, der im Rahmen dieser Arbeit neu entwickelt wurde, folgt der groben Struktur des Referenzalgorithmus, unterscheidet sich aber im rechenaufwändigsten Teil, nämlich der Korrespondenzsuche (siehe Abschnitt 3.5).

In beiden Fällen läuft die Messung und die 3D-Rekonstruktion nach folgendem Schema ab (Abbildung 3.1):

1. Bildaufnahme (siehe Abschnitt 2.7)
2. Rektifizierung der Bilder (siehe Abschnitt 2.3)
3. Korrespondenzsuche (siehe Abschnitt 3.3)
4. Triangulation (siehe Abschnitt 2.4)

Zunächst wird mit den beiden Kameras jeweils eine Bildsequenz einer festen Länge N (z.B. $N = 10$) aufgezeichnet. Und zwar wird sequenziell immer synchron ein Bildpaar nach dem anderen aufgenommen. Die Kameras werden hierfür mit einem gemeinsamen Hardware-Trigger-Signal ausgelöst und haben die gleiche Licht-Integrationszeit. Für jede Kamera entsteht ein Bildstapel der Länge N .

Die Bilder werden danach mittels zweier im Voraus erstellter Rektifizierungsmatrizen (siehe Abschnitt 2.3) rektifiziert. Die rektifizierten Bilder haben die gleiche Zeilenanzahl, und die Epipolarlinien verlaufen parallel zu den Bildzeilen. Ein Objektpunkt, der im linken rektifizierten Bildstapel auf Zeile v abgebildet wird, wird auch im rechten rektifizierten Bildstapel auf die Zeile v abgebildet.

Danach werden korrespondierende Pixelpaare zwischen den Kameras gesucht. Für jedes Pixel im linken rektifizierten Bildstapel wird ein Pixel im rechten rektifizierten Bildstapel gesucht, auf das derselbe Objektpunkt abgebildet wird. Um zu bewerten, welches Pixel das korrespondierende ist, wird ein Ähnlichkeitsmaß verwendet. Das ähnlichste Pixel wird ausgewählt. Dies wird ausführlich in Abschnitt 3.3 erläutert.

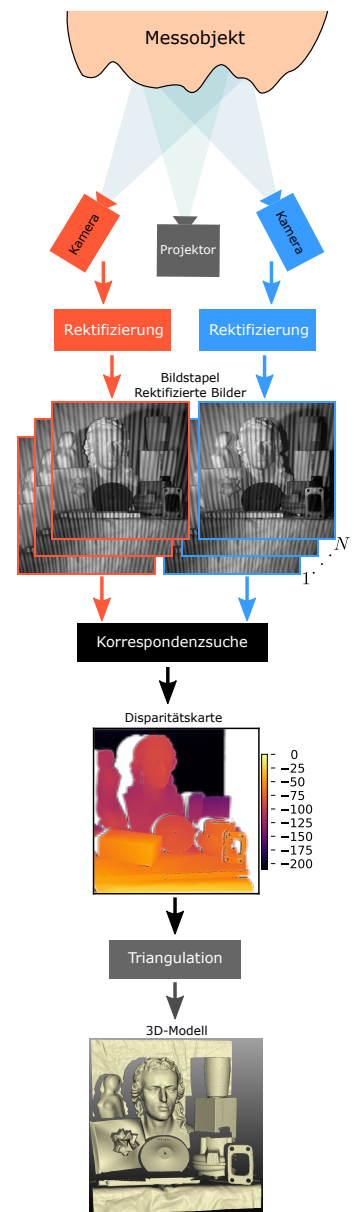


Abb. 3.1: Ablauf der 3D-Rekonstruktion.

Der Rest dieses Kapitels wird sich hauptsächlich mit diesem Teil des Rekonstruktions-Algorithmus beschäftigen, der Korrespondenzsuche und deren Optimierung.

Die Triangulation erfolgt wie in Abschnitt 2.4 beschrieben.

3.2 Vergleich mit verwandten Messprinzipien

Um die in den folgenden Abschnitten beschriebenen Verfahren der Korrespondenzsuche in Relation zur Korrespondenzsuche ähnlicher Messprinzipien zu setzen, werden hier kurz die Unterschiede zwischen dem GOBO-basierten Ansatz, der weit verbreiteten Phasenverschiebungs-Profilometrie und dem Passiven Stereo-Sehen bei der Suche nach korrespondierenden Pixeln beschrieben.

Die wichtigsten Unterschiede zwischen der Phasenschiebungs-Profilometrie und dem GOBO-basierten Ansatz sind in Tabelle 3.1 aufgelistet:

Tab. 3.1: Unterschiede zwischen der Korrespondenzsuche bei der Phasenschiebungs-Profilometrie und beim GOBO-basierten Sensor-Prinzip.

Phasenschiebungs-Profilometrie	GOBO-basierter Sensor
Es wird ein klar definiertes, vorher bekanntes Sinus-Muster auf das Objekt projiziert. (Für ein Muster mit Periode λ kann die Helligkeitssequenz $\vec{x} = x_1, \dots, x_N$ für das Pixel in Spalte u mittels $x_i = \frac{1}{2} + \frac{1}{2} \cdot \sin(\frac{u}{\lambda} + 2\pi \frac{i}{N})$ berechnet werden.)	Das Muster muss vorher nicht bekannt sein. Es muss lediglich hinreichend räumliche und zeitliche Varianz aufweisen. Ein aperiodisches Streifenmuster kann diese Kriterien erfüllen.
Zu jedem Pixel in der Kamera kann ein korrespondierendes Projektorpixel gefunden werden, indem die Helligkeitswerte dieses einen Kamerapixels dekodiert werden. Es müssen keine weiteren Pixel betrachtet werden.	Ein korrespondierendes Pixel muss mittels eines Ähnlichkeitsmaßes unter vielen Kandidaten im Bildstapel der zweiten Kamera gesucht werden.

Wegen des erheblichen Unterschieds in der Art der Korrespondenzsuche wird hier nicht weiter auf die Rekonstruktions-Algorithmen bei Phasenschiebesensoren eingegangen. Der interessierte Leser sei auf [36–38] verwiesen.

Für passive Stereo-Sensoren gibt es zahlreiche Korrespondenzsuch-Algorithmen [64, 65], von denen viele auf Echtzeitanwendungen abzielen. In Tabelle 3.2 sind die wesentlichen Unterschiede zum GOBO-basierten Ansatz beschrieben.

Passives Stereo	GOBO-Sensor
Alle Informationen für die Korrespondenzsuche müssen aus räumlichen Bildeigenschaften, z.B. Textur, Farbe, Objektkanten, Schatten etc., gefunden werden.	Die temporale Veränderung des Musters wird verwendet, d.h., für ein gegebenes Pixel in der linken Kamera kann ein korrespondierendes Pixel allein mittels der Helligkeitssequenz dieses Pixels gefunden werden, ohne dass dabei ein weiteres Pixel der linken Kamera betrachtet werden muss.
Korrespondenzen innerhalb gleichförmiger, texturloser Bildbereiche müssen aus umliegenden Bildbereichen interpoliert werden.	Die projizierten Muster ermöglichen es, unabhängig von der Objekttextur Pixelkorrespondenzen zu finden.
Räumliche Bildeigenschaften sehen von jeder der betrachtenden Kameras unterschiedlich aus, da sie Perspektivbedingt unterschiedlich projektiv auf den jeweiligen Sensor abgebildet werden. Die resultierende geometrische Veränderung zwischen den beiden Kamerabildern kann die Messgenauigkeit negativ beeinflussen und muss für genaue Messergebnisse algorithmisch behandelt werden.	Wenn ausschließlich temporale Pixeleigenschaften genutzt werden, d.h. ausschließlich die Helligkeitssequenzen einzelner Pixel, kann die Perspektiv-bedingte geometrische Veränderung vernachlässigt werden.
Das reflektierte Umgebungslicht kann von jeder Kamera aus gesehen eine unterschiedliche Intensität aufweisen, da die Reflektivität der meisten Objekte vom Winkel zur Beobachtungs- und Beleuchtungsposition abhängt.	Zusätzlich zum (unerwünschten) Umgebungslicht haben die aufprojizierten Muster eine unterschiedliche Intensität für jede Kamera. Die Objektreflektivität für projiziertes Licht und Umgebungslicht kann unterschiedlich sein, weil die Winkel zu den Lichtquellen jeweils unterschiedlich sind.

Tab. 3.2: Unterschiede zwischen passivem Stereo und GOBO-Projektor-basiertem Ansatz bei der Pixelkorrespondenzsuche.

Zum Finden von Pixelkorrespondenzen ist ein Ähnlichkeitsmaß erforderlich. Bei passivem Stereo werden mittels dieses Ähnlichkeitsmaßes Bildregionen in der linken mit Bildregionen in der rechten Kamera verglichen [65]. Dies ist erforderlich, weil die zur Zuordnung notwendige Information aus der Objekttextur innerhalb einer Region gezogen werden muss. Bei Stereo-Sensoren mit veränderlicher, aktiver Musterprojektion gäbe es diese Möglichkeit auch, aber es ist üblich, ausschließlich die Helligkeitssequenz eines einzelnen Pixels mit der Helligkeitssequenz eines Pixels der anderen Kamera zu vergleichen. D.h. es werden ausschließlich die zeitlichen Helligkeitsänderungen betrachtet, nicht aber die räumlichen, was höhere Genauigkeiten ermöglicht [66]. Kombinationen aus räumlichen und zeitlichen Bildeigenschaften können auch genutzt werden [67].

Als Ähnlichkeitsmaße für passives Stereo kommen beispielsweise die Summe Absoluter Differenzen (SAD), die Summe der Quadratischen Differenzen (engl.: sum of square differences, SSD) und die Normierte Kreuzkorrelation (engl.: Normalized Cross Correlation, NCC) zum Einsatz [22, 65, 68].

3.3 Korrespondenzsuche mittels Normierter Kreuzkorrelation

In diesem Abschnitt wird der *Referenzalgorithmus* beschrieben. Die hier beschriebenen Prinzipien sind ebenfalls für den neu entwickelten Algorithmus (Abschnitt 3.5) relevant.

Mittels des Projektors wurde ein sich zeitlich änderndes Muster auf das Messobjekt aufgebracht. Mit der Aufnahme von N konsekutiven Bildern resultiert daraus für jedes Kamerapixel eine eigene Helligkeitssequenz. Jeder Objektpunkt kann mit einer solchen Helligkeitssequenz beschrieben werden. Korrespondierende Pixel haben den gleichen Objektpunkt als Original und müssen demzufolge auch die gleiche Helligkeitssequenz aufweisen.

Um zu einem Pixel in der linken Kamera das korrespondierende Pixel aus einer Reihe von Kandidatenpixeln in der rechten Kamera zu finden, werden deren Helligkeitssequenzen mit der des gegebenen (linken) Pixels verglichen. Das ähnlichste Kandidatenpixel wird ausgewählt.

Für diesen Vergleich ist ein Maß erforderlich, welches für ein Paar von Helligkeitssequenzen deren Ähnlichkeit quantifiziert. In [69] wurden mehrere Ähnlichkeitsmaße für den Vergleich von Helligkeitssequenzen untersucht. Dabei stellte sich die *Normierte Kreuzkorrelation* (engl.: *normalized cross correlation*, NCC) bezüglich der Messgenauigkeit unter den untersuchten Maßen als optimal heraus.

Definition 3.3.1

Seien $\vec{x} = x_1, \dots, x_N$ und $\vec{y} = y_1, \dots, y_N$ zwei Sequenzen von Helligkeitswerten. Die *Normierte Kreuzkorrelation* (NCC) ρ ist definiert als:

$$\rho(\vec{x}, \vec{y}) = \frac{1}{N} \sum_{i=1}^N (n_{x_i} \cdot n_{y_i}) \quad \text{mit} \quad n_{x_i} = \frac{x_i - \bar{x}}{\sigma_x} \quad (3.1)$$

wobei $\bar{x} = \frac{1}{N} \sum_{j=1}^N x_j$ der Mittelwert und $\sigma_x = \sqrt{\frac{1}{N} \sum_{j=1}^N (x_j - \bar{x})^2}$ die Standardabweichung von \vec{x} sind.

Die NCC nimmt Werte zwischen -1 und $+1$ an, wobei $+1$ die höchstmögliche Ähnlichkeit der Sequenzen \vec{x} und \vec{y} bedeutet.

Unerwünschtes Umgebungslicht und eine blickwinkelabhängige Reflektivität des Messobjektes verändern die von den Kameras aufgenommenen Helligkeitssequenzen, und zwar für jede der beiden Kameras unterschiedlich. Vorausgesetzt das Umgebungslicht und die Objekt-Reflektivitäten sind zeitlich konstant, ist die NCC in der Lage, davon unabhängig die Ähnlichkeit zu bewerten: Die Subtraktion des Mittelwertes \bar{x} und das Teilen durch die Standardabweichung σ_x entfernt unerwünschte Umgebungslichtkomponenten U und gleicht blickwinkelabhängige Unterschiede der Messobjekt-Reflektivität R aus:

$$\rho(\vec{x}, \vec{y}) = \rho(R \cdot \vec{x} + U \cdot \vec{1}, \vec{y}) \quad (3.2)$$

Dies gilt allerdings nur im linearen Arbeitsbereich der Kameras. Bei Unter- oder Überbelichtung ist dies i.A. nicht mehr der Fall. In [70] gibt es eine Untersuchung des Einflusses nicht-linearer Kamerakennlinien auf die Korrespondenzsuche.

Ein Nachteil der NCC für die schnelle 3D-Rekonstruktion ist, dass sie rechenaufwändig ist. Die Normierungen (Subtraktion des Mittelwertes und Teilen durch die Standardabweichung) können als einmalige Vorberechnung ausgeführt werden. Ohne sie sind bei einer Bildsequenzlänge N für jeden Pixelvergleich N Multiplikationen und $N - 1$ Additionen zur Berechnung der NCC erforderlich.

Problematisch ist dies deshalb, weil sehr viele Pixelvergleiche durchgeführt werden müssen: Für jedes Pixel im linken Bildstapel müssen im Extremfall alle Pixel auf der gleichen Zeile im rechten Bildstapel verglichen werden. Dies bedeutet, dass bei rektifizierten Bildern der Größen $U_l \times V$ Pixel und $U_r \times V$ Pixel insgesamt $V \cdot U_l \cdot U_r$ mal die NCC berechnet werden muss (siehe Abbildung 3.2). Beispielsweise müsste die NCC für rektifizierte Bilder der Größen 1200×1000 und 1100×1000 insgesamt maximal $1000 \cdot 1200 \cdot 1100 = 132$ Millionen mal berechnet werden, was bei einer Bildsequenzlänge $N = 10$ insgesamt 1,32 Milliarden Multiplikationen beinhalten würde.

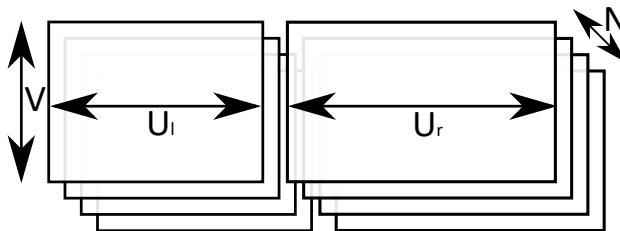


Abb. 3.2: Veranschaulichung der Dimensionen der rektifizierten Bildstapel.

Dieses Problem wird zwar je nach Sensorgeometrie und Messvolumen leicht entschärft, da oft nicht die gesamte Zeile im rechten Bildstapel abgesucht werden muss, sondern nur ein eingeschränktes Disparitäts-Intervall, welches einem bestimmten Entfernungintervall im Messvolumen entspricht (siehe Abbildung 2.10). Das Grundproblem bleibt jedoch erhalten.

3.4 Implementierung und Laufzeit des Referenzalgorithmus

Der Referenzalgorithmus verwendet die NCC als Ähnlichkeitsmaß und wurde mit OpenCL [71] für die Berechnung auf der Grafikkarte (GPU) implementiert. Aus der Verwendung von OpenCL resultiert eine hohe Flexibilität bezüglich der eingesetzten Rechenhardware. Es sind lediglich hardwarespezifische OpenCL-Treiber erforderlich. Diese gibt es z.B. für GPUs von AMD und NVIDIA, aber auch für die CPUs von Intel und AMD.

Die Berechnung auf der GPU läuft parallelisiert ab. Sie hat erhebliche Vorteile gegenüber der CPU-Berechnung, da der Algorithmus gut parallelisierbar ist, weil im Wesentlichen für jedes Pixel im linken Bildstapel die gleichen Operationen durchgeführt werden müssen.

Der OpenCL-Code wird von einem vom Autor dieser Arbeit entwickelten C++-Framework aus gestartet (siehe Abbildung 3.3). Ein zentrales *Controller*-Modul bezieht Kamerabilder über ein *Image-Provider*-Modul, fasst diese zu Stereo-Bildsequenzen zusammen und gibt sie an ein *Rekonstruktionsmodul* weiter. Dieses führt die 3D-Rekonstruktion auf der GPU (OpenCL) durch, wozu einer von zwei Algorithmen, nämlich der Referenzalgorithmus (Abschnitt 3.3) oder der BiCoS-Algorithmus (Abschnitt 3.5), zum Einsatz kommt.

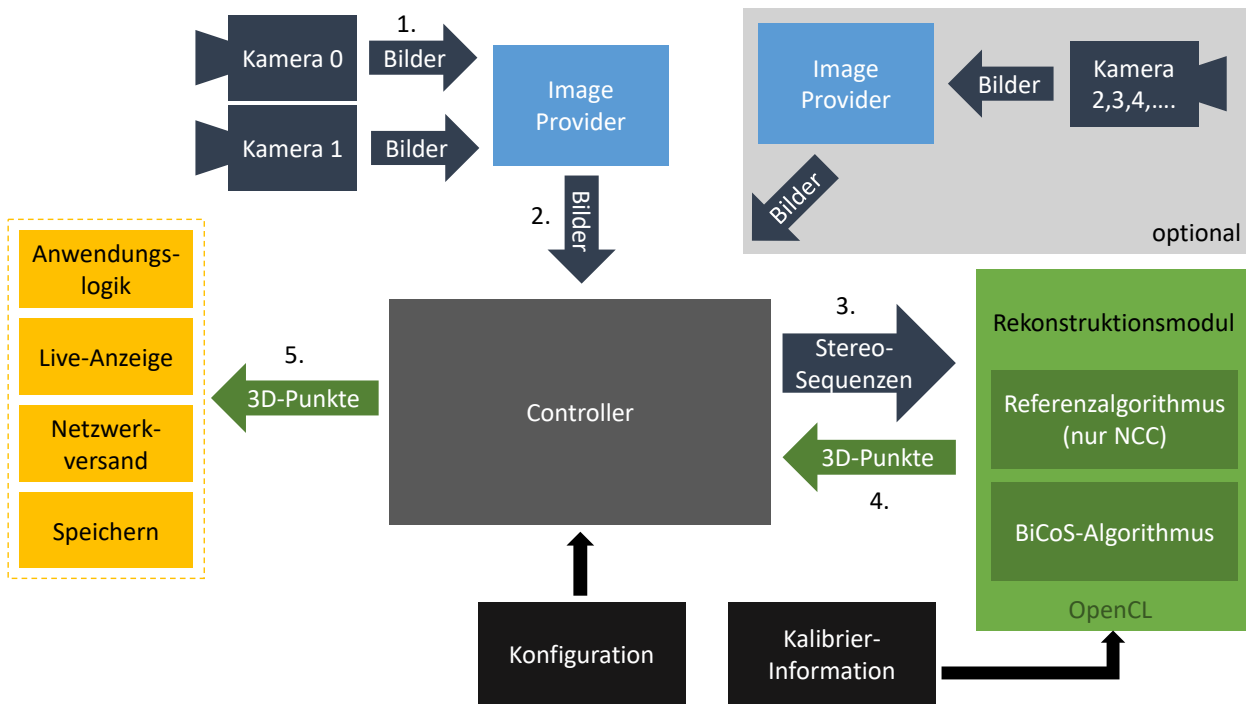


Abb. 3.3: Softwarearchitektur des 3D-Rekonstruktions-Framework. (Der BiCoS-Algorithmus wird in Abschnitt 3.5 eingeführt.).

Die rekonstruierte 3D-Punktwolke wird vom *Controller*-Modul dann an die Anwendungslogik (gelb) weitergegeben. Außerdem umfasst das Framework Module zum Speichern der 2D- und 3D-Daten*, zum Versenden der Daten übers Netzwerk und ein einfaches Live-Anzeigemodul für die resultierenden 3D-Punktwolken. Außerdem bietet es die Möglichkeit, weitere Kameras (z.B. Farb- oder Wärmebildkameras) anzubinden und deren Bilder auf die 3D-Punktwolken zu projizieren.

*: Wegen des hohen Datendurchsatzes muss das Speichermedium, z.B. ein SSD-Speicher, je nach Anwendung unter Umständen eine hohe Datenrate unterstützen.

Die Rechenzeiten der Bearbeitungsschritte des Algorithmus sind in Abbildung 3.4 exemplarisch für einen Sensor mit 2 MPx, gerechnet auf einer NVIDIA GeForce GTX 2080 GPU, dargestellt.**

** : Der verwendete Sensor und dessen Parametrisierung werden in Kapitel 5 beschrieben.

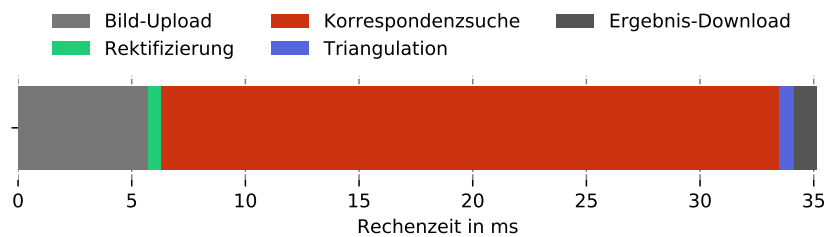


Abb. 3.4: Exemplarische Rechenzeiten der Bearbeitungsschritte des Referenzalgorithmus inklusive Upload der Bilder in den GPU-Speicher und Download des Ergebnisses.

Die schnelle GPU-Implementierung ermöglicht bereits eine 3D-Rekonstruktion innerhalb von 35 ms. Mit Abstand am rechenaufwändigsten ist die NCC-basierte Korrespondenzsuche. Danach folgt die Zeit für den Upload der Bilder in den GPU-Speicher. Letztere ist bei gegebener Kameraauflösung, Bildsequenzlänge und Rechenhardware nicht änderbar. Aus diesem Grund muss eine Verbesserung des 3D-Rekonstruktions-Algorithmus bei der Korrespondenzsuche ansetzen.

3.5 Optimierte Korrespondenzsuche

3.5.1 Lösungsansatz

Die Korrespondenzsuche ist bei weitem der rechenaufwändigste Teil des 3D-Rekonstruktions-Algorithmus (siehe Abbildung 3.4). Dafür gibt es zwei Gründe: 1. ist die Anzahl der notwendigen Pixelvergleiche hoch, 2. sind die Vergleiche wegen der hohen Anzahl zu berechnender Multiplikationen* sehr rechenaufwändig. Daraus folgen direkt zwei Lösungsansätze, um die Rechenzeit zu verkürzen:

1. Verringerung der Anzahl der Pixelvergleiche,
2. Verringerung des Rechenaufwandes je Pixelvergleich.

Die Anzahl der Pixelvergleiche lässt sich reduzieren, wenn ein kleinerer Tiefenbereich im Messvolumen abgesucht wird, da ein Tiefenbereich sich auf ein Disparitätsintervall abbildet. Dies könnte z.B. genutzt werden, indem ein Rekonstruktions-Algorithmus die Annahme macht, dass die Korrespondenzen für zwei benachbarte Pixel im linken Bildstapel auch im rechten Bildstapel nah beieinander liegen. Das ist zwar für stetige Objektflächen der Fall, nicht aber an den Kanten von Objekten, die mit zusätzlicher Logik zu erkennen wären. Außerdem hat diese Methode den Nachteil, dass die Pixelkorrespondenzen nacheinander berechnet werden müssten, um die Nachbarschafts-Information aus der jeweils vorhergehenden Suche nutzen zu können. Eine solche sequenzielle Abarbeitung ist nicht gut für die Implementierung auf hochgradig parallelisierten Recheneinheiten wie Grafikkarten oder FPGAs geeignet.

Der Autor dieser Arbeit hat sich deshalb für den zweiten Lösungsansatz entschieden: Es wird ein zweigeteilter Pixel-Korrespondenz-Suchalgorithmus vorgeschlagen (Abbildung 3.5): In Teil 1 werden Korrespondenzen mit grober Genauigkeit mittels eines geschwindigkeitsoptimierten Ähnlichkeitsmaßes gesucht. In Teil 2. werden die grob gefundenen Korrespondenzen mit Hilfe der NCC Subpixel-genau verfeinert. Es wird gezeigt werden (siehe Abschnitte 3.7 und 5.5), dass es möglich ist, mittels dieser Aufteilung eine erhebliche Geschwindigkeitsverbesserung zu erreichen, ohne dabei die Genauigkeit der Ergebnisse zu verschlechtern.

3.5.2 Konzept: schnelleres Ähnlichkeitsmaß

Da die NCC wegen vieler Multiplikationen und Additionen rechenaufwändig ist, ist das Ziel ein neues Ähnlichkeitsmaß zu finden, welches deutlich weniger Rechenoperationen benötigt. Ein bekannter Ansatz zur Reduktion der Rechenoperationen ist, sogenannte **Binäreigenschaften** (engl.: *binary features*, BF) zu verwenden. Jede Binäreigenschaft wird als ein Bit gespeichert. Ein Vektor aus solchen Bits kann aus den Helligkeitssequenzen berechnet werden. D.h. anstatt mit Ganzzahl- oder Fließkommavektoren zu rechnen, werden Vektoren aus Binäreigenschaften, also Bitvektoren verwendet.

Zwei Bitvektoren mit einer Länge von 64 Bit können in einer einzigen 64-Bit-Rechenoperation verglichen werden (XOR-Operation) und mit einer weiteren (POPCOUNT) kann dann die Anzahl der ungleichen bzw. gleichen Bits zwischen den Vektoren ermittelt werden.

Binäreigenschaften werden erfolgreich in der Photogrammetrie angewendet, um Korrespondenzen zwischen unkalibrierten Kameras herzustellen [72–74],

*: Siehe detaillierte Analyse in Abschnitt 3.3

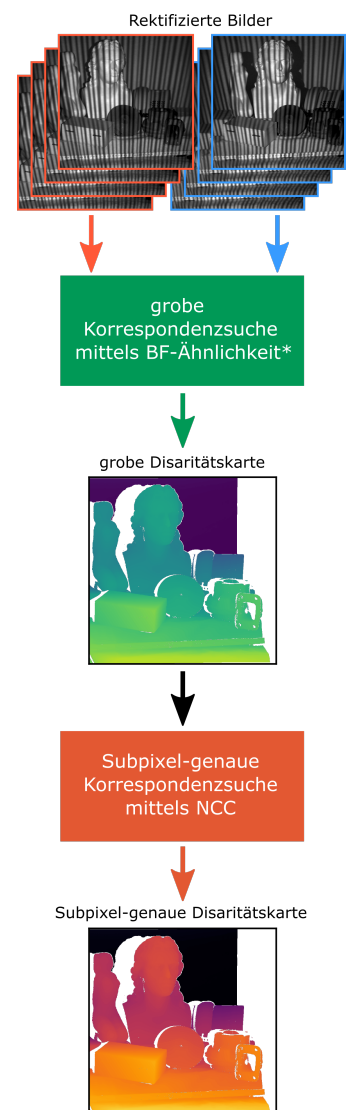


Abb. 3.5: Aufteilung der 3D-Rekonstruktion in grobe und Subpixel-genaue Korrespondenzsuche. *) Die BF-Ähnlichkeit wird in Abschnitt 3.5.2 eingeführt.

wobei hier die Binäreigenschaften von Bildregionen berechnet werden (ähnlich wie beim Passiven-Stereo-Verfahren, vgl. Tabelle 3.2). Eine vergleichende Analyse dieser Verfahren findet sich in [75]. Außerdem werden Binäreigenschaften auch unabhängig von der Korrespondenzsuche zur Textur-Klassifikation eingesetzt [76].

Ein neues Ähnlichkeitsmaß für den Vergleich zweier Helligkeitssequenzen ergibt sich durch die Berechnung der beiden Bitvektoren aus den Helligkeitssequenzen und dem anschließenden Vergleich der Bitvektoren miteinander. Dieses neue Ähnlichkeitsmaß [53, 54] wird im Folgenden **BF-Ähnlichkeit** genannt.

Definition 3.5.1 Die *BF-Ähnlichkeit* ist definiert als die Anzahl der übereinstimmenden Binäreigenschaften der Helligkeitssequenzen zweier Pixel.

Schnell wird die BF-Ähnlichkeits-Berechnung, weil der Bitvektor aus Binäreigenschaften für jedes Pixel in den beiden Bildstapeln unmittelbar nach der Rektifizierung vorberechnet werden kann. D.h. jeder Bitvektor muss nur einmal berechnet werden, während bei der folgenden Korrespondenzsuche jeder Bitvektor mit vielen anderen verglichen wird.

In Abschnitt 3.5.3 wird beschrieben, wie die Binäreigenschaften vorberechnet werden. Das genaue Vorgehen bei der Korrespondenzsuche mittels Binäreigenschaften wird in Abschnitt 3.5.4 erläutert.

3.5.3 Vorbereitung der Binäreigenschaften

Für die Berechnung des Bitvektors für ein Pixel werden insgesamt drei verschiedene Methoden angewendet, um aus der Helligkeitssequenz die Binäreigenschaften zu erzeugen:

- A mittels Vergleichs zweier Helligkeitswerte miteinander (Abbildung 3.6)
- B mittels Vergleichs eines Helligkeitswertes mit der mittleren Helligkeit der Sequenz (Abbildung 3.7)
- C mittels Vergleichs zweier Summen aus jeweils zwei Helligkeitswerten miteinander (Abbildung 3.8)

Jeder Vergleich liefert eine Binäreigenschaft. Beispielsweise ergibt (unter Anwendung von Methode A) ein Vergleich des Helligkeitswertes x_1 mit x_2 einen Bit-Wert von 1, wenn $x_1 > x_2$ und sonst 0. Dabei ist es egal, ob als Vergleichsoperator $>$, $<$, \geq oder \leq verwendet wird, solange dies konsistent erfolgt.

Für Methode A wird jeder Helligkeitswert in der Helligkeitssequenz eines Pixels mit jedem anderen verglichen. Für eine Sequenzlänge von 10 Helligkeitswerten ergeben sich damit 45 Vergleiche bzw. Binäreigenschaften. Methode B liefert weitere 10 Binäreigenschaften, wenn jeder einzelne Helligkeitswert mit dem Mittelwert aller verglichen wird.

Methode C erfordert einen zusätzlichen Rechenschritt:

1. Innerhalb der Helligkeitssequenz eines Pixels werden Paare aus Helligkeitswerten gebildet.
2. Die beiden Helligkeitswerte eines Paares werden addiert.
3. Zwei Summen werden miteinander verglichen, und zwar nur solche Summen, die keine gemeinsamen Summanden haben. Das Ergebnis des Vergleichs ist die Binäreigenschaft.

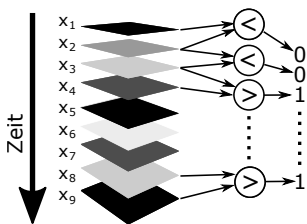


Abb. 3.6: Erzeugung von Binäreigenschaften mittels Vergleichs von Helligkeitswerten (Methode A) [54]

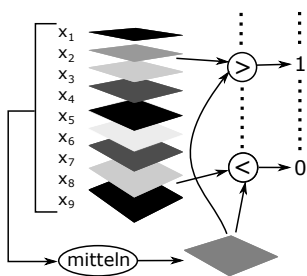


Abb. 3.7: Erzeugung von Binäreigenschaften mittels Vergleichs von Helligkeitswerten mit dem Mittelwert (Methode B) [54]

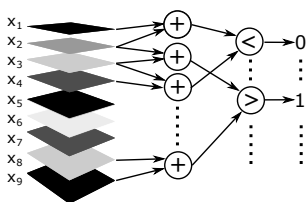


Abb. 3.8: Erzeugung von Binäreigenschaften aus Summen von Helligkeitswerten (Methode C) [54]

Die drei verschiedenen Erzeugungsmethoden erlauben es auch für kurze Sequenzlängen ausreichend viele Binäreigenschaften für jedes Pixel zu erzeugen. In Abbildung 3.9 ist die Anzahl der Binäreigenschaften, die mit jeder der drei Methoden erzeugt werden kann in Abhängigkeit der Bildsequenz-Länge dargestellt. Um die Rechenzeit zu begrenzen werden maximal 64 Binäreigenschaften produziert. Es werden zunächst alle möglichen Binäreigenschaften berechnet, die mit Methode A erzeugt werden können, sollten dies weniger als 64 sein, folgt Methode B und dann Methode C falls immer noch weniger als 64 Binäreigenschaften erzeugt wurden. Durch die Begrenzung auf 64 können alle Binäreigenschaften eines Pixels in einer 64-Bit breiten Variable gespeichert werden.

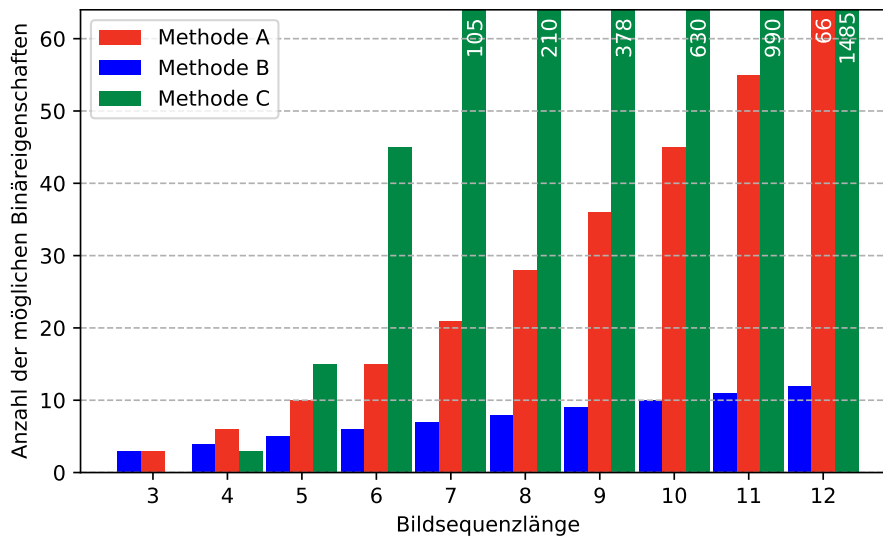


Abb. 3.9: Anzahl der erzeugbaren Binäreigenschaften nach Erzeugungsmethode [54].

3.5.4 Konsistenzcheck und Median-Filter

Um für ein Pixel im linken, rektifizierten Bildstapel ein korrespondierendes Pixel zu finden, wird die BF-Ähnlichkeit für jedes der Kandidatenpixel im rechten Bildstapel bestimmt. Das ähnlichste Pixel wird ausgewählt. Eine so gefundene Pixelkorrespondenz wird auf Konsistenz geprüft, indem die Suche nochmals umgekehrt durchgeführt wird. D.h. für das im ersten Suchlauf gefundene Pixel im rechten Bildstapel, werden die BF-Ähnlichkeiten der Kandidaten im linken Bildstapel berechnet. Nur wenn das bei dieser Rückwärtssuche gefundene Pixel des linken Bildstapels das ursprüngliche Pixel ist oder unmittelbar an dieses angrenzt, wird die Korrespondenz akzeptiert.

Das Ergebnis dieser Korrespondenzsuche ist eine grobe Disparitätskarte. Grob ist sie in dem Sinne, dass die Disparitäten ganzzahlig, also nicht Subpixel-genau sind. Um Ausreißer zu entfernen und Löcher zu füllen, welche durch die Rückwärtssuche entstanden sind, wird diese grobe Disparitätskarte mittels eines räumlichen Medianfilters der Größe 3×3 gefiltert.*

3.5.5 Subpixel-genaue Korrespondenzverfeinerung

Die bis hierher beschriebene grobe Korrespondenzsuche ergibt eine Disparitätskarte, die als Startwert für die nachfolgende Subpixel-genaue Korrespondenzsuche dient. Die verfeinerte Korrespondenzsuche für ein einzelnes Pixel der linken Kamera erfolgt nach folgender Vorschrift:

*: Dies bedeutet **nicht**, dass das abschließende 3D-Ergebnis gefiltert ist, da die in diesem Schritt erzeugten grobe Disparitäten lediglich die Startwerte für die folgende Korrespondenz-Feinsuche sind.

*: Vorausgesetzt die Startlösung war korrekt in dem Sinne, dass sie innerhalb des m -Intervalls um die wahre Korrespondenz lag, sollte das Pixel P_i auch genau dem entsprechen, das sich ergäbe, wenn mit dem Referenzalgorithmus, d.h. ausschließlich mit NCC gesucht worden wäre.

- ▶ Suche nochmals mittels NCC pixelgenau in einem Intervall m Pixel links und rechts des in der Grobsuche gefundenen Pixels. Das Ergebnis ist ein Pixel P_i .*
- ▶ Suche, ausgehend vom Pixel P_i , durch lineare Interpolation dessen Helligkeitswerte mit denen der Nachbarpixel P_{i-1} bzw. P_{i+1} die beste Subpixel genaue Korrespondenz.

Diese Subpixel-Verfeinerung kann durchgeführt werden, indem innerhalb des Intervalls zwischen P_{i-1} und P_{i+1} eine feste Anzahl (z.B. 100) äquidistanter Subpixel berechnet wird (mittels linearer Interpolation der Helligkeitswerte). Jedes dieser interpolierten Subpixel wird dann mittels der NCC mit dem Pixel der anderen Kamera verglichen. Das Subpixel mit dem höchsten Korrelationswert wird ausgewählt und ist die verfeinerte Korrespondenz.

Bei dieser „Brute-Force“-Methode werden also sehr viele Subpixel berechnet und verglichen. Um die Berechnung zu beschleunigen, wird stattdessen folgende effizientere Methode eingesetzt:

Es wird folgendes Prinzip der binären Suche [77] verwendet: Das linke und rechte Nachbarpixel (P_{i-1} und P_{i+1}) werden als *linke Grenze* und *rechte Grenze* der Suche festgelegt. Das bereits gefundene, bis jetzt optimale Pixel P_i wird als *Mitte* festgelegt. Danach wird mittig zwischen *linker Grenze* und *Mitte* sowie zwischen *Mitte* und *rechter Grenze* jeweils ein neues Subpixel durch Interpolation der Helligkeitswerte erzeugt, also $P_{i-\frac{1}{2}}$ und $P_{i+\frac{1}{2}}$. Das Subpixel mit dem besseren Korrelationswert wird die neue *Mitte*. Die alte *Mitte* wird zur neuen *linken Grenze* falls $P_{i+\frac{1}{2}}$ besser war oder zur *rechten Grenze* falls $P_{i-\frac{1}{2}}$ besser war. Das neue Intervall ist also nur noch halb so groß.

Dieses Prinzip wird mehrfach wiederholt, wobei sich in jedem Wiederholungsschritt die Breite des Intervalls halbiert. Abbruchbedingung ist eine feste, konfigurierbare Anzahl von Schritten. Bereits nach 7 Schritten ist das Intervall kleiner, als beim zuvor beschriebene Brute-Force-Verfahren mit einer Unterteilung von 100 Schritten ($2^7 = 128$).

3.6 Eigenschaften der BF-Ähnlichkeit

3.6.1 Invarianz gegenüber Umgebungslicht und richtungsabhängiger Reflektivität

Die NCC ist invariant unter den folgenden Beleuchtungsveränderungen zwischen den Kameras: Umgebungslicht, Objektreflektivität sowie Kamerasensitivität (siehe Gleichung 3.2). In diesem Abschnitt wird gezeigt, dass dies auch für die BF-Ähnlichkeit (siehe Definition 3.5.1) gilt.

Jede Binäreigenschaft wird mittels Vergleich zweier Helligkeitswerte (bzw. Helligkeitsmittelwert oder Summe von Helligkeitswerten) desselben Pixels berechnet. Jegliches zusätzliches, zeitlich konstantes Umgebungslicht wird auf beide in den Vergleich eingehende Werte addiert und ändert somit das Ergebnis des Vergleichs nicht. Ebenso ändert es sich nicht, wenn beide Vergleichswerte um einen positiven Faktor skaliert werden. Da sich sowohl die Reflektivität des Objektes, als auch die Kamerasensitivität als Faktor beschreiben lassen, gibt es

auch eine Invarianz bezüglich dieser Eigenschaften. Für zwei Helligkeitswerte x_1 und x_2 , Reflektivität R und Umgebungslicht U gilt:

$$x_1 < x_2 \Leftrightarrow (R \cdot x_1 + U) < (R \cdot x_2 + U) \quad \forall R > 0. \quad (3.3)$$

3.6.2 Statistischer Zusammenhang zwischen NCC und BF-Ähnlichkeit

Kann die BF-Ähnlichkeit die NCC approximieren? D.h. ist bei einem hohen NCC-Wert auch ein hoher BF-Ähnlichkeits-Wert zu erwarten und bei einem niedrigen NCC-Wert ein niedriger BF-Ähnlichkeits-Wert? Um diesen Zusammenhang statistisch zu untersuchen wurde folgendes Experiment durchgeführt:

Für eine Bildsequenz eines 3D-Sensors* wurden die Bilder rektifiziert und dann alle möglichen Pixel miteinander verglichen. Für jeden Vergleich wurde sowohl die NCC und die BF-Ähnlichkeit berechnet. D.h. es entstand jeweils ein Paar Ähnlichkeitswerte. Diese Paare wurden in ein zweidimensionales Histogramm eingetragen (Abbildung 3.10). D.h. für einen bestimmten BF-Ähnlichkeitswert und ein bestimmtes NCC-Intervall gibt es einen Bin, dessen Wert die Häufigkeit des Auftretens dieser Kombination von BF-Ähnlichkeit und NCC widerspiegelt. Das Histogramm wurde auf die Anzahl der NCC-Werte in jeder Spalte normiert.

*: Für eine Beschreibung der Messszene, des Projektors und der Kameras siehe Abschnitt 5.1.

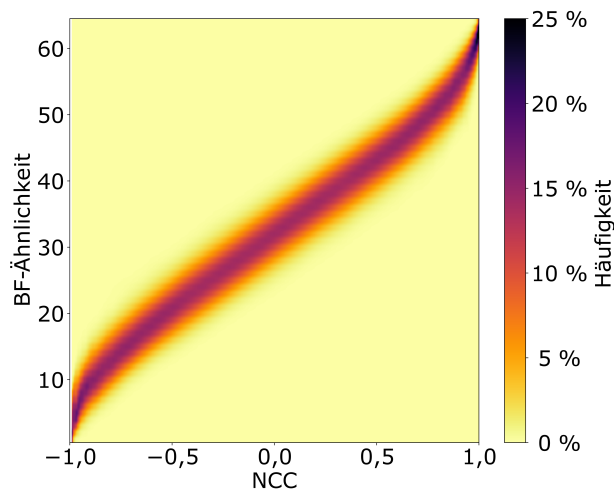


Abb. 3.10: Korrelation der beiden Ähnlichkeitsmaße: 2D-Histogramm aller Paare aus NCC- und BF-Ähnlichkeitswerten, empirisch ausgewertet für alle Pixelvergleiche bei der Korrespondenzsuche einer Messung mit Bildsequenz-Länge $N = 10$. Zur besseren Erkennbarkeit ist das Histogramm spaltenweise normiert.

Die beiden Ähnlichkeitsmaße zeigen einen weitgehend linearen Zusammenhang. Die BF-Ähnlichkeit kann also als grobe Approximation der NCC angesehen werden. Für NCC-Werte nahe 1 wird der Verlauf steiler. Dies ist vorteilhaft, da es bedeutet, dass der wichtige Bereich hoher Ähnlichkeiten durch die BF-Ähnlichkeiten genauer aufgelöst wird als z.B. der Bereich um 0,5.

3.7 Implementierung des schnellen Korrespondenzsuch-Algorithmus

Der schnelle, BF-Ähnlichkeits-basierte Korrespondenzsuch-Algorithmus wird der im Folgenden auch **BiCoS-Algorithmus** (BiCoS = *Binary Correspondence*)

Search) genannt. Er wurde mit OpenCL implementiert und in das C++-Framework integriert, genau wie der Referenzalgorithmus. Somit kann durch einfache Parameteränderung in einer Konfigurations-Datei der Algorithmus ausgewählt werden.

Um die Rechenzeit der Arbeitsschritte des 3D-Rekonstruktions-Algorithmus zu ermitteln, wurde eine Szene mit dem in Kapitel 5 beschriebene 3D-Sensor aufgenommen und im Live-Betrieb auf einer NVIDIA GeForce GTX 2080 GPU rekonstruiert. Die Rekonstruktion wurde 100 Mal durchgeführt und die Rechenzeiten jeweils mit dem NVIDIA NSIGHT Profiler protokolliert. Dabei lagen die Variationen der Rechenzeiten bei $\pm 2,1\%$. Hier werden aus Gründen der Anschaulichkeit nur die Durchschnittswerte angegeben. Abbildung 3.11 zeigt die Rechenzeiten für beide Algorithmen.

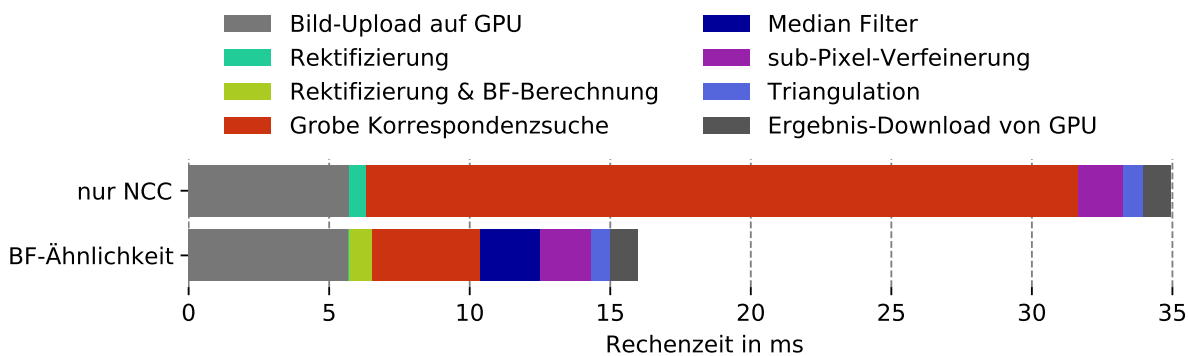


Abb. 3.11: Vergleich der Rechenzeiten (inklusive Zeiten für den Bild-Upload in den GPU-Speicher und den Download der Ergebnisse) für den rein NCC-basierten Rekonstruktions-Algorithmus (oben) und den geschwindigkeitsoptimierten Algorithmus (unten) jeweils für Bildsequenz-Länge 10 und das in Kapitel 5 beschriebene Messsystem.

Beim rein NCC-basierten Referenzalgorithmus ist der rechenaufwändigste Schritt die grobe Korrespondenzsuche mittels NCC als Pixelähnlichkeitsmaß (25,4 ms). Beim schnellen Algorithmus dauert die grobe Korrespondenzsuche mittels Binäreigenschaften erheblich kürzer (3,9 ms). Dazu muss allerdings die Zeit für den Median Filter von 2,1 ms addiert werden, den es beim Referenzalgorithmus nicht gibt. Die Subpixel-Verfeinerung dauert beim Referenzalgorithmus geringfügig kürzer (1,6 ms gegenüber 1,8 ms) weil dort keine pixelgenaue Suche im Umkreis des Grobsuch-Ergebnisses erfolgen muss. Die Rektifizierung der Bilder dauert beim Referenzalgorithmus lediglich 0,6 ms während der schnelle Algorithmus 0,8 ms braucht, weil während der Rektifizierung auch noch die Binäreigenschaften berechnet werden.

In Summe lässt sich mittels des im Rahmen dieser Arbeit entwickelten schnellen Rekonstruktions-Algorithmus die Rechenzeit gegenüber dem Referenzalgorithmus bei ansonsten ähnlicher Implementierung mehr als halbieren. Die 3D-Rekonstruktion trägt damit noch knapp 17 ms zur Messlatenz bei. Die Qualität der Messergebnisse wird in Kapitel 5 genauer untersucht und verglichen.

Beschleunigung der 3D-Rekonstruktion

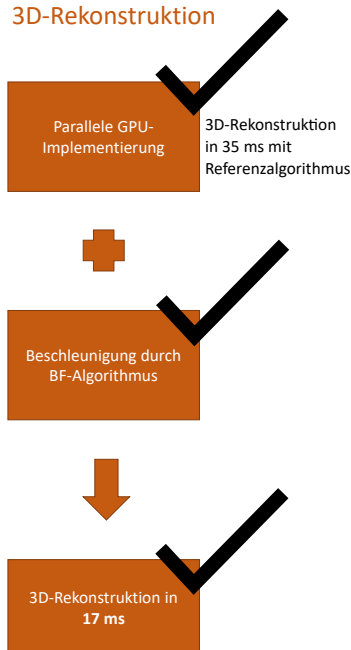


Abb. 3.12: Erreichte Verbesserungen

Bei einer Kamera-Framerate von 100 Hz und einer Bildsequenzlänge von $N = 10$ liegt die Aufnahmelatenz bei $10/100 \text{ Hz} = 100 \text{ ms}$. Soll diese reduziert werden, können schnellere und lichtempfindlichere Kameras verwendet werden. Falls das z.B. aus Kostengründen nicht möglich ist, so muss die Bildsequenzlänge verkürzt werden. Eine naive Reduktion der Bildsequenzlänge N führt jedoch zu Messartefakten [78].

Abbildung 4.1 zeigt ein 3D-Modell mit solchen Messartefakten. Die Messartefakte entstehen durch in Clustern auftretende Fehlzuordnungen bei der Korrespondenzsuche, was zu falschen Disparitäten und letztendlich auch falschen 3D-Koordinaten führt.

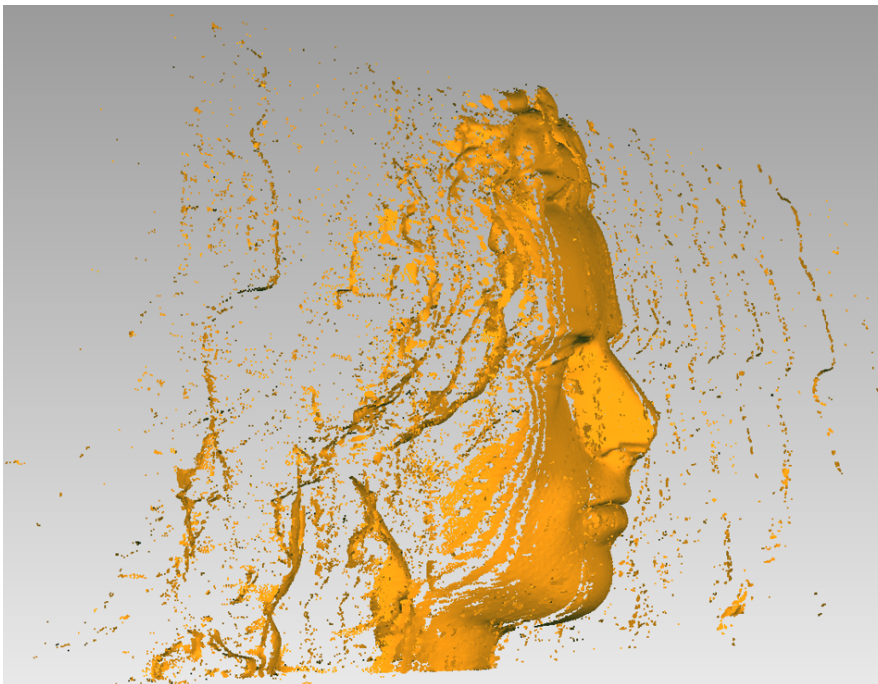


Abb. 4.1: 3D-Modell mit unerwünschten Messartefakten, das mit nicht optimiertem Muster und zu kurzer Bildsequenzlänge von $N = 6$ aufgenommen wurde. Die Blickrichtung des Sensors war von rechts nach links [56].

Diese Fehlzuordnungen bei der Korrespondenzsuche entstehen, weil die projizierten Muster bei der kurzen Bildsequenzlänge unter dem Einfluss des Bildrauschens nicht mehr überall eindeutig sind. D.h. für einige Musterbereiche auf dem Messobjekt gibt es andere Musterbereiche, die, unter Anwendung des bei der Korrespondenzsuche verwendeten Ähnlichkeitsmaßes, sehr ähnlich aussehen und deshalb verwechselt werden.

In diesem Kapitel werden zwei Maßnahmen beschrieben, mit denen die Zuverlässigkeit der Korrespondenzsuche verbessert werden kann:

1. eine Änderung des Projektionsmodus: anstatt kontinuierlich wird gepulst projiziert,
2. ein Optimierungsverfahren für die auf dem GOBO aufbrachten Projektionsmuster.

4.1 Projektionsmuster, Codewörter, Freiheitsgrade

Um eine Punktkorrespondenz zwischen den beiden Kameras eindeutig zu finden, ist es erforderlich, dass die projizierten, aperiodischen Streifenmuster jeden Punkt auf der Epipolarlinie möglichst eindeutig kodieren. Die Helligkeitssequenzen, welche die Kameras für jedes Pixel aufnehmen, können in diesem Sinne als „Codewort“ betrachtet werden. Diese Codewörter, sollen über die Länge des sichtbaren Musters eindeutig sein. Wäre diese Eindeutigkeit nicht gegeben, so gäbe es zu einem Pixel in der linken Kamera mehrere gleich ähnliche Kandidaten in der rechten, wobei die falschen dann nicht vom korrekten zu unterscheiden wären. Dieses Problem besteht in abgeschwächter Form, wenn das Muster zwar theoretisch eindeutig, aber der Unterschied zwischen manchen Codewörtern so gering ist, dass sie angesichts des Bildrauschens nur mit einer geringen statistischen Wahrscheinlichkeit korrekt unterscheidbar sind.

Bei einer Sequenzlänge von $N = 10$ Bildpaaren ist jedes Codewort ein Vektor aus 10 Helligkeitswerten. Bei Anwendung des Ähnlichkeitsmaßes NCC (siehe Abschnitt 3.3) erfolgt beim Vergleich zweier Helligkeitssequenzen eine Normierung mittels Subtraktion des Mittelwertes und Division durch die Standardabweichung des Codewortes. Nach dieser Normierung sehen zwei Codewörter, die sich ausschließlich in ihrer absoluten Helligkeit und einem Offset unterscheiden, gleich aus. Die Normierung reduziert die Codewörter also um zwei Freiheitsgrade. Bei einer Sequenzlänge von $N = 10$ haben die Codewörter beim Vergleich miteinander also 8 Freiheitsgrade. Bei einer Sequenzlänge von $N = 6$ dagegen nur noch 4.

In Abschnitt 3.6.1 wurde gezeigt, dass für die BF-Ähnlichkeit die gleichen Invarianzen bezüglich der absoluten Helligkeit und einem Offset der Helligkeitswerte besteht, wie bei der NCC. Daraus folgt, dass die Anzahl der Freiheitsgrade beim Vergleich zweier Codewörter mittels BF-Ähnlichkeit nicht größer sein kann, als bei der NCC.

Für Muster, mit zufälligen, gleichverteilten Streifenbreiten zeigte sich, dass ab einer gewissen Länge N der Codewörter, d.h. der aufgenommenen Bildsequenzen, eine hinreichende Eindeutigkeit erreicht werden kann, um 3D-Rekonstruktionen mit relativ wenigen Ausreißern durchführen zu können [43]. Die dafür notwendige Bildsequenzlänge N ist abhängig vom Signal-zu-Rausch-Abstand, also insbesondere von den verwendeten Kameras, der Kameraintegrationszeit, der Lichtstärke des Projektors und dem Umgebungslicht. Für typische Industriekameras, Objekte mit mittlerer bis hoher Reflektivität (d.h. nicht schwarz), schwaches Umgebungslicht und ausreichend Projektionslicht, um den Dynamikumfang der Kameras zu ca. 70 % auszunutzen, hat sich gezeigt, dass eine Sequenzlänge von $N = 10$ oder mehr ausreicht [43, 79].

4.2 Autokorrelationsanalyse

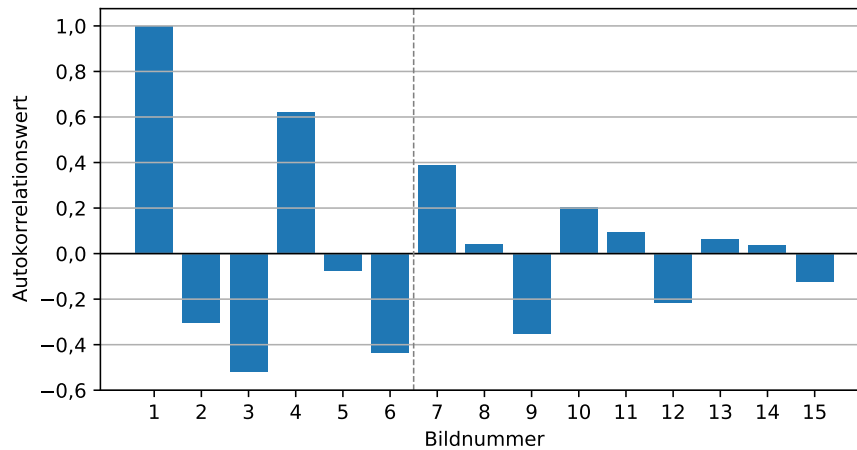
Da mit dem GOBO-Projektor nicht für jede Bildaufnahme ein komplett neues, unabhängiges Muster projiziert wird, sondern das gleiche Muster weitergedreht wird, ist eine statistische Abhängigkeit der Helligkeitswerte zu erwarten. Um die tatsächliche statistische Abhängigkeit der Helligkeitswerte innerhalb der Helligkeitssequenzen/Codewörter abzuschätzen, wurde eine Autokorrelationsanalyse durchgeführt.

*: eine Beschreibung der Messszene folgt in Abschnitt 5.1

Dazu wurden 512 Kamerabilder einer Messszene* bei laufendem Projektor

aufgenommen. Es wurde ein zufälliges, Muster mit gleichverteilten Streifenbreiten verwendet^{**}. Die aufgenommenen Bilder haben, wegen des rotierenden GOBO-Musters, eine hohe zeitliche Varianz.

Für jedes Pixel lag also eine Folge von 512 Helligkeitswerten vor. Danach wurde für jedes Pixel diese Folge schrittweise (zeitlich) verschoben und in jedem Schritt mit der unverschobenen Folge korreliert (mittels normierter Kreuzkorrelation). Die Ergebnisse für die ersten 15 Schritte sind in Abbildung 4.2 dargestellt.



^{**}: Es handelt sich um das in Abschnitt 5.2 beschriebene GOBO R.

Abb. 4.2: Zeitliche Autokorrelation der von den Kameras aufgenommenen Helligkeitswerte des **nicht optimierten** GOBO-Musters bei **kontinuierlicher** Projektion: gezeigt sind die Autokorrelationswerte des n -ten Helligkeitswertes mit den zeitlich folgenden Werten. Die betragsmäßig hohen Werte, insbesondere die der Korrelation mit dem 3., 4. und 6. Wert, bedeuten einen verringerten Informationsgehalt der Codewörter. Erst die späteren Helligkeitswerte weisen eine niedrigere Korrelation mit dem ersten auf. Die gestrichelte Linie symbolisiert die Grenze einer Helligkeitssequenz der Länge $N = 6$. Die Werte rechts dieser Linie spielen bei Rekonstruktion mit $N = 6$ keine Rolle mehr.

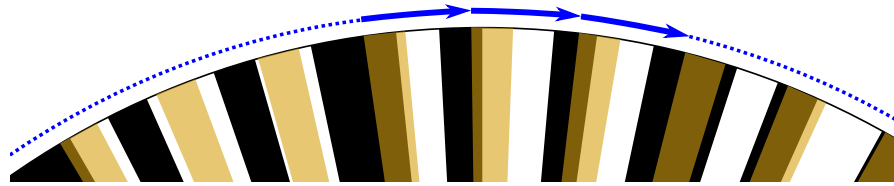
Die Autokorrelationswerte zeigen eine erhebliche Abhängigkeit der Helligkeitswerte voneinander, insbesondere die der Korrelation mit dem 3., 4. und 6. Wert sind betragsmäßig hoch. Vereinfacht ausgedrückt könnte dies bedeuten, dass auf einen hellen Pixelwert zwei Bilder weiter mit deutlich erhöhter Wahrscheinlichkeit ein dunkler und noch ein Bild weiter wieder ein heller Pixelwert folgt. Erst für den 5. Helligkeitswert ist eine Vorhersage nicht mehr einfach möglich. Der 6. korreliert aber wieder negativ, d.h. er ist mit erhöhter Wahrscheinlichkeit wieder dunkel.

4.3 Verbesserung der Mustereindeutigkeit durch schnellere GOBO-Rotation

Die so voneinander statistisch abhängigen Helligkeitswerte erschweren die Unterscheidbarkeit der Codewörter insofern, dass bestimmte Helligkeitskombinationen deutlich häufiger auftreten als andere, und deshalb für diese mehr Kandidaten für eine Verwechslung vorhanden sind. Die Ergebnisse in Abbildung 4.2 zeigen aber auch, dass das Problem mit zunehmender zeitlicher Entfernung vom ursprünglichen Helligkeitswert abnimmt.

Dies führt direkt zu einem Lösungsansatz: könnte das GOBO-Rad zwischen zwei Bildaufnahmen deutlich weitergedreht werden, so könnte der stark korrelierende Teil des Musters übersprungen werden. Bei kontinuierlicher Projektion würde eine schnellere GOBO-Rotation aber zu mehr Bewegungsunschärfe und damit zu geringerem Muster-Kontrast führen. Dieser naive Ansatz funktioniert deshalb leider nicht. Stattdessen wurde folgender Ansatz untersucht, der in Abbildung 4.3 veranschaulicht wird:

Abb. 4.3: Veranschaulichung der gepulsten Projektion: Hintergrund: GOBO-Rad mit Muster (schwarz/weiß), gelb: Beleuchtungsintervalle, blaue Pfeile: Rotation bis zur nächsten Beleuchtung.



Das Projektionslicht wird gepulst, d.h. innerhalb des Zeitintervalls zwischen zwei Aufnahmen ist das Projektionslicht lediglich einen Bruchteil der Zeit an. Das GOBO-Rad wird in dieser kürzeren Zeit genauso weit gedreht, wie ursprünglich im gesamten Zeitraum zwischen zwei Aufnahmen. Damit wird die gleiche Bewegungsunschärfe realisiert.

Bei einem Beleuchtungszeitintervall von 40 % bedeutet dies z.B. eine um Faktor 2,5 schnellere Rotation. In den übrigen, unbeleuchteten 60 % der Zeit dreht sich das Rad entsprechend weiter.

Zu beachten ist dabei, dass bei gleicher Strahlungsleistung in dem verkürzten Beleuchtungsintervall dann auch nur 40 % des Lichts abgegeben wird. In der Praxis wird aber eine Eigenschaft der Beleuchtungs-LEDs ausgenutzt: diese können gepulst mit deutlich höheren Strömen betrieben werden und haben deshalb in dem kurzen Zeitintervall eine deutlich höhere Strahlungsleistung. Für den 3D-Sensor, der in Kapitel 5 vorgestellt wird, ist die abgegebene Strahlungsenergie je Bildaufnahme in beiden Beleuchtungsmodi (gepulst und kontinuierlich) deshalb ungefähr gleich. Abbildung 4.4 zeigt die mit 40 % Beleuchtungszeit erzielten Autokorrelationswerte.

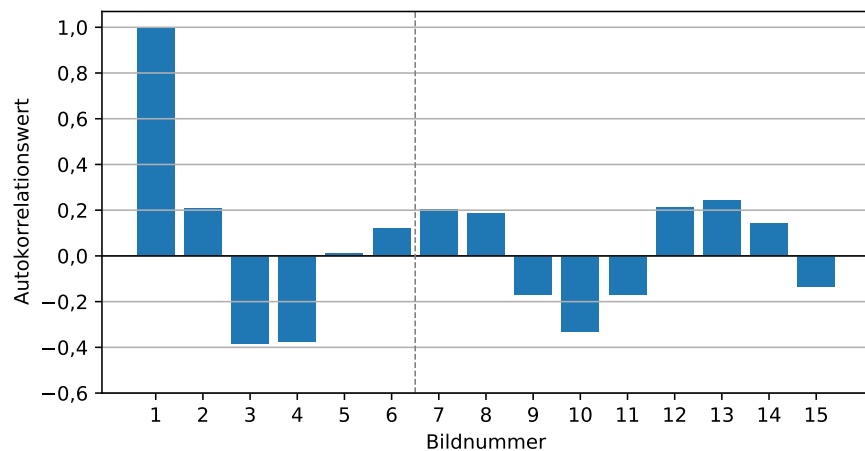


Abb. 4.4: Zeitliche Autokorrelation der von den Kameras aufgenommenen Helligkeitswerte des **nicht optimierten** GOBO-Musters bei **gepulster** Projektion (40 % Beleuchtungszeit): gezeigt sind die Autokorrelationswerte des n -ten Helligkeitswertes mit den zeitlich folgenden Werten. Die gestrichelte Linie symbolisiert die Grenze einer Helligkeitssequenz der Länge $N = 6$.

Diese Ergebnisse zeigen eine Verbesserung der Autokorrelationswerte. Dennoch sind der zweite und dritte Folgewert weiterhin betragsmäßig signifikant. Vereinfacht ausgedrückt wäre bei einem hellen Ausgangshelligkeitswert der zweite und dritte Wert mit erhöhter Wahrscheinlichkeit dunkel. In Abschnitt 5.5 wird gezeigt, dass diese hier geringfügig erscheinende, betragsmäßige Verringerung der Autokorrelationswerte in der Praxis eine große Verbesserung bedeutet.

Im Rahmen der vom Autor betreuten Bachelor-Arbeit von Malte Dubrall [80] wurde untersucht, ob es technisch möglich ist, mittels einer speziellen Motor-Ansteuerung eine ungleichmäßige Bewegung des GOBO-Rads zu realisieren. Und zwar sollte sich das GOBO-Rad während der Belichtung langsam drehen und während der unbeleuchteten Phase schnell. Das Ziel war eine weitere Verringerung der statistischen Abhängigkeit der Helligkeitswerte. Leider war

es nicht möglich, dieses Schema für die hohen Kamera-Frameraten umzusetzen, die für Messungen mit niedriger Latenz erforderlich sind.

4.4 Optimierung der Projektionsmuster

Dieses Unterkapitel beschreibt ein Verfahren zur Optimierung der aperiodischen Streifenmuster für verbesserte Codewort-Eindeutigkeit. Es basiert auf der Anpassung der relativen Streifenbreiten. Für typisches GOBO-Streifenmuster mit 30 oder mehr Streifen ist die Anzahl der möglichen Muster riesig (bei 10 möglichen Streifenbreiten pro Streifen gäbe es 10^{30} mögliche Muster). Deshalb ist es nicht möglich, alle Muster zu testen und das beste auszuwählen. Stattdessen werden ausgehend von einem zufällig erzeugten Startmuster in einem iterativen Verfahren die relativen Streifenbreiten schrittweise geändert. Diese schrittweisen Änderungen erfolgen systematisch mit dem Ziel, die Selbstähnlichkeit des Musters, bzw. die paarweisen Ähnlichkeiten der erzeugten Codewörter, zu minimieren.

Der für die Genauigkeit der Messung wesentliche Musterparameter der **durchschnittlichen** Streifenbreiten wurden in [78] untersucht und für diese Arbeit als gegeben angenommen.

4.4.1 Musterselbstähnlichkeit

Um die Codewort-Eindeutigkeit zu erhöhen, muss die Selbstähnlichkeit des Streifenmusters reduziert werden. Diese Musterselbstähnlichkeit kann gemessen werden, indem jedes Codewort mit jedem anderen verglichen wird, mit dem es auch während der Korrespondenzsuche verglichen werden könnte. Bei jedem Vergleich wird die Ähnlichkeit der beiden Codewörter mittels eines Ähnlichkeitsmaßes bestimmt. Haben alle Codewort-Paare eine niedrige Ähnlichkeit, ist die Wahrscheinlichkeit gering, dass unter Realbedingungen zwei Codewörter verwechselt werden.

In den Abschnitten 3.3 und 3.5.2 wurden zwei Pixel-Ähnlichkeitsmaße im Detail vorgestellt: die Normierte Kreuzkorrelation (NCC) und die Anzahl der übereinstimmenden Binäreigenschaften (BF-Ähnlichkeit). Da die BF-Ähnlichkeit keine stetige Funktion ist, sondern diskrete Werte (von 0 bis 64) annimmt, und deshalb auch nicht differenzierbar ist, ist sie für eine systematische Optimierung weniger gut geeignet. Aus diesem Grund erfolgt die Optimierung mit dem Ähnlichkeitsmaß NCC. In Abschnitt 3.6.2 wurde gezeigt, dass die NCC stark mit der BF-Ähnlichkeit korreliert, deshalb kann erwartet werden, dass eine hohe Eindeutigkeit, die mittels NCC gemessen wurde, auch zu einer guten Eindeutigkeit bezüglich der BF-Ähnlichkeit führt.

4.4.2 Optimierungskonzept

Die Optimierung eines aperiodischen Streifenmusters, die in den kommenden Abschnitten im Detail beschrieben wird, hat folgende Struktur:

1. Erzeuge ein zufälliges Startmuster definiert durch diskrete Streifenbreiten.
2. Simuliere die Abbildung des Projektors und die Aufnahme mit der Kamera in einer 1D-Simulation. Das Ergebnis sind die vom Muster erzeugten Helligkeitssequenzen bzw. Codewörter.

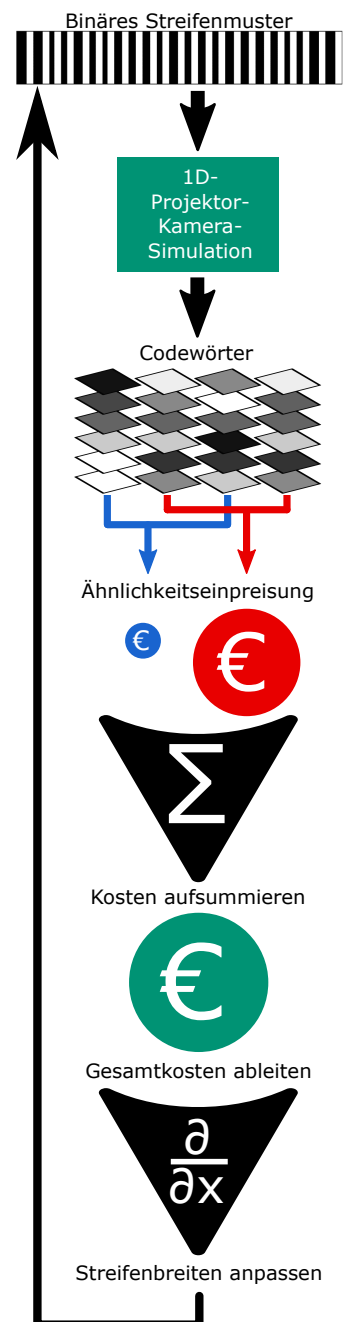


Abb. 4.5: Konzept der Musteroptimierung

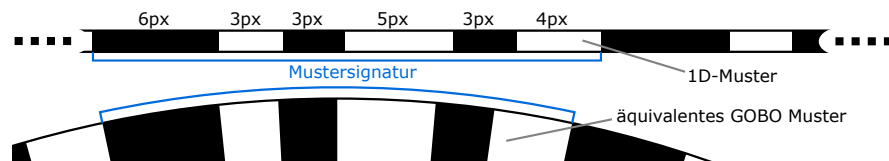
3. Vergleiche alle Codewörter paarweise mittels NCC.
4. Belege die NCC-Ähnlichkeitswerte mit einem Kostenwert: Sehr hohe Ähnlichkeitswerte produzieren hohe Kosten, mittlere und niedrige Ähnlichkeitswerte produzieren niedrige oder keine Kosten.
5. Addiere alle Kosten zu einem Gesamtkostenwert.
6. Berechne die Ableitung der Gesamtkosten in Abhängigkeit des ursprünglichen Streifenmusters.
7. Modifiziere die ursprünglichen Streifenbreiten derart, dass aufgrund der Ableitung eine Kostensenkung zu erwarten ist.
8. Wiederhole ab (2.).

Abbildung 4.5 zeigt den gesamten Ablauf in vereinfachter Form.

4.4.3 Projektor-Kamera-Simulation

Bei der GOBO-Projektion basiert die Musterentstehung auf einem Zusammenspiel von auf dem GOBO-Rad aufgetragenen binären Streifen, der Objektiv-Unschärfe und der Bildintegration der Kameras während der Rotation des GOBO-Rads (Abbildung 2.12). Um die tatsächlich von den Kameras gesehenen Helligkeitssequenzen, zu erhalten, die aus einem konkreten binären Streifenmuster entstehen, müssen diese Effekte also simuliert werden.

Abb. 4.6: Veranschaulichung der 1D-Approximation des GOBO-Musters: Oben: 1D-Modell in Pixeln (Mustersignatur [6, 3, 3, 5, 3, 4]). Unten: äquivalentes Muster auf dem GOBO-Rad. Jedes Pixel entspricht einem festen Winkel, z.B. $0,1^\circ$ [55].



Da es sich bei den Streifenmustern um eindimensionale Muster handelt, die tangential auf das GOBO-Rad aufgebracht wurden, ist es möglich, die wichtigen Eigenschaften des Musters mit einem 1D-Kamera-Projektor-Modell zu simulieren. Dieses Modell ersetzt einen Streifen auf dem GOBO-Rad durch einen Ausschnitt in einem 1D-Pixel-Array. Die Streifenbreite auf dem GOBO-Rad ist durch einen Winkel-Bereich definiert, wohingegen die Streifenbreite im 1D-Array durch eine Anzahl von Pixeln definiert ist. Beispielsweise könnten GOBO-Streifen mit Breiten von $0,5^\circ$, $0,3^\circ$ und $0,4^\circ$ in einen 1D-Streifen mit 5, 3 und 4 Pixeln übersetzt werden. Das Übersetzungsverhältnis betrüge dann 10 Pixel pro Grad. Das Übersetzungsverhältnis definiert außerdem eine Quantisierung des Musters in der Simulation (im Beispiel auf $0,1^\circ$), da ein Pixel nicht mehr weiter unterteilt wird.

Eine Liste von Streifenbreiten in Pixeln kann als *Mustersignatur* angesehen werden. Aus dieser Signatur kann der 1D-Streifen, also das Pixel-Array erzeugt werden, das nur schwarze (Helligkeit 0) und weiße (Helligkeit 1) Pixel enthält. Abbildung 4.6 zeigt ein Anschauungsbeispiel.

Dabei ist es nicht wichtig, ob ein Muster mit einem weißen oder schwarzen Streifen beginnt. Die Mustersignatur kann sowohl ein Muster repräsentieren, welches mit einem weißen Streifen, als auch das jeweils invertierte, welches mit einem schwarzen beginnt. Beide Muster haben die gleiche Selbstähnlichkeit, deshalb ist es für die Mustersoptimierung nicht erforderlich, sie zu unterscheiden.

Da es nicht erforderlich ist, dass das Muster auf dem GOBO-Rad über die gesamten 360° des Rads eindeutig ist, sondern die Eindeutigkeit nur in einem

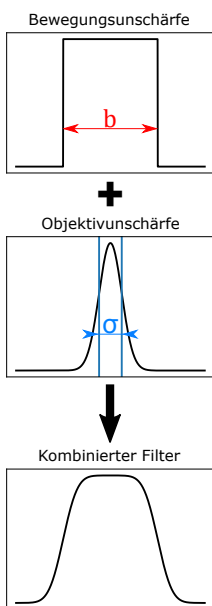


Abb. 4.7: Symbolhafte Darstellung der Filter-Kernel

anwendungsspezifischen, deutlich schmaleren Winkelbereich gewährleistet sein muss, werden die Muster so erzeugt, dass sie sich mehrmals auf dem Rad wiederholen. Dem entsprechend werden die Mustersignaturen so interpretiert, dass sie sich wiederholen. Beispielsweise sind die folgenden Mustersignaturen äquivalent, weil jede eine um 2 Streifen nach rechts rotierte Variante der vorigen darstellt: $[6, 3, 3, 5, 3, 4] \equiv [3, 4, 6, 3, 3, 5] \equiv [3, 5, 3, 4, 6, 3]$

Die 1D-Projektor-Kamera-Simulation erfolgt mittels zweier Filter (Abbildung 4.7):

1. Simulation der GOBO-Rad-Rotation während der Bildintegration der Kamera mittels eines Boxcar-Filters,
2. Simulation der Objektivunschärfe mittels eines Gauß-Filters.

Bei beiden Filtern handelt es sich um lineare Filter, deshalb ist die Reihenfolge, in der sie angewendet werden, nicht wichtig, d.h. sie sind kommutativ. Sie werden zu einem einzigen **Projektor-Kamera-Filter** ($F_{\sigma,b}$) kombiniert. $F_{\sigma,b}$ hat 2 Parameter: Die Objektivunschärfe wird repräsentiert durch die Standardabweichung σ (in Pixeln) des Gauß-Filters und die Rotationsweite des Musters während einer Bildintegration durch die Breite b (in Pixeln) des Boxcar-Filters.

Das gefilterte Muster wird verschoben und abgetastet (gesamplet), um die Codewörter, zu erhalten. Dabei sind die folgenden beiden Betrachtungsweisen äquivalent: 1. Das Muster wird N mal um eine Weite von r Pixeln verschoben und nach jeder Verschiebung wird an der gleichen Stelle ein Wert aufgenommen (Abbildung 4.8 unten). 2. Es werden N Werte an jeweils um r verschobenen Positionen aufgenommen (Abbildung 4.8 oben).

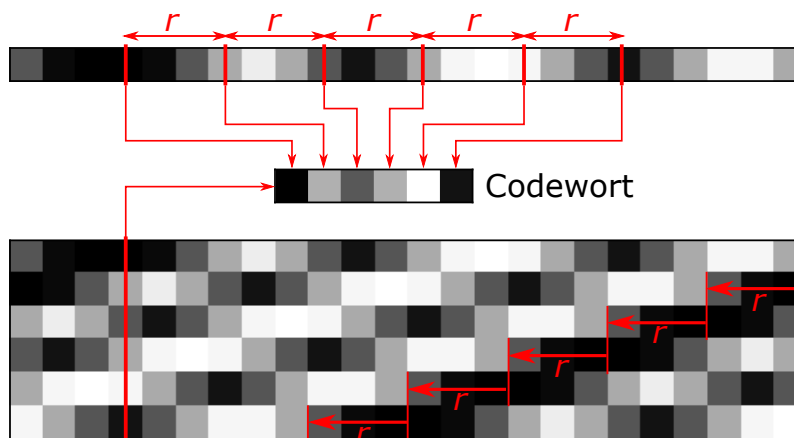


Abb. 4.8: Beispiel mit Bildsequenzlänge $N = 6$. Oben: das Muster, wie aus Abbildung 4.6 gefiltert mit dem Projektor-Kamera-Filter $F_{0,5,3}$. Unten: das gleiche Muster um $r = 3$ Pixel pro Zeile verschoben. Jede Spalte bildet ein Codewort (Mitte) welches aus $N = 6$ Helligkeitswerten besteht [55].

Anmerkung: es gibt einen Zusammenhang zwischen der Breite des Boxcar-Filters b und der Rotationsweite r des Musters pro Bild: Integrieren die Kameras 100% der Zeit bis zum Start der nächsten Bildaufnahme, so ist $b = r$. Wird dagegen z.B. nur die halbe zur Verfügung stehende Zeit integriert, so gilt $b = \frac{r}{2}$ usw..

Folgende real auftretende Effekte werden mit der 1D-Simulation nicht abgebildet:

- ▶ Die Tiefenabhängigkeit der Objektivunschärfe.
- ▶ Die relative Veränderung der Streifenbreite auf dem GOBO-Rad von innen (nahe der Achse) nach außen gegenüber der Objektivunschärfe.
- ▶ Unterschiede der Objektivunschärfe zwischen dem Zentral- und dem Randbereich des Projektionsobjektivs.

- Einschränkungen bei der Produktion realer GOBO-Räder, z.B. begrenzte Auflösung, ausfransen der Streifenkanten, etc..

4.4.4 Ähnlichkeitseinpreisung

Die Ähnlichkeit zweier Codewörter $\vec{x} = (x_1, \dots, x_N)$ und $\vec{y} = (y_1, \dots, y_N)$ wird mittels der normierten Kreuzkorrelation (NCC) $\rho(\vec{x}, \vec{y})$ bestimmt (siehe Definition 3.3.1). Je höher der NCC-Wert ρ , desto ähnlicher sind sich die Codewörter.

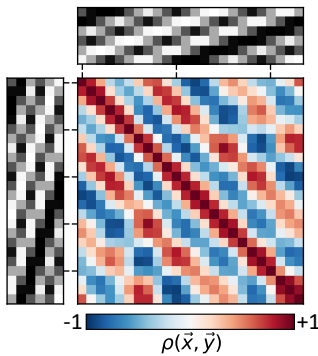


Abb. 4.9: NCC-Werte für alle Codewort-Paare eines simulierten Musters. Oben: die gefilterten und verschobenen Pixelwerte des Musters mit Sequenzlänge $N = 6$ (wie in Abbildung 4.8). Links: die gleichen Pixelwerte auf der vertikalen Achse. Mitte: Matrix, deren Einträge die NCC-Werte enthält, die sich beim Vergleich der Helligkeitssequenz aus der Spalte (von oben) und Zeile (von links) ergeben. NCC-Werte auf der Diagonale sind 1, weil dort ein Codewort mit sich selbst verglichen wird.

Um die Selbstähnlichkeit des gesamten Musters zu bestimmen, wird zunächst jedes Codewort mit jedem anderen verglichen, also die NCC zwischen allen Paaren berechnet. Für das Beispielmuster wird dies in Abbildung 4.9 dargestellt. Ein Beispiel für ein realistischeres (d.h. längeres) Muster wird in Abbildung 4.11 gezeigt.

Die NCC ist 1, wenn ein Codewort mit sich selbst verglichen wird, dies entspräche also der korrekten Zuordnung bei der Pixel-Korrespondenzsuche. Ein Muster hat eine geringe Selbstähnlichkeit, wenn alle Vergleiche zweier unterschiedlicher Codewörter einen niedrigen NCC-Wert haben. In diesem Fall lässt sich, auch mit Rauschen, die korrekte Zuordnung anhand des NCC-Maximums finden.

Ein gutes Muster produziert also beim paarweisen Codewort-Vergleich wenige hohe NCC-Werte. Dies wird mittels einer Gewichtungsfunktion berücksichtigt: jeder paarweise Codewort-Vergleich trägt, gewichtet mit einer nicht-linearen Funktion $C(\rho)$, zu den Gesamtkosten bei. $C(\rho)$ (Definition 4.4.1) wurde so angelegt, dass die Kosten überproportional mit der Höhe des NCC-Wertes steigen, NCC-Werte unter 0,5 jedoch keine Rolle mehr spielen. $C(\rho)$ wird in Abbildung 4.10 dargestellt.

Definition 4.4.1 Die Gewichtungsfunktion $C(\rho)$ ist definiert als:

$$C(\rho) = \begin{cases} (\rho - 0,5)^5 & \rho > 0,5 \\ 0 & \rho \leq 0,5 \end{cases} \quad (4.1)$$

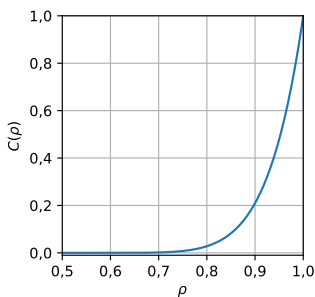


Abb. 4.10: Gewichtungsfunktion $C(\rho)$.

Abbildung 4.12 zeigt die Werte der Gewichtungsfunktion $C(\rho)$ angewendet auf das Beispiel aus Abbildung 4.11.

Die Gesamtkosten \bar{c} werden als Mittelwert der Kosten jedes einzelnen Codewort-Vergleichs berechnet.

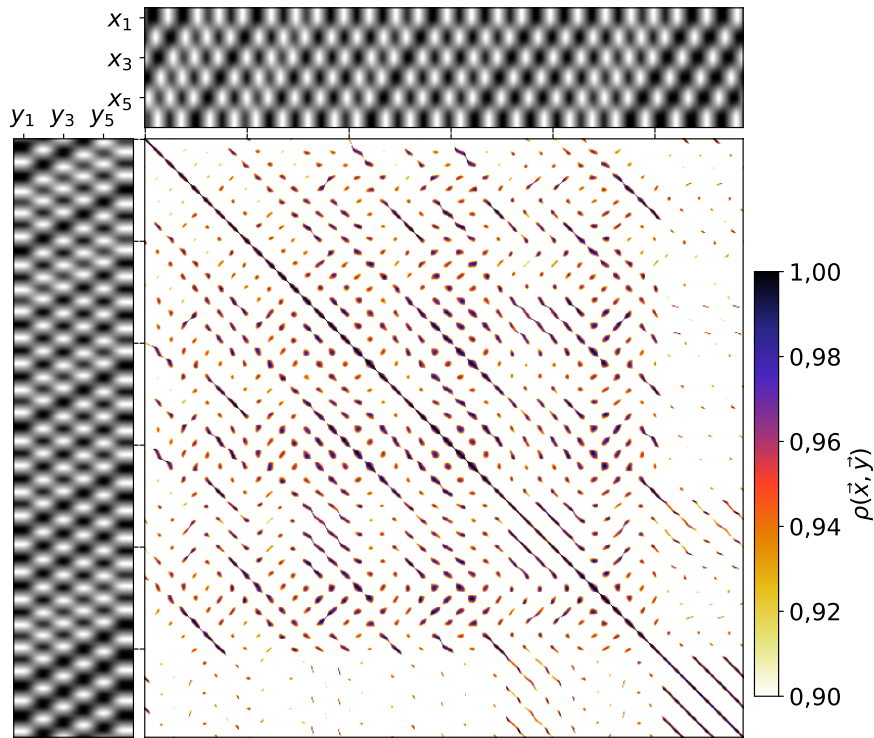


Abb. 4.11: Beispiel für NCC-Werte aller paarweisen Codewort-Vergleiche für ein Streifenmuster mit 23 zufällig erzeugten Streifenpaaren; Codewortlänge: $N = 6$. Es ist nur der für mögliche Verwechslungen kritische Bereich der NCC-Werte von 0,9 bis 1,0 dargestellt. Die hohen bzw. dunklen Bereiche abseits der Diagonalen repräsentieren Vergleiche von Codewörtern, die leicht miteinander verwechselt werden können.

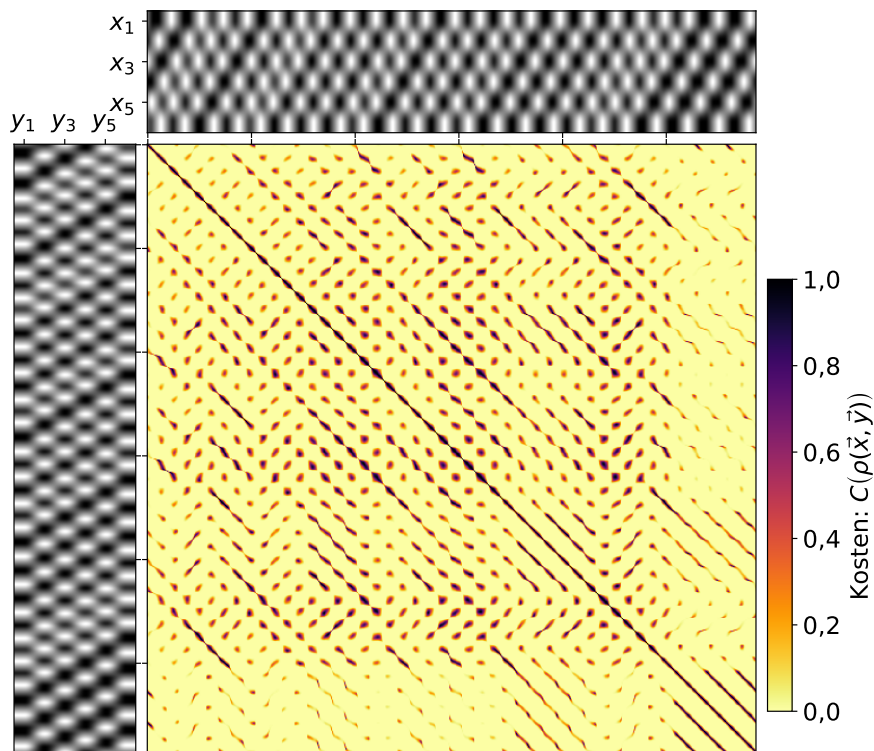


Abb. 4.12: Kostenfunktion angewendet auf die NCC-Werte aus Abbildung 4.11; Gesamtkosten: $\bar{c} = 0,056$

4.4.5 Mathematische Beschreibung der Kostenfunktion

In diesem Abschnitt ist die Kostenfunktion nochmals mittels mathematischer Formeln ausformuliert, dies beinhaltet die Projektor-Kamera-Simulation (unterteilt in Filterung und Abtastung des gefilterten Musters), die paarweisen Vergleiche der aus der Abtastung resultierenden Codewörter und die Einpreisung dieser Vergleiche. Diese mathematische Beschreibung bildet die Grundlage für die im darauffolgenden Abschnitt 4.4.6 beschriebene Ableitung dieser Kostenfunktion.

Gegeben seien die Bildsequenz- bzw. Codewort-Länge N , die Rotationsweite r (in Pixeln, vergleiche Abbildung 4.8) des Musters zwischen zwei aufeinanderfolgenden Aufnahmen, die Rotationsweite b während der Bildintegration*, die Objektivunschärfe σ (s.o.) und eine Liste mit binären Pixeln \vec{p} , die ein Streifenmuster auf dem GOBO-Rad repräsentieren. \vec{p} hat die Länge L .

*: $0 < b \leq r$

** : genauer gesagt ist $F_{\sigma,b}(\vec{p})$ ein linearer, digitaler Filter mit endlicher Impulsantwort (engl. Finite Impulse Response (FIR) Filter).

Die im Optimierungsprozess zu minimierenden Gesamtkosten \bar{c} des Musters werden mit folgenden Schritten berechnet:

1. Die 1D-Projektor-Kamera-Simulation ist ein linearer Filter** $F_{\sigma,b}(\vec{p})$, der von den binären, Pixeln $\vec{p} = (p_1, \dots, p_L)$ auf die von den Kameras gesehenen, nicht binären Pixel $\vec{f} = (f_1, \dots, f_L)$ abbildet. Er wird berechnet mittels Multiplikation der Filterkoeffizienten $\vec{a} = a_1, \dots, a_L$ mit den binären Pixeln \vec{p} : $F_{\sigma,b}(\vec{p})_k = f_k = \sum_{j=1}^N a_j \cdot p_{k+j-1}$.
2. Der Pixelstreifen \vec{f} wird abgetastet um die Codewörter zu erhalten: Jedes Codewort \vec{w}_i mit $i = \{1, \dots, L\}$ ist ein Vektor bestehend aus N Werten die aus \vec{f} gesampelt werden. Jedes Sample ist um r Pixel zum vorigen verschoben (vgl. Abb. 4.8). Dieser Samplingprozess wird mit den Funktionen $S_i(\vec{f}) = (f_i, f_{i+r}, f_{i+2r}, \dots, f_{i+Nr}) = \vec{w}_i$ beschrieben***. Für ein Paar zweier Codewörter wird im Folgenden die Notation $S_{i,j}(\vec{f}) = (S_i(\vec{f}), S_j(\vec{f})) = (\vec{w}_i, \vec{w}_j)$ verwendet.
3. Die normierte Kreuzkorrelation $\rho_{i,j} = \rho(\vec{w}_i, \vec{w}_j) = \rho(S_{i,j}(\vec{f}))$ zweier Codewörter wird für alle Codewort-Paare berechnet. Diese Werte bilden zusammen eine Matrix der Größe $L \times L$ (Beispiele in Abbildungen ?? und 4.11).
4. Mit der Gewichtungsfunktion $C(\rho)$ (Abbildung 4.10) werden aus den NCC-Werten die Kosten $C_{i,j} = C(\rho_{i,j})$ jedes Codewort-Vergleichs aus dessen NCC-Wert $\rho_{i,j}$ berechnet. Daraus resultiert eine Kostenmatrix (Beispiel in Abbildung 4.12).
5. Berechnung der Gesamtkosten als Mittelwert der Kostenmatrix: $\bar{c} = \frac{1}{L^2} \sum_{i=1}^L \sum_{j=1}^L C_{i,j}$.

***: die Indizes von f sind modulo L zu verstehen

Die Verkettung dieser einzelnen Funktionen ergibt folgende Gesamtfehlerfunktion:

$$\begin{aligned} \bar{c} &= \frac{1}{L^2} \sum_{i=1}^L \sum_{j=1}^L C \circ \rho \circ S_{i,j} \circ F_{\sigma,b}(\vec{p}) \\ &= \frac{1}{L^2} \sum_{i=1}^L \sum_{j=1}^L C \left(\rho \left(S_{i,j} \left(F_{\sigma,b}(\vec{p}) \right) \right) \right) \end{aligned} \quad (4.2)$$

4.4.6 Ableitung der Kostenfunktion

Die Kostenfunktion (Gleichung 4.2) kann mittels der Kettenregel der Differentialrechnung abgeleitet werden:

$$\begin{aligned} \nabla \bar{c} &= \left(\frac{\partial \bar{c}}{\partial p_1}(\vec{p}), \dots, \frac{\partial \bar{c}}{\partial p_L}(\vec{p}) \right) \\ &= \frac{1}{L^2} \sum_{i=1}^L \sum_{j=1}^L \left(\frac{dC}{d\rho}(\rho_{i,j}) \cdot \nabla \rho(\vec{w}_i, \vec{w}_j) \cdot J_{S_{i,j}}(\vec{f}) \cdot J_{F_{\sigma,b}}(\vec{p}) \right) \end{aligned} \quad (4.3)$$

Von rechts nach links sind dies:

*: d.h. die Matrix mit allen partiellen Ableitungen

- ▶ $J_{F_{\sigma,b}}(\vec{p})$ die Jacobi-Matrix* des Projektor-Kamera-Filters $F_{\sigma,b}$,
- ▶ $J_{S_{i,j}}(\vec{f})$ die Jacobi-Matrix der Sampling-Funktion $S_{i,j}$,
- ▶ $\nabla \rho(S_{i,j})$ der Gradient der NCC ρ angewendet auf die Codewörter (\vec{w}_i, \vec{w}_j) ,
- ▶ $\frac{dC}{d\rho}(\rho_{i,j})$ die Ableitung der Gewichtungsfunktion $C(\rho)$.

Diese Teile der Verkettung werden im Folgenden einzeln beschrieben.

Ableitung des Projektor-Kamera-Filters

Ein Pixel f_k des gefilterten Musters \vec{f} wird als Skalarprodukt eines Vektors von Filter-Koeffizienten $\vec{a} = a_1, \dots, a_L$ mit dem Vektor der binären Musterpixel $\vec{p} = p_1, \dots, p_L$, verschoben um $k \pmod{L}$, berechnet:

$$f_k = \sum_{j=1}^N a_j \cdot p_{k+j-1} \quad (4.4)$$

Die Berechnung des gesamten Vektors \vec{f} kann deshalb als Matrix-Multiplikation mit der Matrix F beschrieben werden:

$$\vec{f} = F \cdot \vec{p} = \begin{pmatrix} a_1 & a_2 & \dots & a_{L-1} & a_L \\ a_L & a_1 & \dots & a_{L-2} & a_{L-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_3 & a_4 & \dots & a_1 & a_2 \\ a_2 & a_3 & \dots & a_L & a_1 \end{pmatrix} \cdot \vec{p} \quad (4.5)$$

Da die Filter-Koeffizienten konstant sind, ist die Jacobi-Matrix gleich der Matrix F :

*: Indizes jeweils (mod L)

$$\frac{\partial}{\partial p_i} f_k = a_{(i-k+1) \pmod{L}} \quad \text{bzw.} \quad J_F(\vec{p}) = J_F = F \quad F \in \mathbb{R}^{L \times L} \quad (4.6)$$

Ableitung der Sampling-Funktion

Die Sampling-Funktion $S_i(\vec{f})$ wählt N Werte $f_i, f_{i+r}, f_{i+2r}, \dots, f_{i+(N-1)r}$ aus \vec{f} . Sie kann deshalb als Matrixmultiplikation mit einer binären Matrix $S_i \in \{0, 1\}^{N \times L}$ beschrieben werden, die auf jeder Zeile genau eine Eins und ansonsten Nullen hat. Analog dazu kann die Samplingfunktion für ein Codewort-Paar $S_{i,j}(\vec{f}) = (S_i(\vec{f}), S_j(\vec{f}))$ als Matrix $S_{i,j} \in \{0, 1\}^{2N \times L}$ beschrieben werden.

Die Jacobi-Matrix dieser Funktion ist also gleich dieser konstanten Matrix:

$$J_{S_{i,j}} = S_{i,j} \quad S_{i,j} \in \{0, 1\}^{2N \times L} \quad (4.7)$$

Ableitung der NCC

Die normierte Kreuzkorrelation ρ zweier Codewörter $\vec{x} = x_1, \dots, x_N$ und $\vec{y} = y_1, \dots, y_N$ ist definiert als

$$\rho(\vec{x}, \vec{y}) = \frac{1}{N} \sum_{i=1}^N (n_{x_i} \cdot n_{y_i}) \quad \text{mit} \quad n_{x_i} = \frac{x_i - \bar{x}}{\sigma_x} \quad (4.8)$$

wobei

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad \text{und} \quad \sigma_x = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (4.9)$$

Die partiellen Ableitungen dieser Terme sind:

$$\frac{\partial}{\partial x_i} \bar{x} = \frac{1}{N} \quad \text{und} \quad \frac{\partial}{\partial x_i} \sigma_x = \frac{n_{x_i}}{N} \quad (4.10)$$

$$\frac{\partial}{\partial x_i} n_{x_i} = \frac{1}{\sigma_x} - \frac{1 + n_{x_i}^2}{N \sigma_x} \quad \text{und} \quad \frac{\partial}{\partial x_i} n_{x_j} = -\frac{1 + n_{x_i} \cdot n_{x_j}}{N \sigma_x} \quad \forall j \neq i \quad (4.11)$$

Daraus folgen die partiellen Ableitungen der NCC:

$$\frac{\partial}{\partial x_i} \rho(\vec{x}, \vec{y}) = \frac{1}{N \sigma_x} \left(n_{y_i} - \frac{n_{x_i}}{N} \sum_{j=1}^N n_{x_j} \cdot n_{y_j} \right) = \frac{n_{y_i} - n_{x_i} \cdot \rho(\vec{x}, \vec{y})}{N \sigma_x} \quad (4.12)$$

Der Gradient von $\rho(\vec{x}, \vec{y})$ ist ein Zeilenvektor:

$$\nabla \rho(\vec{x}, \vec{y}) = \left(\frac{\partial \rho}{\partial x_1}, \dots, \frac{\partial \rho}{\partial x_N}, \frac{\partial \rho}{\partial y_1}, \dots, \frac{\partial \rho}{\partial y_N} \right) \in \mathbb{R}^{2N} \quad (4.13)$$

Ableitung der Kosten-Gewichtungsfunktion

Die Kosten-Gewichtungsfunktion $C(\rho)$ ist definiert als:

$$C(\rho) = \begin{cases} (\rho - 0,5)^5 & \rho > 0,5 \\ 0 & \rho \leq 0,5 \end{cases} \quad (4.14)$$

Die Ableitung ist:

$$\frac{d}{d\rho} C(\rho) = \begin{cases} 5 \cdot (\rho - 0,5)^4 & \rho > 0,5 \\ 0 & \rho \leq 0,5 \end{cases} \quad (4.15)$$

4.4.7 Iterative Anpassung der Streifenbreiten

In diesem Abschnitt wird die iterative Anpassung der Streifenbreiten mit dem Ziel der Minimierung der Gesamtkosten \bar{c} beschrieben.

Zunächst wird ein zufälliges binäres Muster mit Pixelhelligkeiten \vec{p}_0 und gleichverteilten Streifenbreiten erzeugt. Für dieses Muster werden die Gesamtkosten $\bar{c}(\vec{p}_0)$ und deren Ableitung nach den Pixelhelligkeiten $\nabla\bar{c}(\vec{p}_0)$ berechnet. Um das Muster zu optimieren, müssten eigentlich die Pixelhelligkeiten gemäß der Ableitung heller oder dunkler gemacht werden. Da es sich aber um ein binäres Streifenmuster handelt, kann ein Pixel nur ganz hell oder ganz dunkel sein. Die Pixelhelligkeit kann nur an den Kanten der Streifen verändert werden, indem diese entweder in Richtung des helleren oder dunkleren Streifens verschoben werden. Die Streifengrenzverschiebung führt zu einem neuen Muster mit Pixelhelligkeiten \vec{p}_1 . Abbildung 4.13 zeigt ein Musterbeispiel mit Ableitungswerten und durch diese vorgegebene Richtungen, in die die Streifengrenzen verschoben werden müssen.

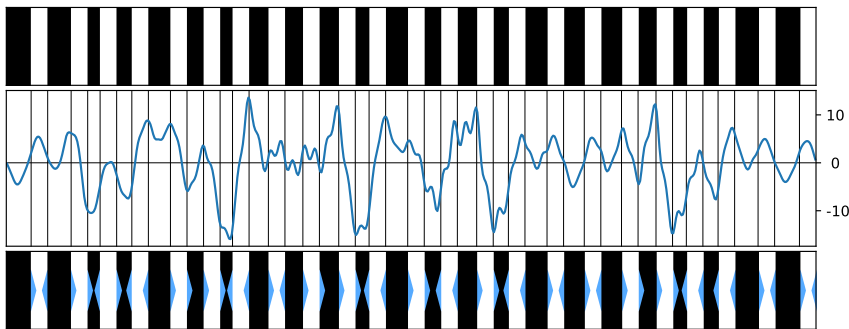


Abb. 4.13: Anpassung des Musters gemäß Kostenableitung. Oben: zufälliges Muster. Mitte: Ableitung $\nabla\bar{c}$ der Kosten nach den Pixelhelligkeiten \vec{p} . Unten: Die blauen Dreiecke zeigen die Richtung an, in die die Streifengrenzen gemäß der Ableitung verschoben werden müssen, um die Kosten zu reduzieren.

Dieser Prozess wird iterativ wiederholt, um schrittweise die Muster $\vec{p}_2, \dots, \vec{p}_n$ zu erhalten. Bei der Streifenbreitenverschiebung wird eine festgelegte minimale und maximale Streifenbreite niemals überschritten. Im Beispiel aus den Abbildungen 4.11, 4.12, 4.14 und 4.15 liegen die Grenzen bei 9 px und 18 px.

Es werden mehrere Verschiebungen gleichzeitig ausgeführt. Dazu wird für jede Streifengrenze eine Korrekturweite (in Pixeln) erzeugt. Die Korrekturweite zeigt immer in die vom Gradienten vorgegebene Richtung. Die Entfernung ist zufällig normalverteilt (mit Erwartungswert 0 und Standardabweichung S proportional zum Gradienten $\nabla\bar{c}$). Die Korrekturweiten für alle Streifengrenzen bilden einen Korrekturvektor, der auf die Positionen der Streifengrenzen angewendet wird. Danach wird überprüft, ob das so modifizierte Muster tatsächlich niedrigere Gesamtkosten \bar{c} hat. Ist dies der Fall, wird es behalten, ansonsten wird mit dem alten Muster weitergemacht und von neuem ein Korrekturvektor zufällig erzeugt, wobei die Länge des Vektors schrittweise reduziert wird*.

*: S wird bei jedem Schritt reduziert

Beispieloptimierung

Beispielhaft wird hier ein Optimierungslauf dargestellt für das Muster, für das bereits Zwischenergebnisse in den Abbildungen 4.11 und 4.12 gezeigt wurden. Die Bildsequenzlänge wurde auf $N = 6$ festgelegt, es wurden Streifenbreiten im Intervall $[9 \text{ px}, 18 \text{ px}]$ erlaubt, die Rotationsgeschwindigkeit wurde auf $r = 12 \text{ px}$ pro Bild und die Objektivunschärfe auf $\sigma = 1 \text{ px}$ festgelegt. Abbildung 4.14 zeigt links einen Ausschnitt des sich verändernden Musters, rechts die Kosten jedes dieser Muster. Abbildung 4.15 zeigt die Kostenmatrix für das optimierte Muster. Im Vergleich zum unoptimierten Muster (siehe dessen Kostenmatrix in Abbildung 4.12) sind die Kostenwerte in den Randbereichen stark reduziert.

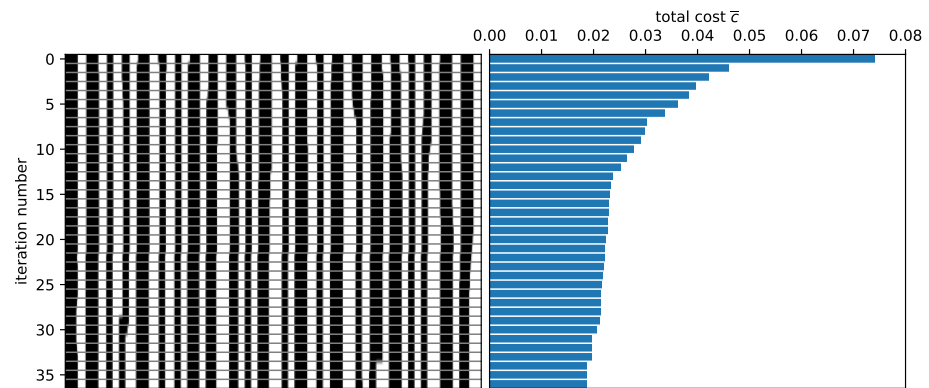


Abb. 4.14: Visualisierung eines Optimierungslaufs [55]. Links: Ausschnitt aus dem Muster nach jedem erfolgreichen Optimierungsschritt (oben: zufälliges Startmuster, unten: optimiertes Muster). Rechts: Gesamtkosten \bar{c} des jeweiligen Musters.

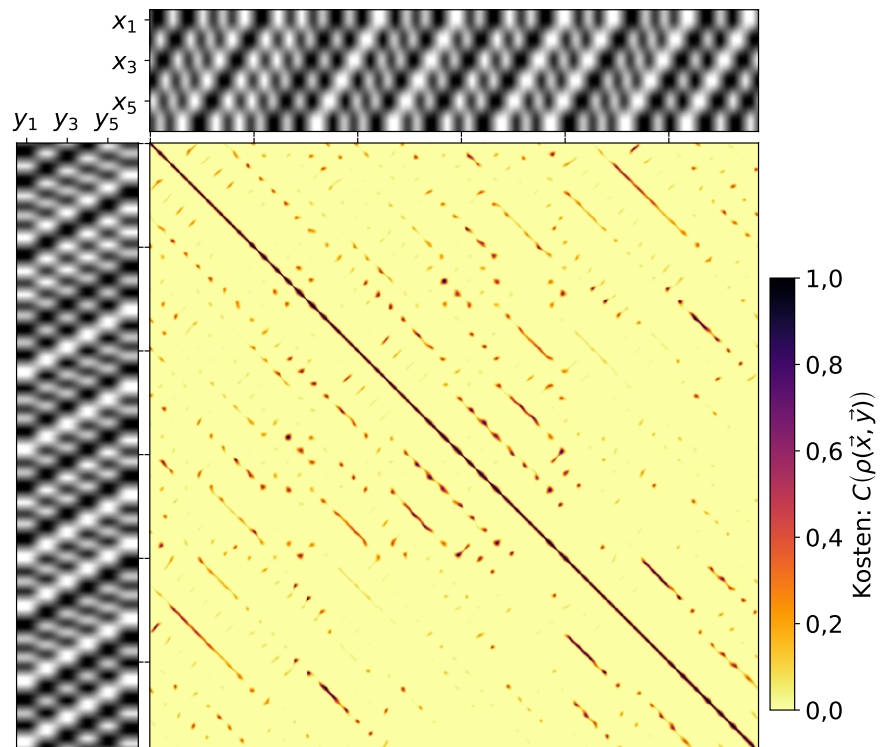


Abb. 4.15: Kostenmatrix des optimierten Musters; Gesamtkosten $\bar{c} = 0,013$ (vgl. Kostenmatrix des nicht-optimierten Musters in Abbildung 4.12).

4.5 Software-Tool zur Musteroptimierung

Die bis hier beschriebene Musteroptimierungsmethode wurde in ein vom Autor entwickeltes Software-Tool integriert. Dieses erlaubt, GOBO-Muster für unterschiedlichste Sensoren zu generieren. Damit ist es möglich, für verschiedene Kameraabstände und -auflösungen, Messvolumenausdehnungen und Projektor-Geometrien Muster zu optimieren. Als Ergebnis kann das Tool CAD-Dateien in mehreren Formaten generieren, darunter PDF und DXF, die für die Herstellung von GOBO-Musterrädern eingesetzt werden können.

4.6 Autokorrelation der optimierten Muster

Es wurde eine Autokorrelationsanalyse von zwei optimierten GOBO-Mustern mittels der in Abschnitt 4.2 beschriebenen Methode durchgeführt. Abbildung 4.16 zeigt die Autokorrelationswerte für ein GOBO-Rad, welches auf kontinuierliche Projektion mit Bildsequenzlänge $N = 6$ optimiert wurde*. Abbildung 4.17 zeigt die Autokorrelationswerte für ein GOBO-Rad, welches auf gepulste Projektion mit $N = 6$ optimiert wurde**.

*: Es handelt sich um das in Abschnitt 5.2 beschriebene GOBO L6.

** : GOBO K6.

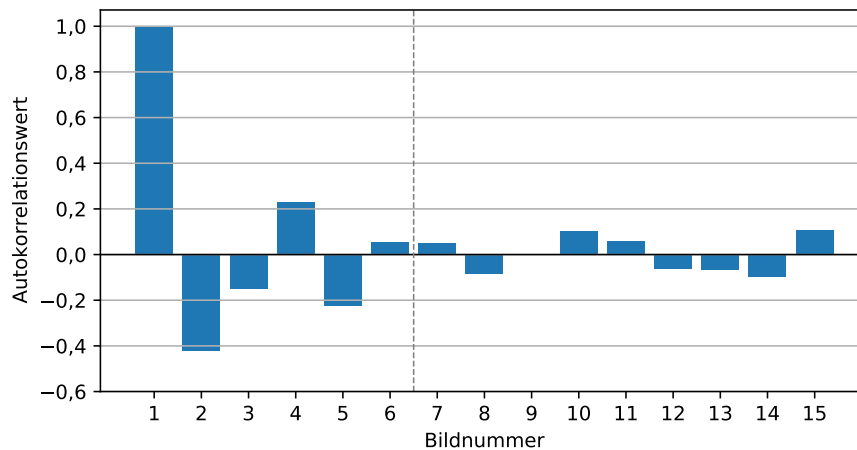


Abb. 4.16: Zeitliche Autokorrelation eines auf Bildsequenzlänge $N = 6$ optimierten GOBO-Musters bei kontinuierlicher Projektion. Die gestrichelte Linie symbolisiert die Grenze einer Helligkeitssequenz der Länge $N = 6$. Gegenüber dem nicht optimierten Muster (Abbildung 4.2) sind die Autokorrelationswerte deutlich reduziert.

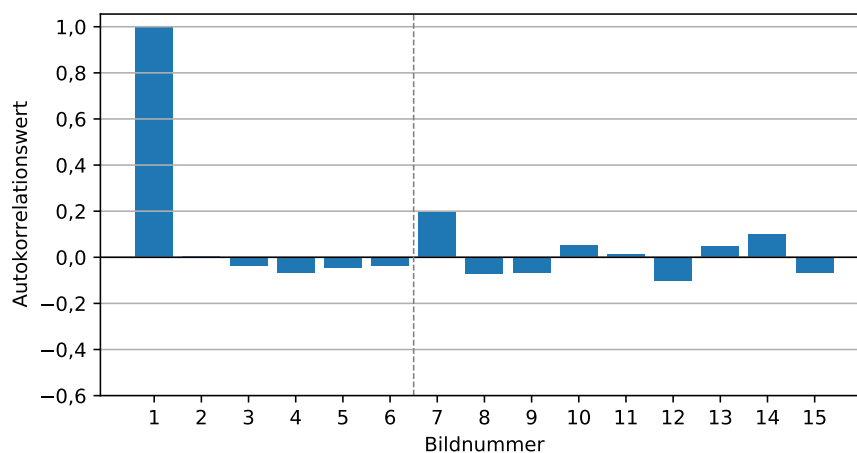


Abb. 4.17: Zeitliche Autokorrelation der von den Kameras aufgenommenen Helligkeitswerte eines auf Sequenzlänge $N = 6$ optimierten GOBO-Musters bei gepulster Projektion. Die ersten sechs Helligkeitswerte weisen keine signifikante Autokorrelation mehr auf. Erst der sechste Folgewert, der bei der Optimierung nicht mehr berücksichtigt wurde, zeigt wieder eine geringfügige Korrelation mit dem ersten.

Die Ergebnisse zeigen, dass sich die Autokorrelationswerte mittels der Musteroptimierung betragsmäßig deutlich reduzieren lassen. In der Kombination mit

dem gepulsten Projektionsmodus gibt es für das auf Sequenzlänge $N = 6$ optimierte Muster keine signifikante Korrelation der ersten sechs Helligkeitswerte miteinander.

4.7 Zusammenfassung

In diesem Kapitel wurden zwei Verbesserungen der Musterprojektion vorgeschlagen:

1. eine gepulste Beleuchtung
2. ein Optimierungsverfahren für die Erhöhung der Mustereindeutigkeit der Projektionsmuster.

Die gezeigten Musterkorrelationsanalysen, insbesondere die in Abbildung 4.17 gezeigte, sind ein starkes Indiz dafür, dass mit der Kombination dieser beiden Verbesserungen die statistische Unabhängigkeit der projizierten Codewörter erreicht werden kann.

Im folgenden Kapitel 5 wird gezeigt, dass sich damit bei Bildsequenzlänge $N = 6$ quasi ausfallfreie 3D-Punktwolken erzielen lassen. Gegenüber der sonst üblicherweise eingesetzten Sequenzlänge $N = 10$ bedeutet dies eine Reduktion der Aufnahmelatenz um 40 %.

Um die Auswirkungen der in Kapitel 3 und 4 vorgeschlagenen Verbesserungen für ein reales Messsystem bewerten zu können, werden in diesem Kapitel Messungen und deren Ergebnisse vorgestellt. Insbesondere wird untersucht, wie stark sich die Bildsequenzlänge gegenüber der sonst üblicherweise verwendeten von $N = 10$ reduzieren lässt, wenn der schnelle BiCoS-Algorithmus zusammen mit dem gepulsten Projektionsmodus und optimierten Projektionsmustern eingesetzt wird.

Es wurden mehrere GOBO-Räder produziert. Diese wurden in einem am Fraunhofer IOF entwickelten 3D-Sensor getestet. Außerdem wurden zwei Messszenen aufgebaut und jeweils von einer festen Position aus vermessen.

5.1 Sensorsystem und Messszenen

Beide Messszenen haben eine Ausdehnung von ca. $2\text{ m} \times 1\text{ m}$. Die erste Messszenen ist sehr kontrastreich. Sie enthält verschiedene Objekte, deren Oberflächen von schwarz bis weiß reichen, und stellt somit eine schwer messbare Szene dar. Sie soll das Messsystem an seine Grenzen bringen. Abbildung 5.1 zeigt diese Szene (rot umrandet) und den davor aufgestellten Sensor. Die zweite Messszenen enthält nur helle Objekte. (Abbildung 5.2)

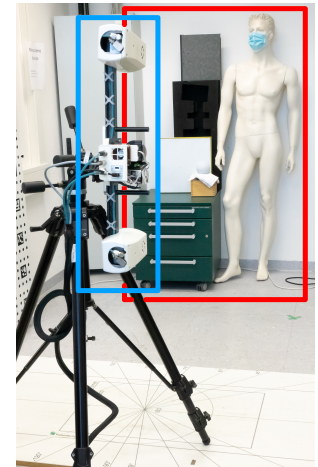


Abb. 5.1: Rot umrandet: kontrastreiche Messszenen, blau umrandet: Sensor

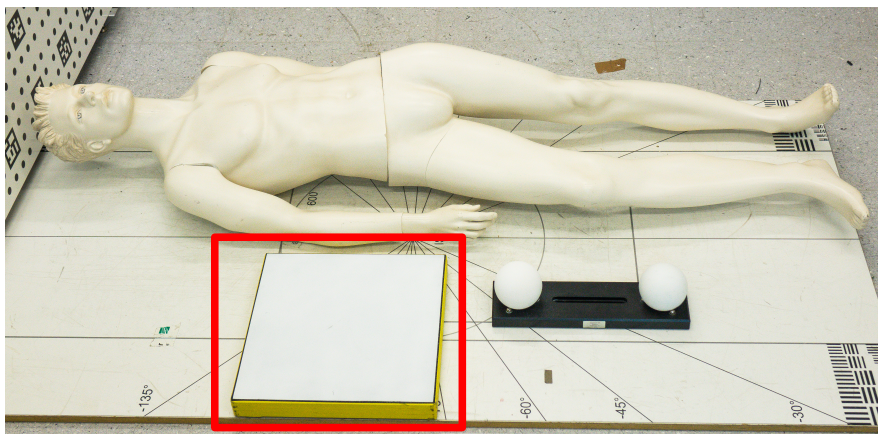


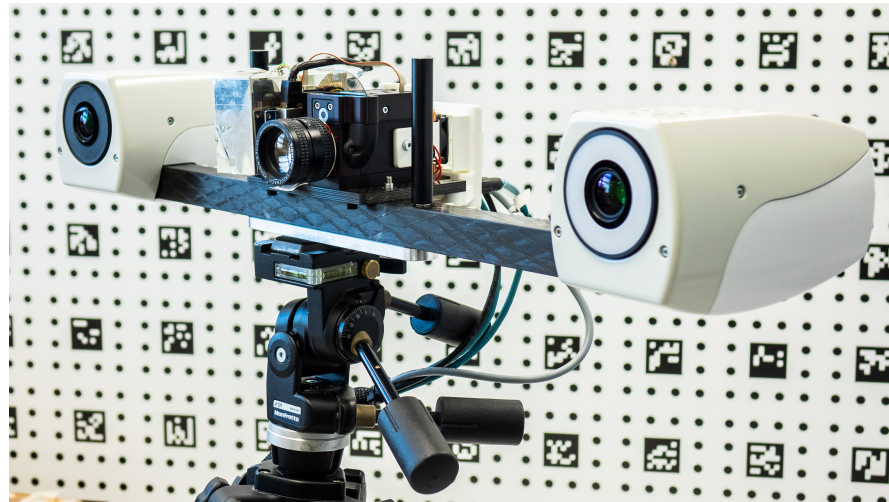
Abb. 5.2: Messszenen mit hellen Objekten. Rot umrandet: Ebenennormal

Der Sensor (Abbildung 5.3) besteht aus einem Stereo-Kamera-Paar mit einem Kameraabstand von 61 cm und einem GOBO-Projektor (Abbildung 5.4), der Nah-Infrarot-Licht projiziert (mittlere Wellenlänge: 850 nm). Zwischen den Kameraobjektiven und den Bildsensoren sind Langpassfilter eingebaut, welche das sichtbare Licht blockieren, für das Projektionslicht aber durchlässig sind. Dies dient der Reduktion des mitaufgenommenen Umgebungslichts. In Tabelle 5.1 sind die Sensoreigenschaften aufgelistet.

Tab. 5.1: Eigenschaften des 3D-Sensors, der für die Messungen verwendet wurde.

Messfeldgröße	2 m × 1 m
Kameras	Baumer VLXT-28M.I (GigE-Vision)
Kameraabstand	61 cm
Kameraauflösung für Messung	1920 × 960
Bildeinzugsrate	100 Hz
Wellenlänge des Projektionslichts	850 nm (NIR)
GOBO-Durchmesser	60 mm
durchleuchtete GOBO-Fläche	20 mm × 10 mm

Abb. 5.3: Sensor mit den zwei eingehausten Messkameras (links und rechts) und geöffnetem GOBO-Projektor (Mitte) vor einem für die Sensorkalibrierung verwendeten Muster.



5.2 GOBO-Räder



Abb. 5.4: GOBO-Projektor des Sensors mit abgenommenem Projektorgehäuse.

Es wurden GOBO-Räder (Beispiel in Abbildung 1.6) für zwei Messmodi optimiert: Ein **Messmodus mit gepulstem Projektionslicht**, bei dem die Kamerateintegrationszeit und die Beleuchtungsdauer für ein Bild jeweils 3,9 ms beträgt, und ein **Messmodus mit kontinuierlichem Projektionslicht**, bei dem die Integrationszeit 9,8 ms beträgt und das Projektionslicht kontinuierlich an ist. In beiden Modi beträgt die Bildeinzugsrate 100 Hz. Zu beachten ist hierbei erstens, dass bei kontinuierlicher Projektion **nicht** mehr Projektionslicht emittiert wird, da die Projektor-LED in diesem Modus weniger effizient arbeitet, und dass zweitens im gepulsten Messmodus das GOBO-Rad innerhalb der 3,9 ms die gleiche Rotationsweite hat, die es im kontinuierlichen Messmodus innerhalb von 9,8 ms hat. D.h. im gepulsten Modus dreht sich das GOBO-Rad um 151 % schneller und damit auch entsprechend weiter, bevor das nächste Bild aufgenommen wird.

Es wurden insgesamt 7 verschiedene Muster erzeugt (Tabelle 5.2) und auf GOBO-Räder aufgebracht: ein rein zufälliges, nicht optimiertes Muster mit gleichverteilten Streifenbreiten, eine Serie mit drei Mustern, die auf kontinuierliche Musterprojektion optimiert wurden, und eine Serie mit weiteren drei Mustern, die auf gepulste Projektion optimiert wurden.

Bei allen GOBO-Rädern sind an den Rändern des Projektionsfeldes 44 bzw. 58 Streifenpaare sichtbar. Die beiden unterschiedlichen Anzahlen rühren von den unterschiedlichen Winkelbereichen auf dem GOBO-Rad her, welche auf der radial außen bzw. innen liegenden Seite des Projektionsfeldes abgebildet werden. Der maximal zu vermessende Tiefenbereich wurde auf 1,5 m bis 6 m Entfernung zum Sensor eingeschränkt. Deshalb konnte auch der abzusuchende Disparitätsbereich und damit auch der Musterbereich, der eindeutig sein muss,

Name	optimiert für		
	Sequenzlänge	Belichtungszeit	Rotationsgeschw.
GOBO K6	6	3,9 ms	10,25 U/min
GOBO K7	7	3,9 ms	10,25 U/min
GOBO K8	8	3,9 ms	10,25 U/min
GOBO L6	6	9,8 ms	4,05 U/min
GOBO L7	7	9,8 ms	4,05 U/min
GOBO L8	8	9,8 ms	4,05 U/min
GOBO R	nicht optimiert		

Tab. 5.2: Liste der getesteten GOBO-Räder.

auf 29 Streifenpaare eingeschränkt werden. Diese 29 Streifenpaare wiederholen sich auf den optimierten GOBOs jeweils insgesamt 17-mal. D.h. auf jedem optimierten GOBO gibt es jeweils 17 Abschnitte, die sich gleichen und von denen jeder einen Winkelbereich von $360/17$ einnimmt.

Für die Optimierungen der GOBO-Muster, die gemäß der in Kapitel 4 beschriebenen Methode durchgeführt wurden, wurden folgende gemeinsame Parameter gewählt:

- ▶ Musterlänge: 29 Streifenpaare,
- ▶ erlaubte Streifenbreiten: zwischen 9 px bis einschließlich 18 px,
- ▶ Objektivunschärfe: $\sigma = 1$ px,
- ▶ Rotationsweite zwischen zwei Bildern: $r = 9$ px für den kontinuierlichen Messmodus bzw. $r = 23$ px für den gepulsten Messmodus.

Es wurden für jedes Rad jeweils 100 zufällige Muster mit gleichverteilten Streifenbreiten erzeugt und dann jeweils iterativ optimiert. Danach wurde jeweils unter den 100 optimierten Ergebnissen das gemäß der Kostenfunktion beste Muster ausgewählt und per Laser-Gravur auf ein Glas-GOBO aufgebracht.

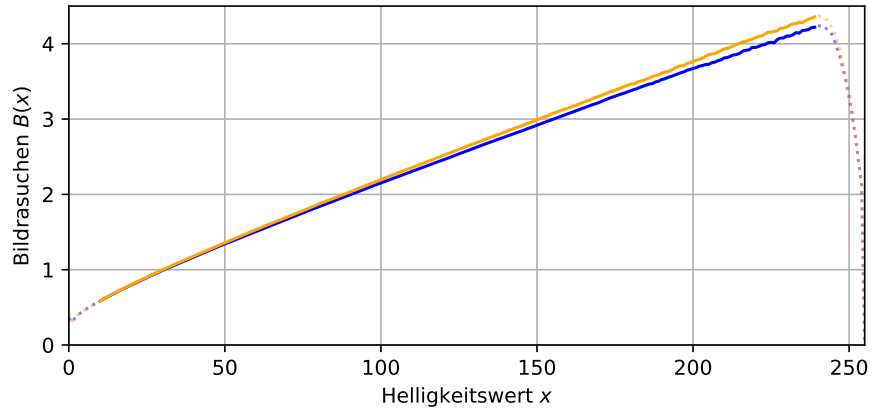
5.3 Bildrauschen der Kameras

Das Bildrauschen der Kameras wurde folgendermaßen bestimmt: Es wurden 1000 Bilder der kontrastreichen Messszene mit aufprojiziertem, statischem Muster (d.h. stillstehendem GOBO-Rad) aufgenommen. Die so erstellten Aufnahmen weisen einen sehr hohen Kontrast auf, so dass der komplette Dynamikumfang der Kamerasensoren ausgenutzt wurde, also alle 256 Helligkeitswerte. Gäbe es kein Bildrauschen, so müssten alle 1000 Bilder exakt gleich sein, da die Szene während der gesamten Aufnahmezeit unverändert war. Deshalb unterscheiden sich die 1000 Bilder ausschließlich um das Rauschen.

Es wurde für jedes Pixel der Median und die Standardabweichung der 1000 Helligkeitswerte berechnet. Danach wurden alle Pixel mit gleichem Median zusammengefasst und deren Standardabweichungen gemittelt. In Abbildung 5.5 sind diese gemittelten Standardabweichungen $B(x)$ gegen die Median-Helligkeiten x aufgetragen.

Die Standardabweichungen $B(x)$ kann man als Rausch-Level interpretieren. Zu beachten ist, dass mit dieser Methode für sehr niedrige und sehr hohe Helligkeitswerte x keine realistischen Rausch-Level berechnet werden können. Der Grund ist, dass dort der Sensor nicht mehr linear arbeitet, wodurch auch das zum Signal addierte Rauschen keinen Erwartungswert von 0 mehr hat. Ein nicht-rauschbehafteter Helligkeitswert von 253 kann z.B. durch das Rauschen nur noch um maximal 2 Werte heller, aber um viele Werte niedriger werden,

Abb. 5.5: Empirisch bestimmtes Bildrauschen $B(x)$ für die linke (blau) und rechte Kamera (orange) in Abhängigkeit von der Helligkeit eines Pixels x . Die gestrichelten Bereiche enthalten unrealistische Werte und müssen als Artefakt der Berechnungsmethode interpretiert werden (siehe Text).



d.h. der Erwartungswert des Rauschens ist < 0 , und damit bekommt auch der real gemessene Helligkeitswert einen Erwartungswert < 253 .

Um ein Signal-zu-Rausch-Verhältnis (Signal-to-Noise-Ratio, SNR) zu bestimmen, muss zusätzlich zum Bildrauschen $B(x)$ noch die Stärke des Signals bestimmt werden. Diese kann in den Bildstapeln einer 3D-Messung ermittelt werden. Bei der 3D-Messung rotiert das GOBO-Rad, deshalb hat die Helligkeitssequenz \vec{x} jedes Pixels eine hohe zeitliche Varianz. Je größer die zeitliche Standardabweichung $S(\vec{x})$ der Helligkeitssequenz ist, desto besser lassen sich die Helligkeitswerte unterscheiden. Deshalb kann $S(\vec{x})$ als Signalstärke interpretiert werden. Mit der mittleren Helligkeit der Helligkeitssequenz \bar{x} kann der für sie relevante Rauschwert $B(\bar{x})$ bestimmt werden. Damit ergibt sich das SNR als: $SNR = S(\vec{x})/B(\bar{x})$

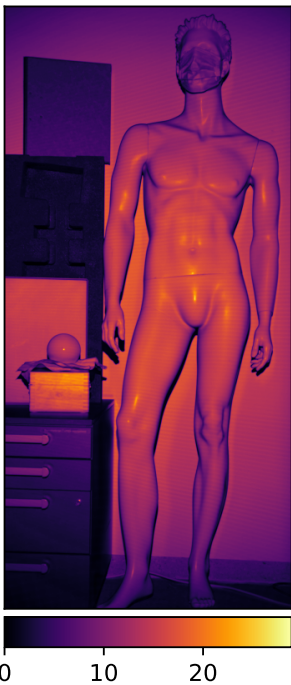


Abb. 5.6: (oben) Signal-zu-Rausch-Verhältnis für eine Bildsequenz der kontrastreichen Messszene. Gemessen mit GOBO K6 im gepulsten Projektionsmodus.

Abbildung 5.6 zeigt dieses Signal-zu-Rausch-Verhältnis für eine Bildsequenz der kontrastreichen Messszene. Das SNR ist in einigen Bereichen der Messszene sehr niedrig (unter 5). Bei diesem niedrigen SNR ist kein gutes Rekonstruktionsergebnis zu erwarten.

In der Messszene mit den hellen Objekten wurde das Signal-zu-Rausch-Verhältnis auf dem Ebenennormal (rot umrandet in Abbildung 5.2) für alle untersuchten GOBO-Räder bestimmt. Abbildung 5.7 zeigt die Ergebnisse.

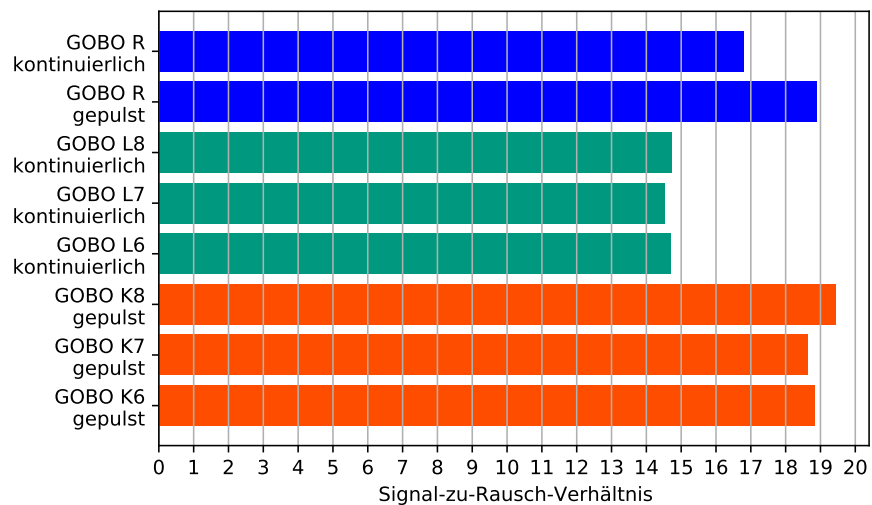


Abb. 5.7: (rechts) Signal-zu-Rausch-Verhältnis auf dem Ebenennormal (siehe Abbildung 5.2) für alle untersuchten GOBO-Räder.

Diese Ergebnisse zeigen, dass das SNR bei gepulster Projektion etwas höher ist als bei kontinuierlicher. Der Grund ist, dass die Projektions-LED im gepulsten

Modus 9 % mehr Licht abgibt, als im kontinuierlichen (gemessen am Vergleich der gepulsten und kontinuierlichen Messreihe mit GOBO R). Außerdem weisen die GOBOs L6, L7 und L8 eine verringerte Transparenz auf. Gegenüber GOBO R führt diese GOBO-Serie zu einer um 35 % reduzierten durchschnittlichen Helligkeit der Bilder (beide im kontinuierlichen Projektionsmodus gemessen). Dies erklärt den weiter verringerten SNR bei dieser Messserie (grün in Abbildung 5.7).

5.4 Messartefakte

Um die Qualität der Grobsuche zu beurteilen, wurde für jedes der sieben GOBO-Räder eine Messserie mit 512 konsekutiven Bildpaaren von jeder der beiden Messszenen aufgenommen. Bei der Bildeinzugsrate von 100 Hz betrug die Beleuchtungs- und Bildintegrationszeit 3,9 ms bzw. 9,8 ms. Die Rotationsgeschwindigkeit des Rads war jeweils auf die bei der Optimierung verwendete Vorgabe von 10,25 U/min bzw. 4,05 U/min eingestellt (siehe Tabelle 5.2). Mit dem nicht optimierten GOBO R wurden bei beiden Beleuchtungsbedingungen mit der gleichen Rotationsgeschwindigkeit wie bei den optimierten Rädern aufgenommen.

Zunächst wurde für jedes GOBO-Rad ein (für die Beurteilung der Grobsuchqualität) als wahr angenommenes Referenz-3D-Modell rein mittels NCC berechnet. Dafür wurden mit einer Bildsequenzlänge von $N = 50$ jeweils zehn 3D-Rekonstruktionen durchgeführt, und die zehn Modelle 3D-Punkt-weise gemittelt. Dabei wurden nur diejenigen 3D-Punkte akzeptiert, bei denen die dazugehörenden Disparitätswerte weniger als 1 px voneinander abwichen. In der kontrastreichen Messszene konnten in den dunkelsten Bildbereichen selbst mit dieser langen Bildsequenzlänge nicht mehr zuverlässig alle 3D-Punkte berechnet werden.

Für jede Messserie wurden dann für die Sequenzlängen $N = 6, 7$ und 8 jeweils 100 3D-Rekonstruktionen durchgeführt, beginnend ab Bildnummer 0, 5, 10, 15 usw.. Die resultierenden 100 3D-Punktvolken wurden mit dem Referenz-3D-Modell verglichen. Die Anzahl der 3D-Punkte, welche bei den Einzelmessungen gegenüber der Referenz fehlen oder mehr als 10 cm abweichen und deshalb als Artefakt der Grobsuche angesehen werden müssen, wurde ermittelt.

Abbildung 5.8 zeigt für die Rekonstruktionsserien mit $N = 6$, wie sich die Messartefakte über die kontrastreiche Szene verteilen: In Bereichen, wo der Signal-zu-Rausch-Abstand (siehe Abbildung 5.6) niedrig ist, ist auch die Anzahl der Ausfälle höher. Außerdem ist auch klar erkennbar, dass sowohl die Muster-Optimierung allein (zweites Bild von links) als auch die gepulste Projektion allein (drittes Bild von links) eine erhebliche Verbesserung gegenüber dem unoptimierten Muster bei kontinuierlicher Projektion haben (ganz links). Quasi artefaktfrei ist allerdings erst das Ergebnis, welches sowohl ein optimiertes Muster als auch die gepulste Projektion einsetzt (ganz rechts). 3D-Modelle für den ersten und letzten Fall werden in Abbildung 5.9 gezeigt.

Eine statistische Auswertung der Häufigkeit der grob falschen oder fehlenden 3D-Punkte für alle Sequenzlängen ist den Abbildungen 5.10 und 5.11 dargestellt. Die Abbildungen haben eine logarithmische Skala. Bei sehr niedrigen Ausfallwerten von unter 0,1 % kann das 3D-Modell als vollständig betrachtet werden. Die Box-Statistiken zeigen: Median (schwarze Linien), 50%-Konfidenzintervall (farbige Boxen) sowie Minimum und Maximum (farbige Linien). Die gezeigten

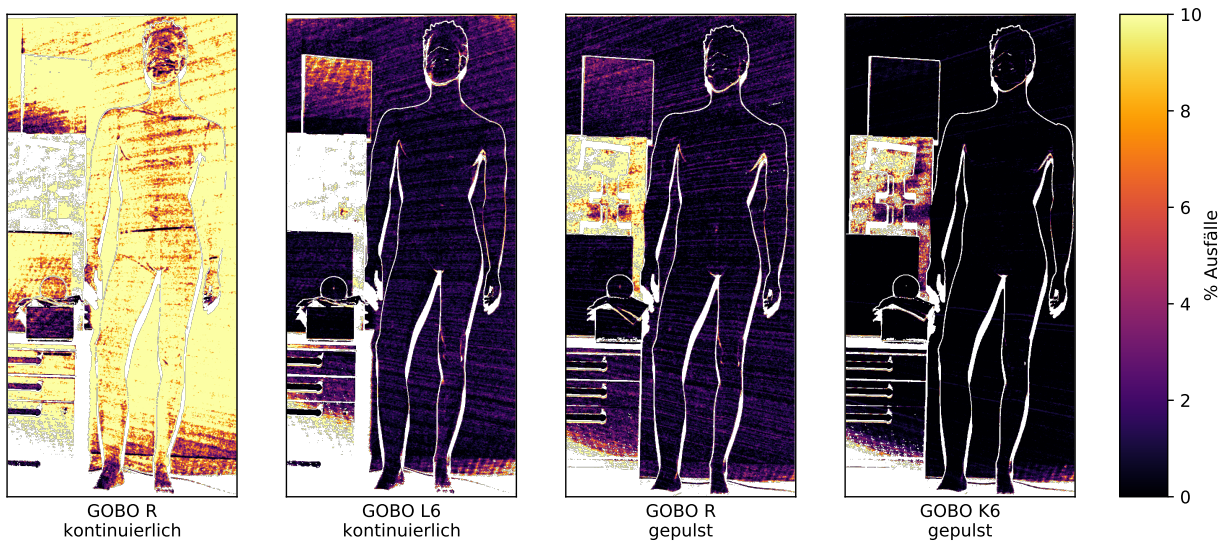


Abb. 5.8: Prozentuale Anzahl der Messartefakte für Bildsequenzlänge $N = 6$ für vier verschiedene Projektionsvarianten. Das mit Abstand beste Ergebnis wird mit einer Kombination aus gepulster Projektion und optimiertem Muster erreicht (ganz rechts).

Statistiken wurden mit den Daten der hellen Messszene erstellt, da in der kontrastreichen Szene die Artefakte der sehr dunklen Bereiche dominieren.

Die Ergebnisse zeigen, dass die 3D-Rekonstruktion mittels des schnellen BiCoS-Algorithmus (Abbildung 5.10) für die Sequenzlängen $N = 6$ und $N = 7$ bei kontinuierlicher Projektion und ohne Musteroptimierung keine praktisch verwendbaren Ergebnisse liefert, da ca. 20% bzw. ca. 10% der 3D-Punkte ausfallen oder grob falsch sind. Die 3D-Ergebnisse werden aber sowohl durch die gepulste Projektion, als auch durch die optimierten Muster um jeweils etwa eine Größenordnung besser. Für die Sequenzlängen $N = 6$ und $N = 7$ sind beide Verbesserungen erforderlich, um weniger als 0,1% Ausfälle zu erreichen. Für die Sequenzlänge $N = 8$ reicht dagegen eine der beiden Verbesserungen aus.

Die Rekonstruktion mittels des rein NCC-basierten Referenzalgorithmus (Abbildung 5.11) produziert generell weniger Artefakte. Für die Sequenzlänge $N = 6$ lohnt es sich dennoch, zumindest eine der beiden Verbesserungen, also entweder ein optimiertes Muster oder die gepulste Projektion einzusetzen. Bei Sequenzlänge $N = 8$ sind schließlich alle Ergebnisse so gut, dass sowohl auf gepulste Projektion als auch auf optimierte Muster verzichtet werden kann.

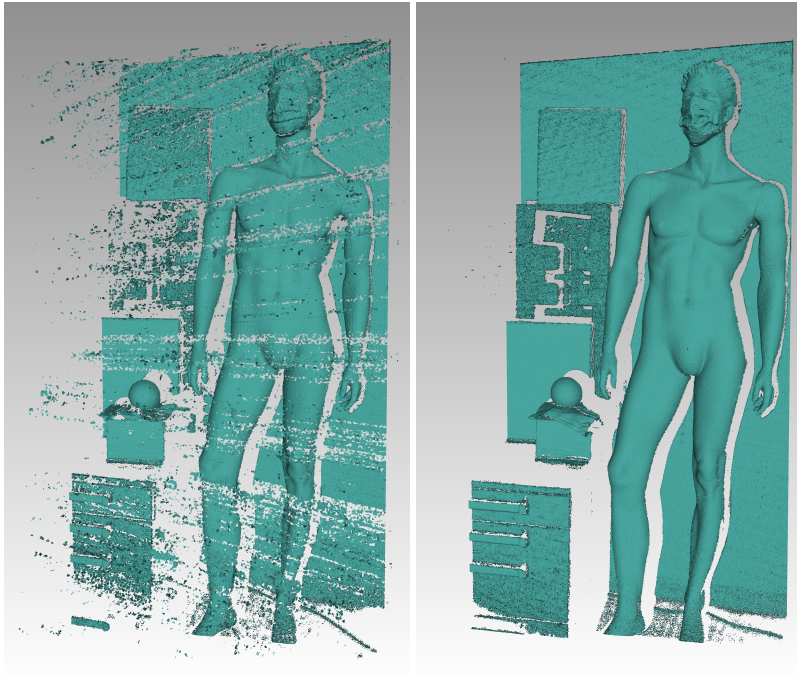


Abb. 5.9: 3D-Modelle erzeugt mit dem BiCoS-Algorithmus bei Sequenzlänge $N = 6$ mit dem unoptimierten *GOBO R* und kontinuierlicher Projektion (links) sowie dem optimierten *GOBO K6* und gepulster Projektion (rechts).

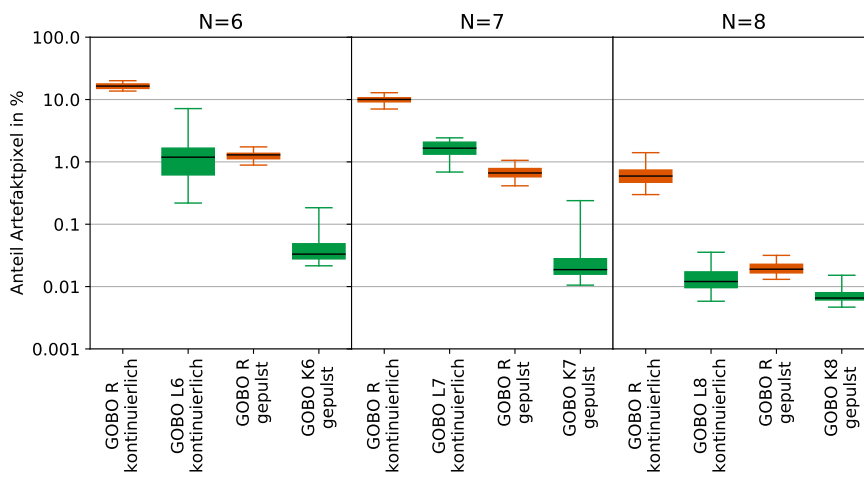


Abb. 5.10: Prozentuale Anzahl der Messartefakte (logarithmische Skala) für die Sequenzlängen $N = 6, 7, 8$ mit BF-Ähnlichkeits-Grobsuche für verschiedene Projektionsvarianten. Orange: nicht optimiertes GOBO, grün: optimiertes GOBO.

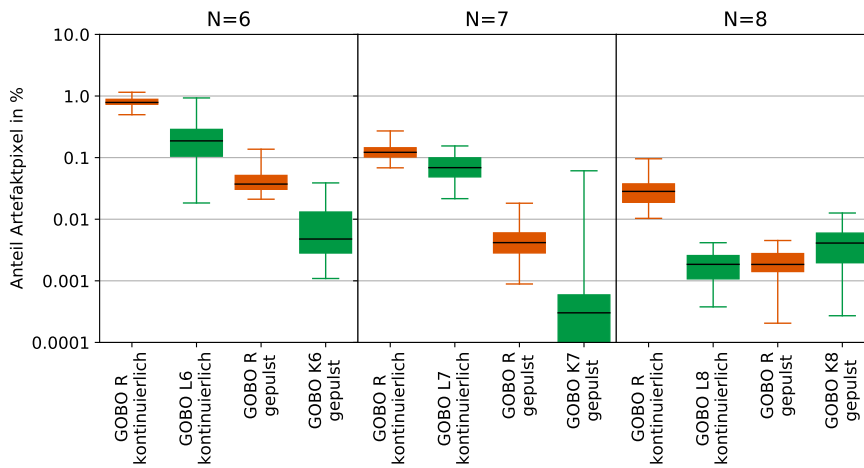


Abb. 5.11: Prozentuale Anzahl der Messartefakte (logarithmische Skala) für die Sequenzlängen $N = 6, 7, 8$ mit NCC-Grobsuche (Referenzalgorithmus) für verschiedene Projektionsvarianten. Orange: nicht optimiertes GOBO, grün: optimiertes GOBO.

5.5 Einfluss der optimierten Muster auf die Messgenauigkeit

In der hellen Messszene (Abbildung 5.2) befindet sich ein Ebenennormal. In den Messdaten wurde jeweils eine Ebene an die 3D-Punkte, die dort gemessen wurden, angefittet und die Standardabweichung der Punkte von dieser gefitteten Ebene bestimmt. Ausreißer, die mehr als 4 mm von der Ebene abweichen, wurden nicht berücksichtigt. Dies ermöglichte auch eine Bestimmung der Ebenenabweichung in den Fällen, in denen viele Messartefakte vorhanden waren (insbesondere bei Sequenzlänge 6 und kontinuierlicher Projektion). Es wird hier nicht mehr nach verwendetem Grobsuchalgorithmus unterschieden, weil beide Algorithmen die NCC für die Feinsuche verwenden und deshalb für einen Punkt das gleiche Ergebnis liefern, falls ein korrektes Grobsuchergebnis vorliegt.

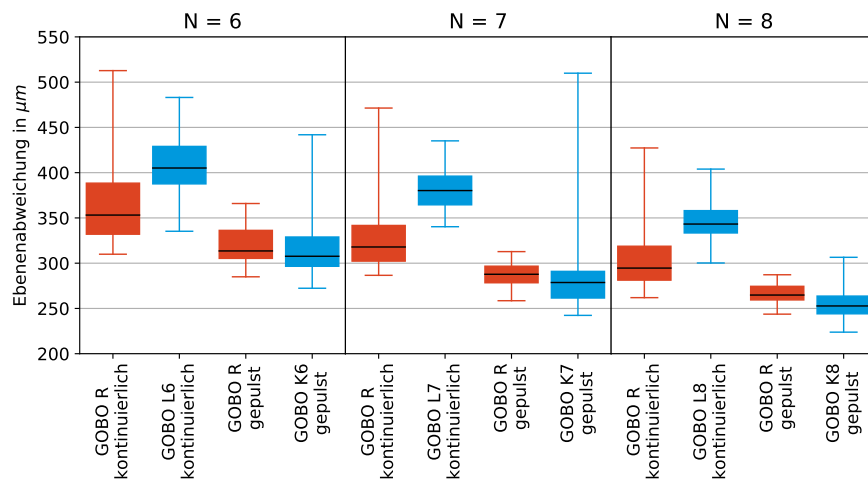


Abb. 5.12: Ebenenabweichung in μm für die Sequenzlängen 6,7,8. Rot: nicht optimiertes GOBO, blau: optimierte GOBOs (Messfeldgröße: $2\text{ m} \times 1\text{ m}$, siehe Abbildung 5.2).

Die Ergebnisse (Abbildung 5.12) zeigen, dass längere Bildsequenzlängen zu geringerer Ebenenabweichung führen. Dies lässt sich damit erklären, dass mehr Bilder auch mehr Information enthalten bzw. das Gesamt-Signal-zu-Rausch-Verhältnis besser wird, je mehr Bilder zur Rekonstruktion genutzt werden.

Abgesehen davon korrelieren die Ebenenabweichungen negativ mit dem ermittelten Signal-zu-Rausch-Verhältnissen auf dem Ebenennormal (siehe Abbildung 5.7): Die Messungen mit gepulster Projektion haben ein SNR von ca. 19 und weisen die geringste Ebenenabweichung auf. Darauf folgt GOBO R im kontinuierlichen Modus (SNR: 16,8) und die optimierten GOBOs bei kontinuierlicher Projektion (SNR: ca. 14,5).

Für den gepulsten Messmodus, bei dem alle getesteten GOBOs ähnliche SNR produzieren, kann damit festgestellt werden, dass die Optimierung des Musters auf Eindeutigkeit weder eine Verschlechterung, noch eine Verbesserung der Messgenauigkeit zur Folge hat. Für den kontinuierlichen Modus können nur starke Verschlechterungen oder Verbesserungen ausgeschlossen werden, weil die Auswirkung der unterschiedlichen SNRs dominiert.

5.6 Rechenzeit und Latenz

Aus der Bildsequenzlänge ergibt sich direkt die Dauer der Bildaufnahme: Bei 100 Hz Bildeinzugsrate folgt z.B. bei Sequenzlänge $N = 6$ eine Bildeinzugsdauer von knapp 60 ms. Wobei hier der gepulste Messmodus einen leichten Vorteil bietet, da dabei die Belichtungszeit kürzer ist (3,9 ms anstatt 9,8 ms), wodurch die Aufnahme des letzten Bildes 4,9 ms früher abgeschlossen ist. Zusammen mit den in Abschnitt 3.7 dargestellten Rechenzeiten ergibt sich daraus eine minimale Messlatenz von 77 ms bei Sequenzlänge $N = 6$ unter Verwendung des BiCoS-Algorithmus. Abbildung 5.13 zeigt die erzielbaren Messlatenzen im Detail.

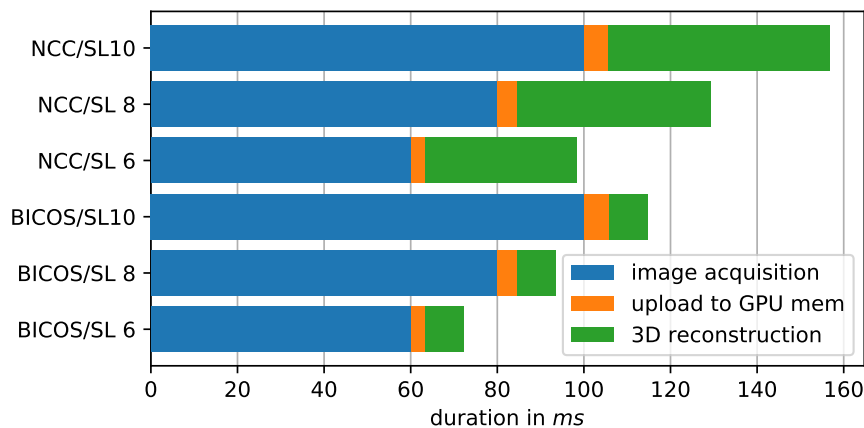


Abb. 5.13: Messlatenz für verschiedene Bildsequenzlängen aufgeschlüsselt nach Bildaufnahme (blau), Upload der Daten in GPU-Speicher (orange) und 3D-Rekonstruktion (grün). Jeweils einmal für Korrespondenz-Grobsuche mittels Kreuzkorrelation und einmal mittels BF-Ähnlichkeit [56].

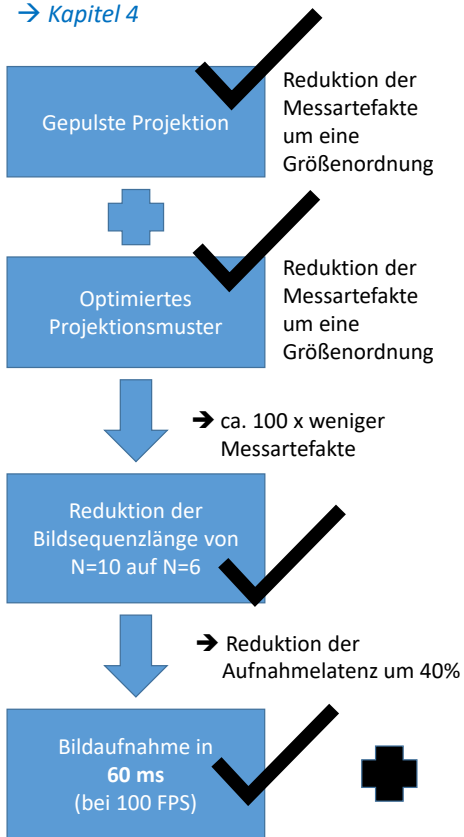
5.7 Erreichte Verbesserungen

Bezogen auf die in Abbildung 1.8 vorgestellten Kernthesen der Arbeit wurden zusätzlich zu den bereits in Kapitel 3 beschriebenen Verbesserungen die folgenden Thesen belegt (Abbildung 5.14):

- ▶ Mittels gepulster Musterprojektion kann bei Sequenzlänge $N = 6$ eine Reduktion der Messartefakte um eine Größenordnung erreicht werden.
- ▶ Durch die Optimierung der Projektionsmuster kann eine Reduktion der Messartefakte um eine weitere Größenordnung erreicht werden.
- ▶ Die Anwendung der gepulsten Projektion zusammen mit den optimierten Mustern führt zu einer Reduktion der Messartefakte um insgesamt zwei Größenordnungen. Sie ermöglicht damit die Nutzung der Bildsequenzlänge auf $N = 6$ bei vollständigen 3D-Modellen. Bei 100 Hz Bildeinzugsrate entspricht dies einer Aufnahmelatenz von 60 ms.
- ▶ Zusammengefasst ermöglichen diese Verbesserungen eine Messlatenz von 77 ms bei moderater Bildeinzugsrate von 100 Hz.

Verkürzung der Aufnahmedauer

→ Kapitel 4



Beschleunigung der 3D-Rekonstruktion

→ Kapitel 3

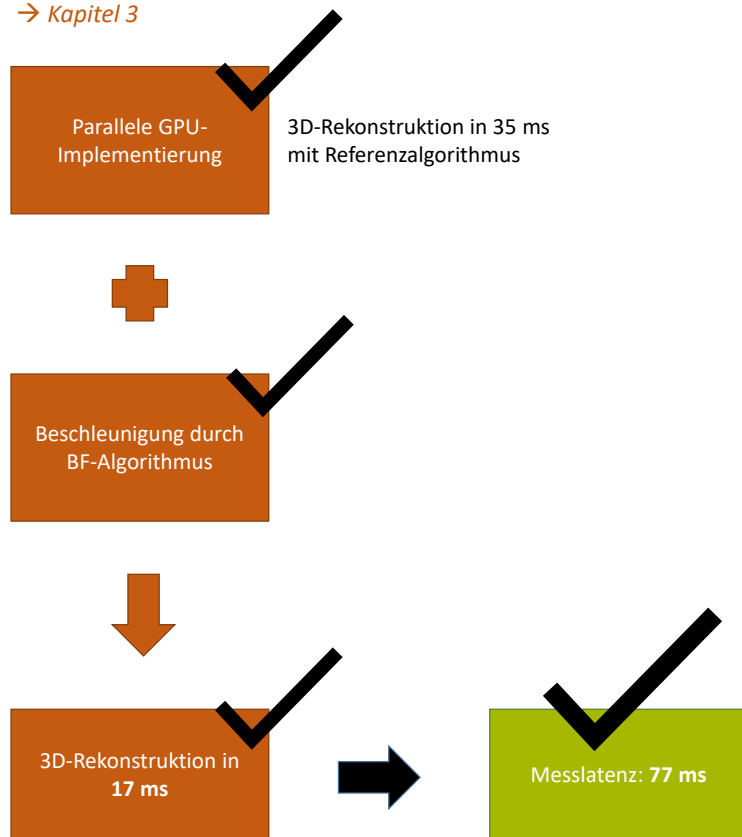


Abb. 5.14: Erreichte Verbesserungen

6.1 Einleitung

In diesem Kapitel werden Anwendungen für latenzarme 3D-Sensorik vorgestellt.

In Abschnitt 6.2 wird eine medizinische Anwendung im Fachbereich HNO vorgestellt. Hier wird ein GOBO-Projektor-basierter 3D-Sensor eingesetzt, um Krankheitssymptome bei Gesichtslähmung zu quantifizieren. Außerdem sollen verbesserte Therapiekonzepte mit direkter Rückmeldung an den Patienten entwickelt werden.

Abschnitt 6.3 zeigt ein aus drei GOBO-Projektor-basierten 3D-Sensoren bestehendes Sensornetzwerk, das zur Überwachung von Patienten während der Strahlentherapie eingesetzt wird.

In Abschnitt 6.4 werden drei weitere Anwendungen kurz vorgestellt, die zwar keine konkreten Latenzanforderungen haben, die aber trotzdem wesentlich von den während der Arbeit an dieser Dissertation gemachten Entwicklungen profitiert haben.

6.2 Sensor für medizinische Untersuchungen der Gesichtsmuskulatur und -nerven

6.2.1 Motivation: Krankheitsbild der Facialisparesie

Die Facialisparesie^[81] (Gesichtslähmung) ist eine Erkrankung, bei der der Nervus Facialis (Gesichtsnerv) geschädigt ist.

Die Nervenschädigung führt zur Lähmung der Gesichtsmuskulatur, was über die Dauer der Erkrankung auch oft zu deren Abbau führt, so dass die Symptome der Krankheit noch lange über die Heilung des Nervs erhalten bleiben.

Häufige Symptome sind:

- ▶ Starke Beeinträchtigung der Mimik und in schlimmen Fällen der Artikulation, was meist zu einer veränderten sozialen Wahrnehmung der erkrankten Person und mittelbar häufig zu Depressionen führen kann.
- ▶ Trockene Augen und daraus folgende Augenerkrankungen, weil sich das Augenlid nicht korrekt schließt.
- ▶ Probleme bei der Nahrungsaufnahme

Bei der Heilung des Gesichtsnervs nach Verletzungen infolge Tumor-Operationen oder nach Unfallverletzungen, kann es zu sogenannten *Synkinesien* kommen, d.h. dass Gesichtsmuskeln in unterschiedlichen Gesichtsbereichen nicht mehr unabhängig voneinander bewegt werden können. Beispiele hierfür sind:

- ▶ Der Mundwinkel bewegt sich mit, wenn das Augenlid geschlossen werden soll.

- ▶ Es kommt zum Austritt von Tränenflüssigkeit, wenn der Mund bewegt wird („Krokodilstränen“).
- ▶ Bei Augenbewegung rümpft sich die Nase.

Um die Symptome der Facialisparesie für den Patienten zu minimieren, werden verschiedene Behandlungsmethoden erforscht. Ein Hauptproblem ist hierbei, den Behandlungserfolg objektiv zu bewerten. Für eine solche Bewertung ist es notwendig, die Schwere der Symptome zu quantifizieren. Ein 3D-Sensor zur Messung der Mimik kann hierfür die notwendigen Daten liefern.

6.2.2 3D-Sensor zur Erfassung des Gesichts

Am Fraunhofer IOF wurde in Zusammenarbeit mit der Abteilung HNO des Universitätsklinikums Jena zu diesem Zweck ein 3D-Sensor entwickelt (siehe Abbildung 6.1). Dessen Messfeld wurde auf die Aufnahme von Gesichtern in einem Arbeitsabstand von 1,5 m optimiert. Damit die Patienten nicht geblendet und irritiert werden, arbeitet die GOBO-Projektionseinheit mit nicht sichtbarem Nah-Infrarot-Licht. Störendes Licht im für Menschen sichtbaren Wellenlängenbereich wird mittels Langpassfiltern zwischen Objektiv und Sensor blockiert.

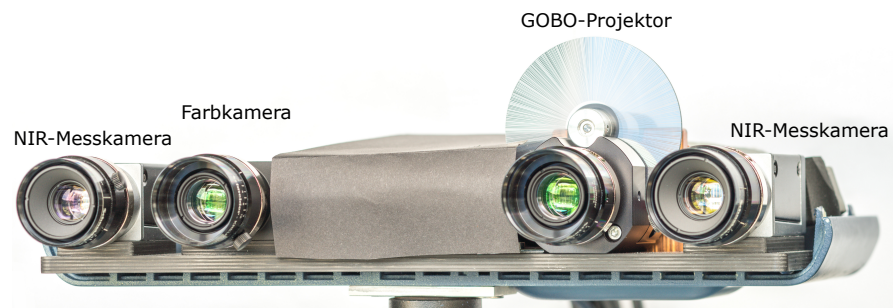


Abb. 6.1: Geöffneter 3D-Sensor für die Erfassung menschlicher Gesichter und deren Mimik.

Die beiden Messkameras arbeiten mit einer Auflösung von 1 MPx bei einer Bildeinzugsrate von 360 Hz. Die in Kapitel 4 vorgeschlagenen Optimierungen des Projektors wurden in der aktuellen Version noch nicht umgesetzt, eine Umsetzung in einer zukünftigen Version ist aber geplant. Die Messungen werden deshalb mit einer Sequenzlänge von $N = 10$ Bildpaaren durchgeführt, was reine Aufnahmelatenzen von 28 ms ermöglicht.

Für die 3D-Rekonstruktion kommt der in Kapitel 3 beschriebene BiCoS-Algorithmus zum Einsatz. Der Rekonstruktionsrechner ist mit einer NVIDIA GeForce 1080 Grafikkarte ausgestattet, auf der die 3D-Punktwolke in 26 ms berechnet wird. Dies führt zu einer Messlatenz von 54 ms.

Abbildung 6.2 zeigt drei mit dem Sensor aufgenommene 3D-Gesichtsmodelle.

Zusätzlich zu den beiden Messkameras hat der Sensor eine kalibrierte Farbkamera. Damit kann entweder Farbinformation auf die resultierenden 3D-Punkte aufgebracht werden, eine Tiefenkarte zum Farbbild erstellt werden oder für einzelne, im Farbbild identifizierte Punkte die 3D-Koordinaten bestimmt werden. Die Farbkamera ermöglicht den Einsatz bekannter und gut erforschter Algorithmen zur Extraktion von Gesichtsmerkmalen, denen dann 3D-Koordinaten zugeordnet werden können.

Tabelle 6.1 listet weitere Kenngrößen des Sensors auf.

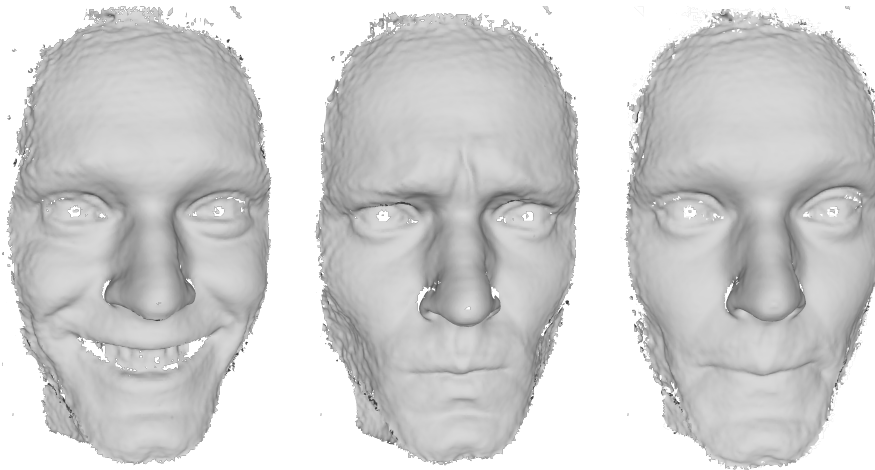


Abb. 6.2: Mit dem Sensor gemessene 3D-Modelle eines gesunden Probanden mit verschiedenen Gesichtsausdrücken.

Sensorgröße	300 × 190 × 100 mm ³
Messfeldgröße	400 mm × 400 mm
Messabstand	1,5 m
2D-Bildauflösung	1 MPx
2D-Bildeinzugsrate	360 Hz
3D-Rate (maximal)	36 Hz
Wellenlänge des Projektionslichts	850 nm (NIR)
GOBO-Durchmesser	60 mm
Durchleuchtete GOBO-Fläche	10 mm × 10 mm

Tab. 6.1: Eigenschaften des 3D-Sensors für die Gesichtsmessung.

6.2.3 Bestimmung von 3D-Landmarken

Konkret werden im Bild der Farbkamera automatisiert 68 Landmarken-Punkte im Gesicht des Patienten nach dem MULTI-PIE-Schema bestimmt [82, 83]. Nach diesem Schritt liegen die Landmarken-Punkte als Pixelkoordinaten im Farbbild vor. Anschließend wird zu jedem dieser Punkte die zugehörige 3D-Koordinate aus dem 3D-Datensatz ermittelt, d.h. dann ist jeder Landmarkenpunkt metrisch bestimmt. In Abbildung 6.3 ist dies beispielhaft dargestellt.

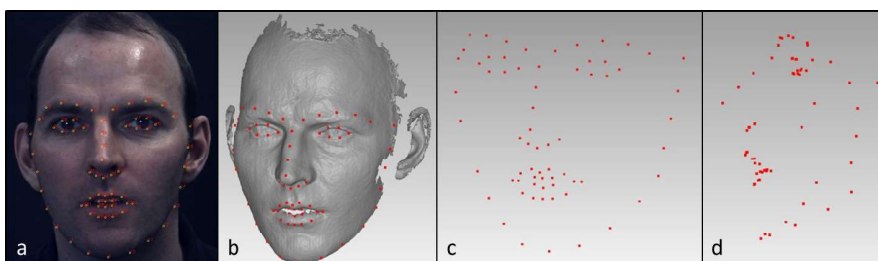


Abb. 6.3: a) Farbbild mit erkannten Landmarken b) Landmarken auf 3D-Modell projiziert c+d) Landmarken in zwei 3D-Ansichten [79]

Anhand dieser Landmarken werden verschiedene Metriken untersucht, um die Gesichtsbewegung zu quantifizieren. Z.B. wird die Fläche zwischen den 6 Augen-Landmarken bestimmt, indem die Augenfläche zwischen den Punkten in 4 Dreiecke aufgeteilt wird und deren Fläche addiert wird (siehe Abbildung 6.4). Um eine gute Datenerfassung für diese Metrik zu gewährleisten, wird der 3D-Sensor vor dem Patienten auf dem Boden positioniert und erfasst dessen Gesicht von schräg unten (siehe Abbildung 6.5). Dadurch werden Löcher in der 3D-Punktwolke durch Schatten vom Augenbrauenbereich vermieden.

Ein anderes Beispiel für eine untersuchte Metrik ist, wie weit sich die Mundwinkel bewegen. Für diese Metrik ist die 3D-Information besonders wichtig, weil

Abb. 6.4: Augenflächenberechnung basierend auf Basis von Dreiecken zwischen den 3D-Koordinaten der Augen-Landmarken. Mit freundlicher Genehmigung von Ellen Schötz/Universitätsklinikum Jena. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.



Abb. 6.5: Untersuchung der mimischen Muskulatur mittels 3D Sensor. Der Patient (links) wird während der Behandlung von unten mit dem 3D-Sensor (rechts unten) beobachtet. Mit freundlicher Genehmigung von PD Dr. med. habil. Gerd Fabian Volk/Universitätsklinikum Jena. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.



nicht nur eine laterale, sondern auch eine zum Ohr, d.h. dorsal (nach hinten) gerichtete Bewegung stattfindet.

Durch Auswertung mehrerer dieser Metriken ist es auch möglich, das Ausmaß der Synkinesie zu quantifizieren. Z.B. kann die Weite der Mundwinkelbewegung ausgewertet werden, wenn der Patient die Augen zukneifen soll.

6.2.4 Behandlung mittels Elektrostimulation

Am Universitätsklinikum Jena wird unter anderem folgende Behandlungsmethode wissenschaftlich evaluiert: Um der häufig auftretenden Rückbildung der Gesichtsmuskulatur entgegenzuwirken, werden unterschiedliche Bereiche des Gesichtsnervs mittels elektrischem Strom angeregt. Diese Elektrostimulation führt zur Kontraktion der Gesichtsmuskulatur, was diese trainiert.

Während jeder Behandlung werden Gesichtsmodelle erzeugt und anschließend ausgewertet. So kann die individuelle Reaktion der Muskeln auf die Stimulation quantifiziert und über mehrere Wochen protokolliert werden. Ein mögliches Ziel ist es, anhand der quantifizierten Reaktion in einer Regelungsschleife die Stromstärke anzupassen.

6.2.5 Interaktives Trainingssystem

Im Rahmen der Bachelor-Arbeit von Henrik Voigt [84] wurde mit dem Sensor der Prototyp eines interaktiven Trainingssystems für Facialisparesie-Patienten entwickelt (siehe Abbildung 6.5). Das Trainingssystem führt den Patienten anhand mehrerer Erklärungsvideos durch verschiedene Übungen, die die Beweglichkeit der Gesichtsmuskulatur fördern sollen. Z.B. soll der Patient die Stirn runzeln, die Augen locker oder fest schließen, die Nase rümpfen, mit geschlossenen oder offenen Lippen lächeln, die Lippen spitzen, die Backen aufblasen, die Zähne fletschen oder bestimmte Sätze artikulieren. Bei jeder Übung wird das Gesicht des Patienten mit dem 3D-Sensor aufgenommen.

Das Trainingssystem soll so weiterentwickelt werden, dass es in Zukunft ohne die Hilfe Dritter vom Patienten auch zuhause eingesetzt werden kann. Dazu ist es wünschenswert, ein direktes Feedback anhand verschiedener Metriken (siehe oben) an den Patienten auszugeben.

6.2.6 Fazit und Ausblick

Der für diese Anwendung entwickelte Sensor erreicht durch die schnelle, parallelisierte GPU-Implementierung des BiCoS-Algorithmus eine 3D-Rekonstruktionslatenz von unter 26 ms, womit in der aktuellen Entwicklungsversion eine Messlatenz von 54 ms erreicht wird.

Die Zusammenarbeit mit dem Universitätsklinikum Jena hat gezeigt, dass die genaue und schnelle 3D-Vermessung von Gesichtern die Möglichkeiten, Krankheitssymptome zu quantifizieren, wie sie bei der Facialisparesie auftreten, stark erweitert. Dadurch werden viele Therapien, wie z.B. die Elektrostimulation, erstmals in ihrem Ergebnis quantitativ messbar und damit wissenschaftlich vergleichbar. Die Möglichkeit, mit niedriger Latenz auf die Messungen zu reagieren, eröffnet den Weg zu neuen Therapieansätzen, bei denen entweder die Maschine auf die menschliche Reaktion reagiert (z.B. Anpassung der Stromstärke bei der Elektrostimulation) oder der Mensch ein qualitativ hochwertiges Feedback bekommt, das ihm hilft, seine eigene Anstrengung beim Training in ihrer Effektivität zu bewerten und gegebenenfalls in eine andere Richtung zu lenken.

In Zukunft soll der Projektor mittels der in Kapitel 4 vorgeschlagenen gepulsten Projektion und einem optimierten GOBO-Muster verbessert werden. Die in Kapitel 5 gezeigten Ergebnisse lassen damit eine Reduktion der Aufnahmesequenzlänge von 10 auf 6 Bilder bzw. der Aufnahmelatenz von 28 ms auf 17 ms erwarten. Zudem könnte ein Update der Grafikkarte auf die in den Experimenten in Kapitel 5 eingesetzte NVIDIA GeForce GTX 2080 GPU eine weitere Verbesserung der Rekonstruktionslatenz bringen, was eine 3D-Rekonstruktionslatenz von < 17 ms erwarten lässt, da der Sensor mit 1 MPx anstatt mit 2 MPx arbeitet, wie der in Kapitel 5 beschriebene. Deshalb erscheint perspektivisch eine Messlatenz von unter 34 ms möglich. In Abbildung 6.6 ist dies nochmals mit Bezug auf die in Abschnitt 1.4 aufgestellten Kernthesen dargestellt.

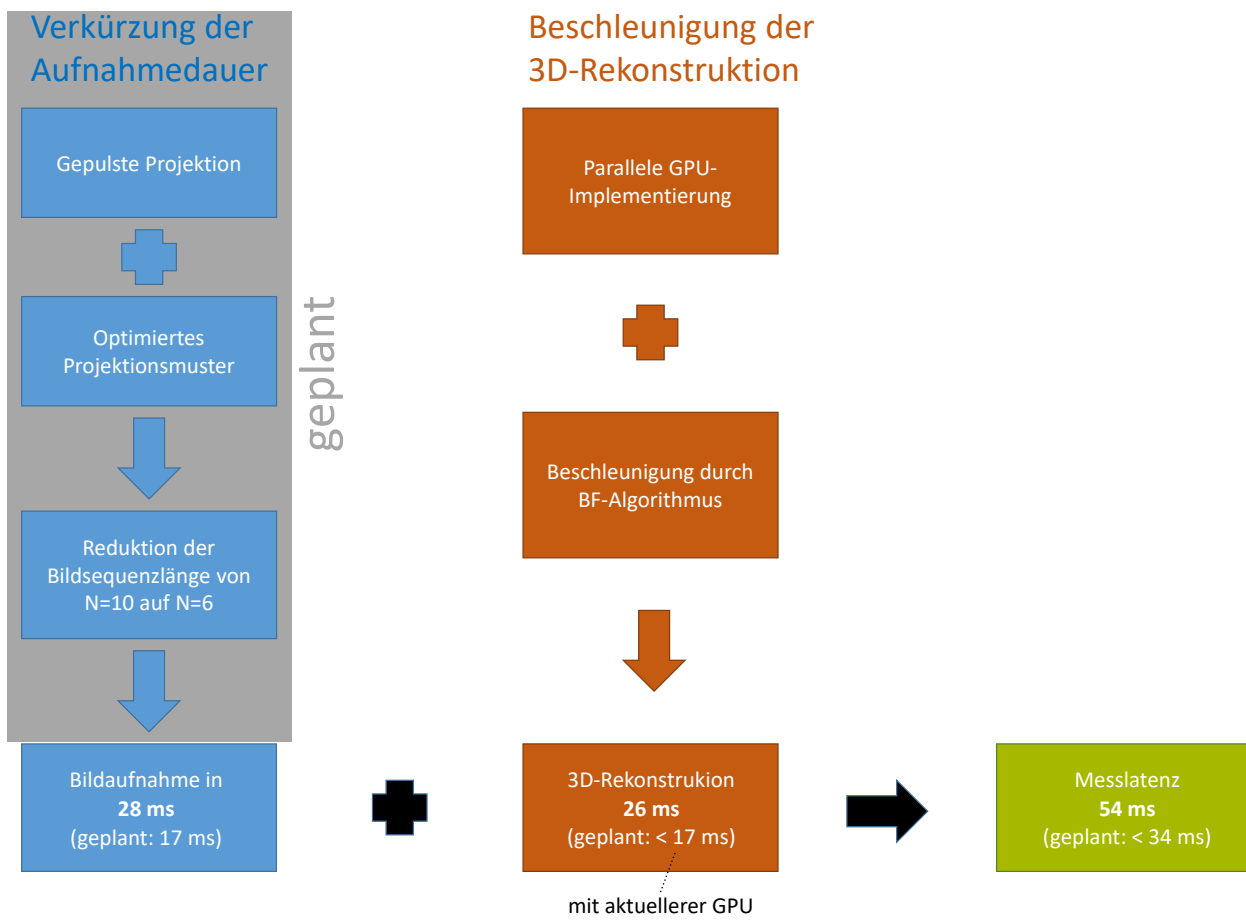


Abb. 6.6: Grafische Darstellung der erreichten Verbesserungen mit Bezug auf die in Abschnitt 1.4 aufgestellten Kernthesen.

6.3 Sensornetzwerk zur Patientenüberwachung während der Bestrahlungstherapie

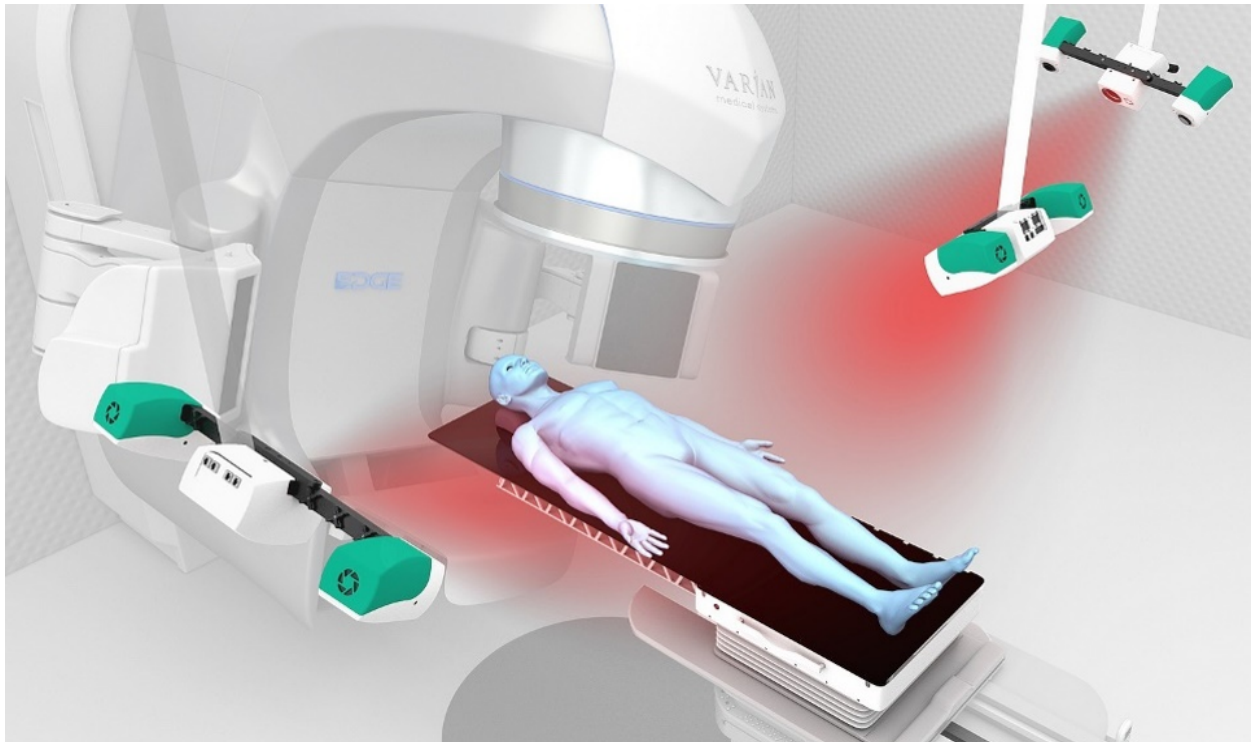


Abb. 6.7: Patientenüberwachung bei der Strahlentherapie mittels dreier 3D-Sensoren. Mit Genehmigung der Firma Varian Medical Systems und des Fraunhofer IOF. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.

6.3.1 Motivation

Die medizinische Strahlentherapie wird in der Krebstherapie eingesetzt, um Tumorgewebe mittels radioaktiver Strahlung zu schädigen. Ziel ist es dabei, dass der Tumor nicht weiterwächst und letztendlich möglichst komplett absterbt. Dabei ist es erforderlich, die Bestrahlung genau auszurichten, um zu gewährleisten, dass der Tumor bestrahlt wird, ohne dass auch das umliegende Gewebe stark geschädigt wird.

Bewegt sich der Patient während der Therapie, besteht die Möglichkeit, dass das Bestrahlungszentrum nicht mehr im Tumor liegt, was nicht nur die gewünschte Wirkung vereitelt, sondern auch zu gefährlichen Schäden außerhalb der Tumor-Region führen kann. Je nach Bestrahlungsort im Körper des Patienten besteht die Möglichkeit, das betroffene Körperteil während der Therapie zu fixieren, was allerdings als sehr unangenehm empfunden werden kann. Andere Körperteile lassen sich nicht fixieren. Insbesondere im Thoraxbereich ist dies wegen der Atmung nicht möglich. Da die Strahlentherapie häufig bei Brustkrebs-Patienten eingesetzt wird, gibt es hier einen erheblichen Verbesserungsbedarf. Bisher war es notwendig, dass der Patient während kurzer Bestrahlungsintervalle die Luft anhält, damit in dieser Zeit keine Bewegung des Brustkorbs stattfindet.

Der Einsatz latenzarmer, genauer 3D-Sensorik kann hierbei einen erheblichen Vorteil bringen: Eine Bewegung des Patienten kann latenzarm erkannt werden, so dass die Bestrahlung gestoppt oder das Bestrahlungsgerät nachgeführt

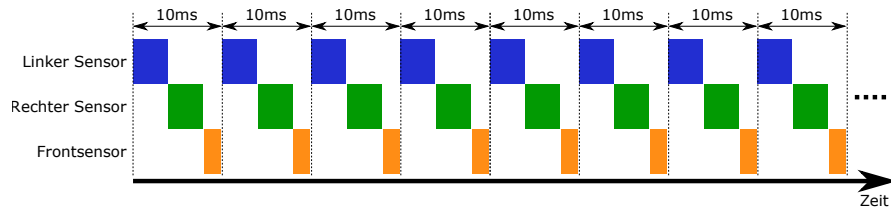
werden kann. Das Luftanhalten ist nicht mehr erforderlich, der Patient kann normal weiteratmen. Die Bestrahlung kann dann z.B. automatisiert immer am Hoch- oder Tiefpunkt der Brustbewegung durchgeführt werden.

6.3.2 Sensornetzwerk

In Zusammenarbeit mit der Firma Varian Medical Systems wurde am Fraunhofer IOF ein Sensornetzwerk zur Überwachung von Patienten während der Strahlentherapie entwickelt. Das Netzwerk besteht aus drei GOBO-Projektorbasierten 3D-Sensoren, die den Patienten gemeinsam ganzkörperlich erfassen. Abbildung 6.7 zeigt ein Rendering des Sensornetzwerks zusammen mit der Bestrahlungsanlage und dem Patienten. Das Messfeld umfasst die gesamte Patientenliege und hat eine Größe von 2 m × 1 m.

Das Sensornetzwerk[85] besteht aus drei Einzelsensoren[86] ähnlichen Aufbaus. Zwei dieser Sensoren beobachten den Patienten jeweils von der linken und rechten Seite der Patientenliege aus. Diese beiden Seitensensoren sind baugleich mit einem Kameraabstand von jeweils 60 cm. Der dritte Sensor (Frontsensor) beobachtet die Liege von der Fußseite aus. Er hat einen Abstand zwischen den Messkameras von 40 cm. Diese kürzere Basislänge ist ausreichend, da sich der Frontsensor näher an der Patientenliege befindet, deshalb kann etwa der gleiche Triangulationswinkel wie bei den Seitensensoren erreicht werden. Aus demselben Grund hat der Frontsensor eine gegenüber den Seitensensoren verkürzte Belichtungszeit, da auf kürzerer Entfernung weniger Projektionslicht verloren geht. Die Kameras des Frontsensors sind um 90° gedreht eingebaut, so dass sich ein Hochkant-Bildformat ergibt.

Abb. 6.8: Zeitlich verschachtelte Bildintegrations- und Beleuchtungszeiten der einzelnen Sensoren: Jeder der 3 Sensoren hat seinen eigenen Zeitschlitz, während dessen er Muster projiziert und seine Kameras Licht integrieren.



Sowohl die Projektion als auch die Kameras werden im gepulsten Modus betrieben (Abbildung 6.8). Die Kameras arbeiten mit einer Framerate von 100 Hz. Die Beleuchtungsintervalle der Projektoren sind so zueinander verschoben, dass jeder Sensor nur das Projektionslicht seines eigenen Projektors aufnimmt, nicht aber das der anderen, was zu Messfehlern führen würde. D.h. zunächst nimmt der linke Sensor ein Bildpaar auf, dann der Frontsensor und schließlich der rechte, danach beginnt die zweite Aufnahme des linken usw.. Die Seitensensoren beleuchten jeweils 3,9 ms, der Frontsensor 1,9 ms.

Tab. 6.2: Technische Daten der 3D-Sensoren

	Seitensensoren	Frontsensor
Messfeldgröße	2 m × 1 m	1 m × 2 m
Kameraabstand	60 cm	40 cm
Kameraauflösung für Messung	1920 × 960	960 × 1920
2D-Bildeinzugsrate	100 Hz	
Bildintegrationszeit	3,9 ms	1,9 ms
Wellenlänge des Projektionslichts	850 nm (NIR)	
GOBO-Durchmesser	60 mm	
Durchleuchtete GOBO-Fläche	20 mm × 10 mm	10 mm × 20 mm
3D-Rate	10 Hz	

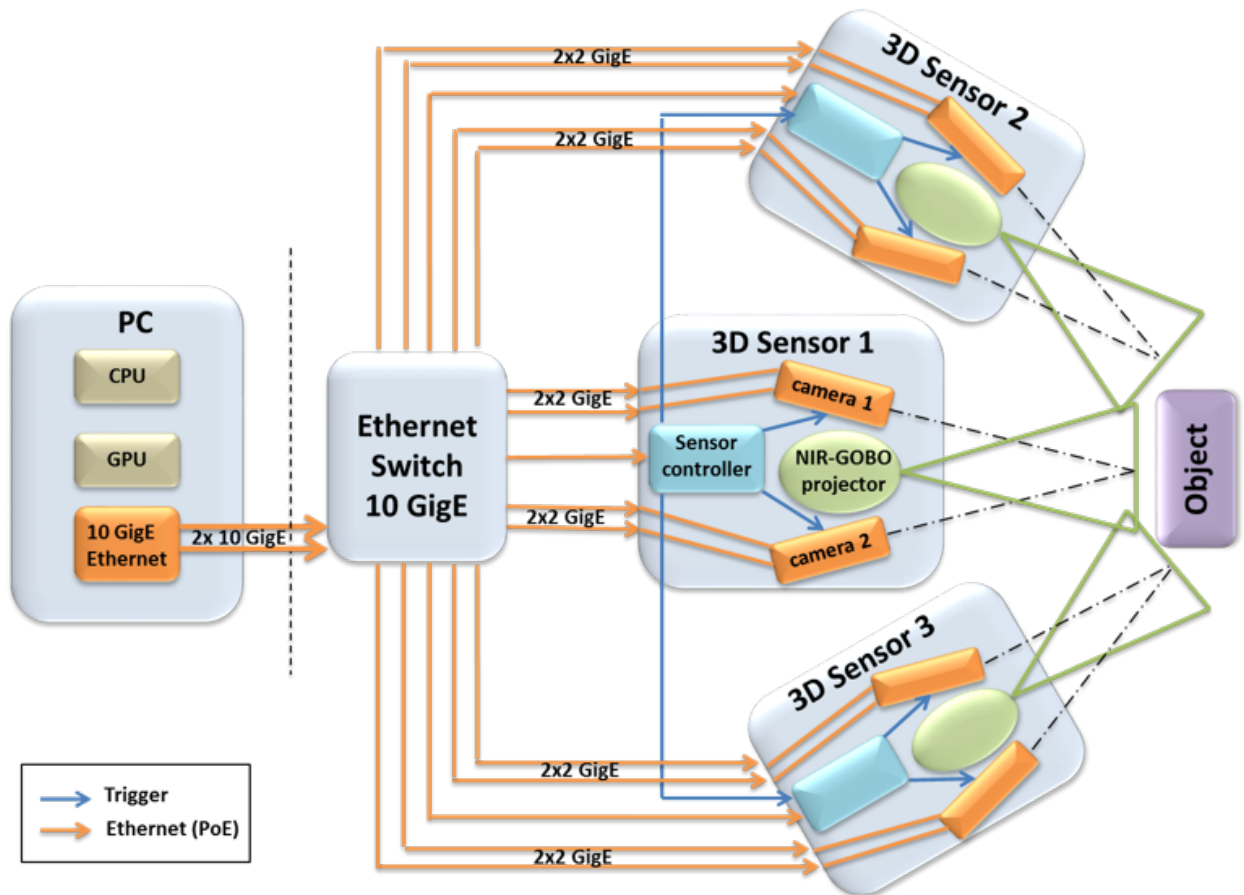


Abb. 6.9: Datenfluss innerhalb des Sensornetzwerks: Jede der sechs Kameras ist mit zwei GigE-Leitungen an einen gemeinsamen Netzwerk-Switch angeschlossen, der die Daten über zwei 10-GigE-Leitungen an den Rekonstruktions-PC weiterleitet. Außerdem werden die Sensoren mittels dreier Sensor-Controller, welche ebenfalls über Ethernet konfiguriert werden, gesteuert. Die Sensoren werden über eine zusätzliche Trigger-Leitung miteinander zeitlich synchronisiert. *Mit freundlicher Genehmigung des Fraunhofer IOF. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.*

Eigenschaften der Sensoren sind in Tabelle 6.2 aufgelistet.

Die Daten der 6-GigE-Kameras werden in einem PC zusammengeführt, wo drei 3D-Rekonstruktionsrechnungen simultan auf einer Grafikkarte ausgeführt werden. Die Sensoren synchronisieren sich untereinander mittels einer Hardware-Triggerleitung. In Abbildung 6.9 ist der Datenfluss nochmals grafisch dargestellt. Um auf dem PC die Bilder einander zeitlich korrekt zuordnen zu können, auch im seltenen Fall, dass ein Bild bei der Übertragung verloren geht, werden Zeitstempel der Kameras genutzt. Der Drift der kamerainternen Uhren wird regelmäßig auf Rechner-Seite protokolliert und jeweils ein Korrektursummand akkumuliert, so dass die Langzeitstabilität der Zuordnung gewährleistet ist.

6.3.3 Erreichte Verbesserungen

Die 3D-Rekonstruktionsrechnung findet auf einem gemeinsamen Rechner statt, der mit einer NVIDIA GeForce GTX 2080 GPU ausgestattet ist. Die 3D-Rekonstruktion wird mittels des BiCoS-Algorithmus durchgeführt. Die 3D-Rekonstruktion für alle drei Sensoren kann in 55 ms durchgeführt werden. Gegenüber dem Referenzalgorithmus wurde damit eine erhebliche Beschleunigung erreicht: Die Rekonstruktionslatenz läge ohne den BiCoS-Algorithmus

selbst mit der schnellen GPU-Implementierung bei deutlich über 100 ms.

Die Sensoren wurden in der ersten Konfiguration noch ohne optimierte Projektionsmuster mit einer Sequenzlänge von $N = 10$ betrieben. Damit lag die Aufnahmezeit bei 100 ms. Mit den neuen, für das System optimierten GOBO-Rädern kann in Verbindung mit der verwendeten gepulsten Projektion eine Sequenzlänge von $N = 6$ eingesetzt werden, womit sich die **Aufnahmelatenz auf 60 ms verkürzt**, womit eine **Messlatenz von insgesamt 115 ms** für das gesamte Sensornetzwerk erreicht wird.

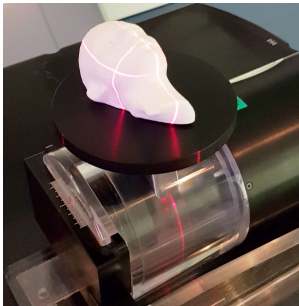


Abb. 6.10: Testaufbau mit dem der Projektpartner die Genauigkeit der Atembewegungserfassung testete: Ein Testkörper (hier Styroporkopf) wird von einem Aktuator definiert auf und ab bewegt. Das 3D-Sensornetzwerk misst die Szene. *Mit Genehmigung der Firma Varian Medical Systems. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.*

6.3.4 Atembewegungsüberwachung

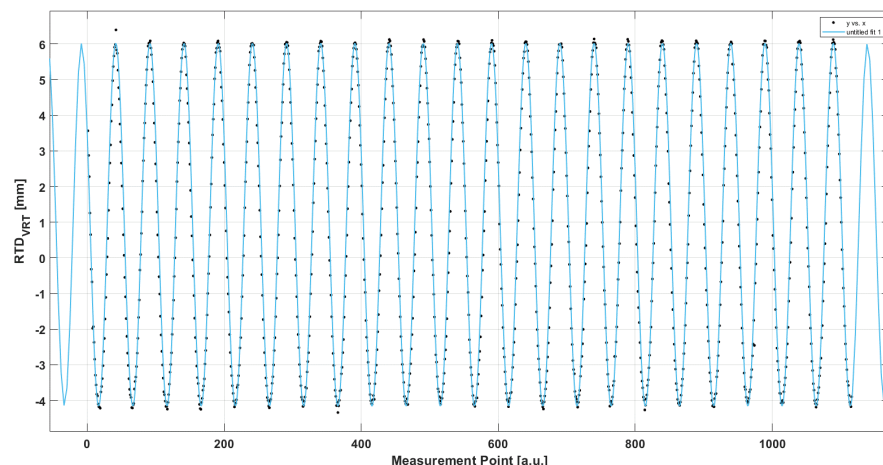
Der Projektpartner Varian Medical Systems führte zur Untersuchung der Messqualität für die Anwendung der Atembewegungsüberwachung folgende Messung durch: Die Atmung wurde simuliert, indem ein Testkörper von einem Aktuator mit definierter Geschwindigkeit auf und ab bewegt wurde (siehe Abbildung 6.10). Das Bewegungsprofil war dabei sinusförmig mit einer Periodenlänge von 6 s und einer Amplitude von 5 mm. Der Messaufbau wurde vom Sensornetzwerk beobachtet. Anschließend wurde jeweils einmal für jeden der drei Sensoren einzeln und einmal für den zusammengesetzten Datenstrom des Netzwerks eine Höhenzeitreihe der Oberfläche erstellt. An diese Zeitreihe wurde eine Sinusfunktion angefitet (siehe Abbildung 6.11). Die Ergebnisse sind in Tabelle 6.3 aufgelistet.

Die Ergebnisse zeigen, dass sich mit den Sensoren die Atmungsbewegung sowohl mit jedem einzelnen Sensor als auch mit dem Netzwerk gemeinsam mit einer Genauigkeit von $65 \mu\text{m}$ oder besser ermitteln lässt.

Tab. 6.3: Ergebnisse des Atemüberwachungstests: gemessene Amplituden, Perioden und die Qualität des Sinus-Fits R^2 . *Mit Genehmigung der Firma Varian Medical Systems. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.*

	erwartet	linker S.	Fronts.	rechter S.	S.-Netzwerk
Ampl. [mm]	5,00	5,032	5,060	5,043	5,065
Periode [s]	6,00	6,00	6,00	6,03	6,01
R^2		0,9945	0,9923	0,9945	0,9906

Abb. 6.11: Ergebnisse des Atemüberwachungstests: vom Sensornetzwerk gemessene Höhen und angefiteter Sinus (blau). *Mit Genehmigung der Firma Varian Medical Systems. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.*



6.3.5 Fazit

Am Fraunhofer IOF wurde erfolgreich der Prototyp eines Sensornetzwerks, bestehend aus drei GOBO-Projektor-basierten 3D-Sensoren, aufgebaut. Das System hat gezeigt, dass eine genaue und gleichzeitig latenzarme 3D-Vermessung in einer gesundheitskritischen Anwendung im Medizinbereich möglich ist. Außerdem wurde gezeigt, dass die konstruktive Zusammenführung der Datenströme dreier 3D-Sensoren möglich ist und damit die Vollständigkeit der 3D-Modelle erheblich verbessert werden kann.

Das System wird derzeit von Varian Medical Systems in die Serienproduktion überführt.

Abbildung 6.12 zeigt, welche Verbesserungen durch die in dieser Arbeit entwickelten Techniken bei dieser Anwendung mit Bezug auf die in Abschnitt 1.4 aufgestellten Kernthesen gemacht wurden.

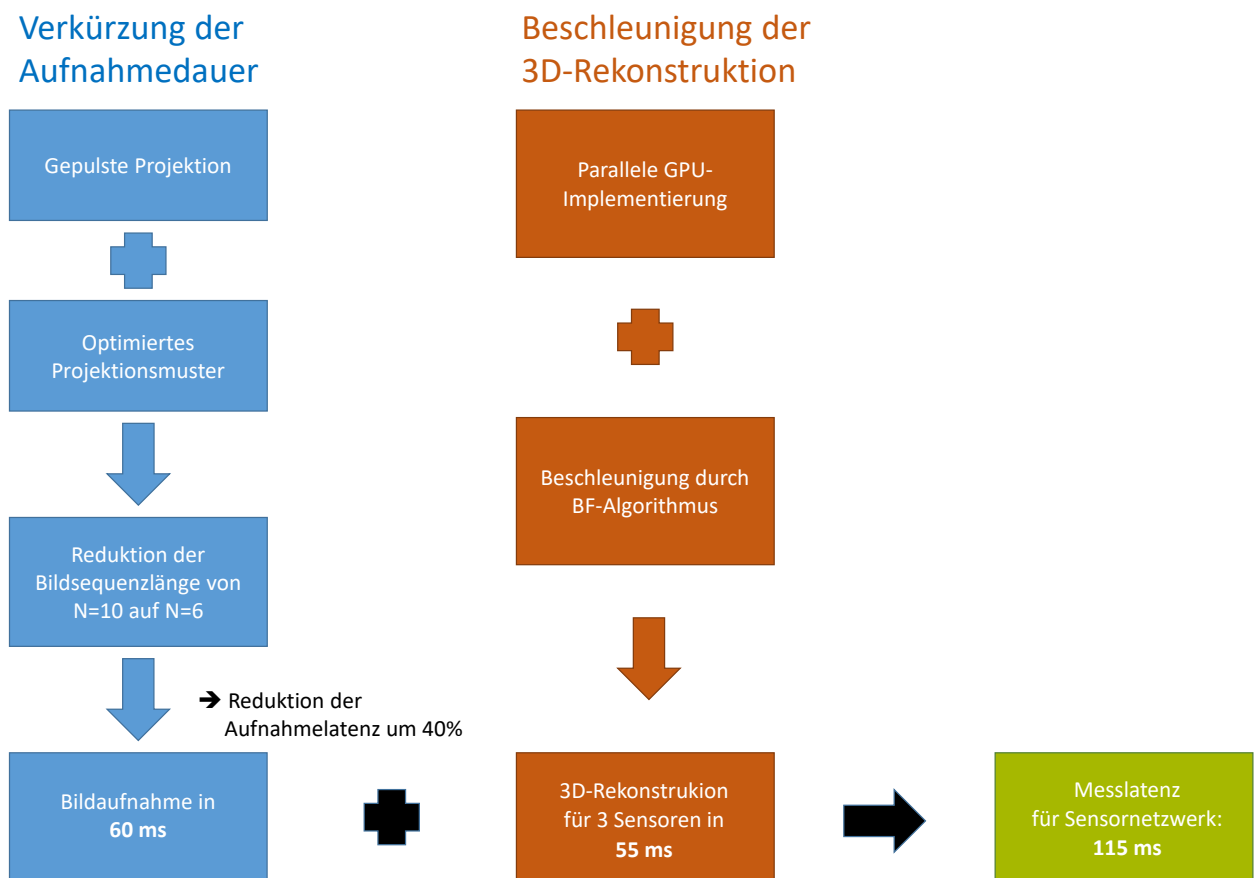


Abb. 6.12: Grafische Darstellung der erreichten Verbesserungen mit Bezug auf die in Abschnitt 1.4 aufgestellten Kernthesen.

6.4 Weitere Anwendungen

In diesem Abschnitt werden weitere Anwendungen beschrieben, die auf 3D-Sensorik mit GOBO-Projektion basieren und an denen der Autor dieser Arbeit mitgewirkt hat. Nicht bei allen hier beschriebenen Anwendungen ist niedrige Messlatenz eine zwingende Anforderung. Allerdings profitieren sie von den Entwicklungen, die im Zuge dieser Arbeit gemacht wurden.

6.4.1 Ausstellungsexponat für Mensch-Roboter-Interaktion

Für eine Ausstellung „Arbeitswelten der Zukunft“ zum Wissenschaftsjahr 2018 im Bundesministerium für Bildung und Forschung (BMBF) wurde am Fraunhofer IOF ein Exponat entwickelt und gebaut, welches die Interaktion zwischen Mensch und Roboter demonstriert (Abbildung 6.13).

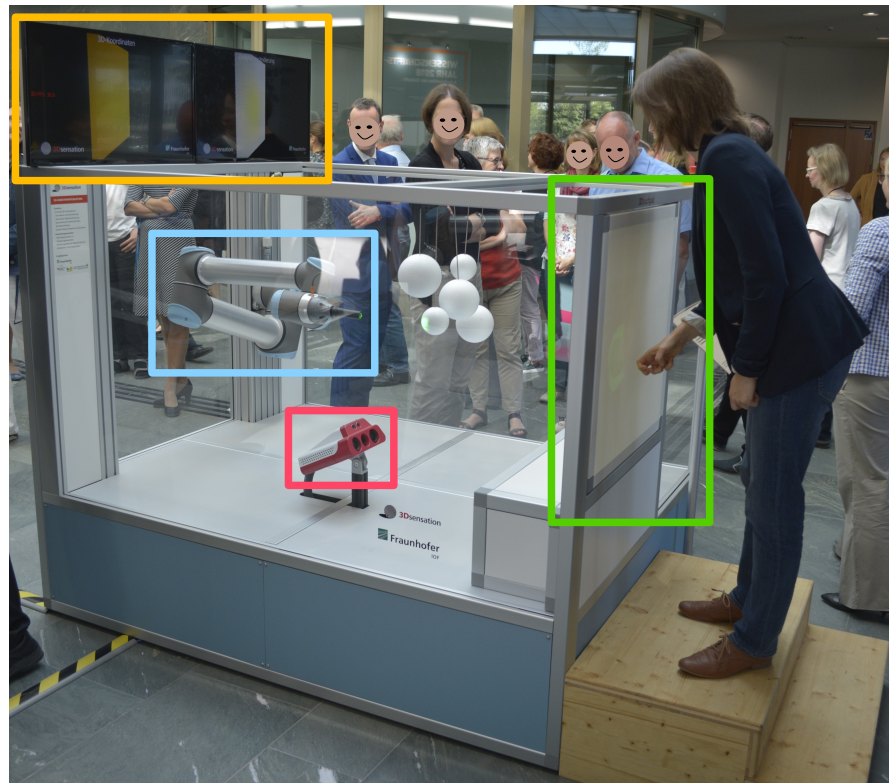


Abb. 6.13: Exponat zur Mensch-Roboter-Interaktion auf der Eröffnungsveranstaltung zur Ausstellung „Arbeitswelt der Zukunft“ des BMBF. Umrandet: grün: Touch-Membran, rot: 3D-Sensor, blau: Roboterarm, orange: Live-3D-Anzeige.

Das Exponat hat sechs wesentliche Komponenten:

1. eine Touch-Membran aus Latex, mit der die Besucher interagieren können,
2. einen 3D-Sensor, der die Membran mit einer 3D-Framerate von 36 Hz kontinuierlich vermisst,
3. einen Auswerte-PC unter dem Boden des Exponats,
4. einen Roboterarm mit einem leuchtenden „Finger“, der in Richtung der Membran nach vorne stuppst, wenn diese eingedrückt wird,
5. fünf Pendel mit Styroporkugeln, die vom Roboter angestuppt werden können,
6. eine Live-3D-Anzeige über der Rückwand.

Wird die Membran eingedrückt, so bestimmt der PC die am weitesten eingedrückte Position anhand des neusten 3D-Datensatzes und gibt diese Position

an den Roboter aus, der daraufhin seine „Hand“ zunächst in einer Ebene parallel zur Membran und danach senkrecht auf die Membran zu, Richtung der ermittelten Position, bewegt. Hat der Besucher die richtige Stelle der Membran eingedrückt, schiebt der Roboter dabei eines der Pendel an und bringt dieses zum Schwingen. Der Roboterarm bewegt sich unmittelbar danach in eine neue Parkposition im hinteren Bereich des Exponats, so dass das Pendel frei schwingen kann.

Der Autor dieser Arbeit hat die Software dieses Exponats entwickelt. Dies umfasst neben der 3D-Sensor-Software, die auf dem in Abschnitt 3.4 beschriebenen Software-Framework basiert, auch die Live-3D-Anzeige, die Notabschaltungslogik und die Steuerung des Roboters. Das Exponat wurde für die mehrwöchige Dauer der Ausstellung quasi Wartungsfrei betrieben.

Der 3D-Sensor entspricht weitestgehend dem in Abschnitt 6.2 beschriebenen.

6.4.2 Sensor für Crash-Tests

Am Fraunhofer IOF wurde ab ca. 2014 ein Hochgeschwindigkeits-3D-Sensor zur Aufnahme von Fahrzeug-Crash-Tests entwickelt (siehe Abbildung 6.14).



Abb. 6.14: Frontalansicht des Crash-Sensors

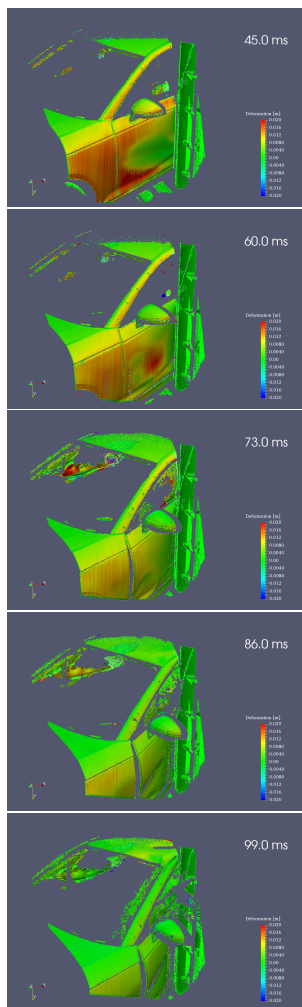


Abb. 6.15: 3D-Modelle von 5 Zeitpunkten während eines Seitenaufprall-Crash-Tests. Mit Genehmigung des Fraunhofer IOF. Vervielfältigung nur im Kontext dieser Veröffentlichung gestattet.

Dieser verwendet zwei Hochgeschwindigkeitskameras vom Typ Photron SA-X2, welche eine Bildwiederholrate von 13 kHz bei einer Auflösung von 1 Mpx bieten. Sie können aber auch – bei veringierter Auflösung – mit einer Bildwiederholrate von 60 kHz betrieben werden, was zur Aufnahme von Airbag-Entfaltungen genutzt wird. Ein GOBO-Projektor mit einer 575 W Gasentladungslampe als Lichtquelle der Beleuchtungseinheit projiziert die aperiodischen Streifenmuster. Die GOBO-Räder haben einen Durchmesser von 30 cm.

Optional kann das Sensor-System durch eine Wärme-Kamera ergänzt werden, so dass die Temperaturänderung der Materialien während des Crashes, die u.a. wegen deren Deformation auftreten, erfasst und auf die resultierenden 3D-Modelle aufprojiziert werden. [51, 52]

Das System wurde seit ca. 2014 schrittweise zur Marktreife entwickelt, ausgehend von einfachen Laboraufbauten, bis zu einem Industrie-tauglichen Sensorsystem, das in täglichem Einsatz bei einem deutschen Autohersteller ist. Der Autor hat einen Großteil der Software dieses Systems entwickelt, wobei das in Abschnitt 3.4 beschriebene Software-Framework zum Einsatz kommt.

Wegen der hohen 2D-Frameraten, mit denen die Kameras betrieben werden, zwischen 12 kHz bis 60 kHz, ist eine 3D-Rekonstruktion während der Messung nicht möglich. Die Kamerabilder werden zunächst im Puffer-Speicher der Kameras abgelegt, nach der Messung dann auf ein dazugehöriges Rechnersystem kopiert und dort ausgewertet.

Crash-Messungen

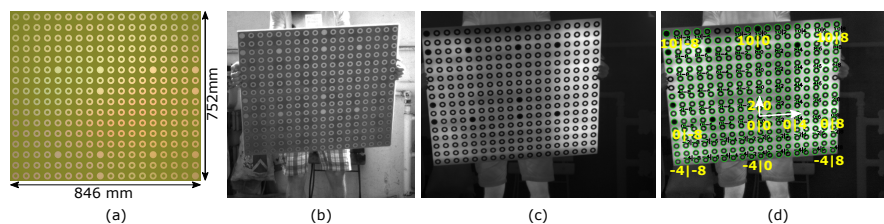
Der Sensor wurde während seiner Entwicklung in Messkampagnen bei mehreren deutschen Autoherstellern zur Messung von Crash-Tests eingesetzt. Unter anderem wurden Messungen zum Fußgängerschutz (Beschuss der Fahrzeugfront mit Bein- und Kopfsimulatoren), zum Insassenschutz (Beschuss von Fahrzeuginnenwänden mit Kopfsimulator), Airbag-Tests und Fahrzeugaufprall-Tests aufgenommen.

Abbildung 6.15 zeigt mehrere 3D-Modelle aus einem Seitenaufprall-Crash-Test.

Erweiterung um Thermografie-Kamera

Der Sensor wurde ergänzt durch eine Hochgeschwindigkeits-Thermografie-Kamera[51, 52, 87]. Kommt diese zum Einsatz, werden alle drei Kameras gemeinsam kalibriert, wozu ein spezielles Kalibrier-Target (Abbildung 6.16) verwendet wird, das sowohl von den Stereo-Kameras, als auch von der Thermografie-Kamera erfasst werden kann.

Abb. 6.16: Kalibrier-Target für Thermografie- und Stereo-Kameras. a) Foto, b) Bild der linken Kamera des Stereo-Paars, c) Bild der Thermografie-Kamera, d) Bild der Thermografie-Kamera mit Markierungen des Bilderkennungsalgorithmus. [87]



Nach der 3D-Rekonstruktion wird jeder 3D-Punkt mittels der Abbildungsgleichungen (siehe Gleichungen 2.1 und 2.3) der Thermografie-Kamera auf

das Wärmebild projiziert, um die Temperaturinformation für diesen Punkt zu bestimmen. Das Ergebnis ist eine 3D-Punktwolke, für die für jeden 3D-Punkt ein Temperaturwert vorliegt. Abbildung 6.17 zeigt die Messung eines Fahrer-Airbags mit diesem System während der Entfaltung.

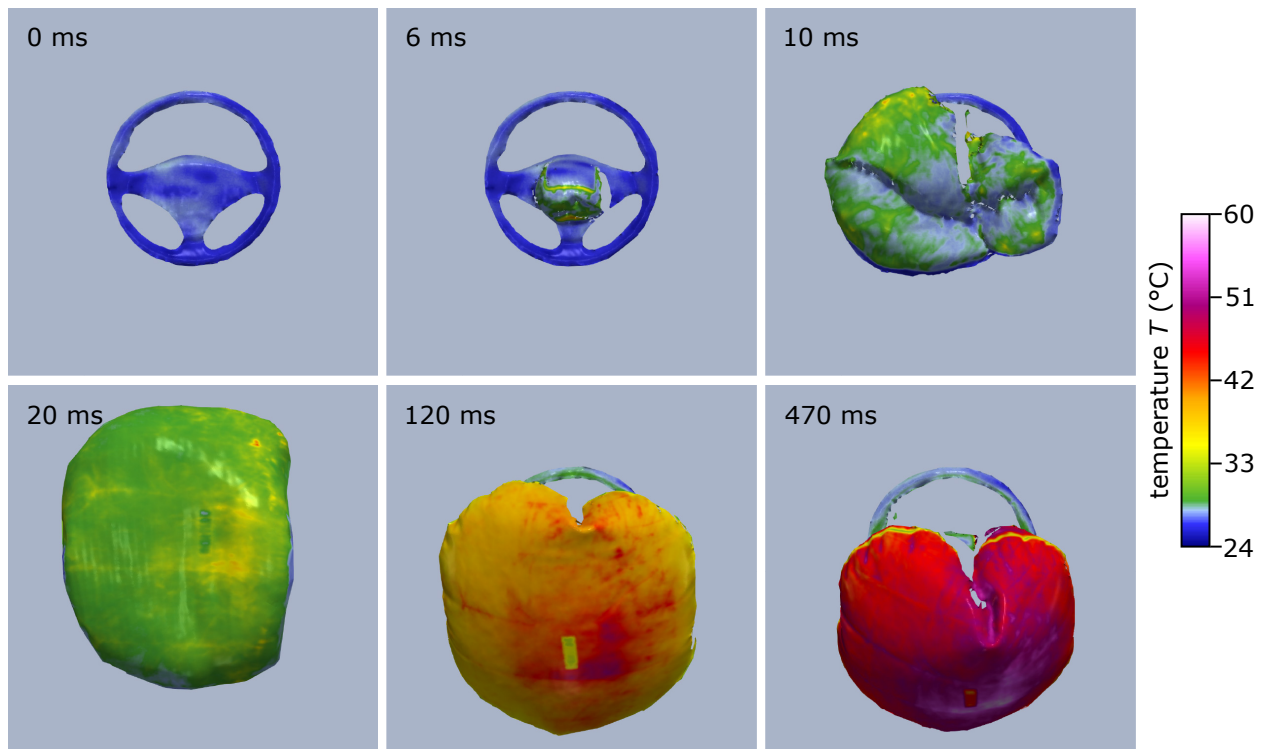


Abb. 6.17: 3D-Messung eines Fahrer-Airbags während der Entfaltung mit Temperaturwerten zu sechs verschiedenen Zeitpunkten [87].

Zukünftige Entwicklung

In jüngster Vergangenheit sind neue Hochleistungs-LED-Scheinwerfer für Crash-Tests kommerziell verfügbar geworden. Diese können gepulst betrieben werden und haben eine Lichtleistung, welche mit den Hochleistungs-Gasentladungslampen vergleichbar oder sogar größer ist. Dabei gibt es auch Modelle, welche eine hinreichend kleine Abstrahlfläche haben, dass ein Einsatz als Projektionsbeleuchtung möglich wird.

Es ist geplant, den Projektor des Crash-Sensors mit einer solchen LED-Beleuchtungseinheit auszustatten, um eine gepulste Projektion zu ermöglichen. Außerdem sollen in diesem Zuge neue GOBO-Räder mit optimiertem Muster produziert werden. Dies könnte eine Sequenzlängenverkürzung auf $N = 6$ ermöglichen.

6.4.3 Multimodaler 3D-sensor

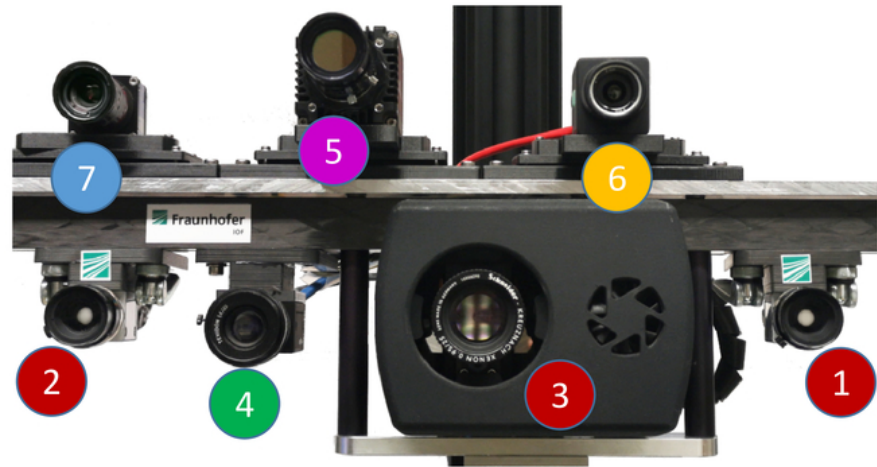


Abb. 6.18: Frontansicht des Multimodalsensors: 1/2: NIR-Stereo-Kameras, 3: NIR-GOBO-Projektor (mittlere Lichtwellenlänge: 850 nm), 4: Farbkamera, 5: SWIR-Kamera, 6: LWIR-Thermografie-Kamera, 7: Polarisationskamera.[88]

In einer Kooperation der TU-Ilmenau und des Fraunhofer IOFs wurde ein multimodaler 3D-Sensor entwickelt [88]. Dieser besteht aus einem GOBO-basierten 3D-Sensor, der um mehrere Spezialkameras erweitert wurde. (Abbildung 6.18)

Die 3D-Geometrie der Messobjekte wird, ähnlich wie bei den in den Abschnitten 6.3 und 6.2 vorgestellten 3D-Sensoren, mittels zweier NIR-Kameras und einem NIR-GOBO-Projektor aufgenommen. Zusätzlich sind insgesamt vier weitere Kameras auf dem Sensor installiert: eine Farbkamera mit dem üblichen Bayer-Farbfiler, eine Kamera für kurzwelliges Infrarotlicht (SWIR), eine Kamera für langwelliges Infrarotlicht (LWIR), d.h. Wärmestrahlung, und eine Polarisationskamera.

Alle Kameras werden gemeinsam kalibriert (Abbildung 6.19 zeigt ein Kalibriertarget und dazugehörige Kamerabilder). Bei einer Messung wird dann zunächst die 3D-Geometrie des Messobjektes mit dem **BiCoS-Algorithmus** rekonstruiert. Danach wird jeder 3D-Punkt mittels der Kameraabbildungsgleichungen (Gleichungen 2.1 und 2.3) auf die Sensoren der Zusatzkameras projiziert, um den jeweiligen Helligkeitswert der Kamera für diesen 3D-Punkt zu bestimmen.

Das Ergebnis sind 3D-Punktwolken mit einer 3D-Framerate von 30 Hz, wobei für jeden 3D-Punkt zusätzlich zu den räumlichen Koordinaten die drei Farbwerte (Rot, Grün, Blau), der SWIR-Helligkeitswert, die Temperatur (berechnet mittels der Daten der LWIR-Kamera) und der Polarisationswert vorliegen. Abbildung 6.20 zeigt ein solches 3D-Modell.

Der Multimodal-Sensor kann überall da angewendet werden, wo lokalisierte Informationen in mehreren Spektralkanälen erforderlich sind. Eine vorgeschlagene Anwendung ist beispielsweise die kontaktlose Überwachung der Pulsfrequenz bei Patienten. Dazu wird zunächst die Stirn des Patienten im Farbbild erkannt und mittels der vorhandenen 3D-Information die zugehörigen SWIR-Werte gefunden. In diesen kann dann die Pulsfrequenz mittels Photoplethysmogramm (PPG)[89] bestimmt werden.

Eine weitere Anwendung ist die Objektsegmentierung. In [90] wurde der Sensor erfolgreich genutzt, um eine korrekte und präzise Segmentierung zwischen menschlichen Händen und gehaltenen Objekten basierend auf den Sensordaten durchzuführen. Das Lösen dieser Segmentierungsaufgabe ist eine wichtige Vorbedingung für die Übergabe von Objekten zwischen Mensch und Roboter.

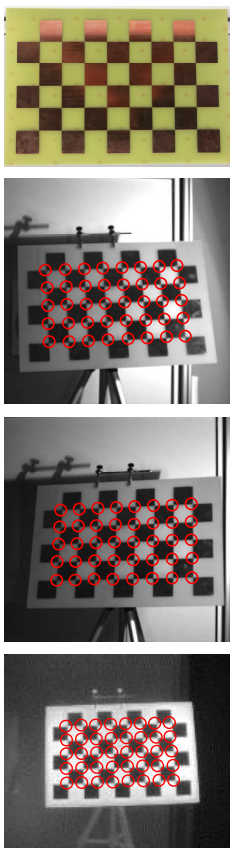


Abb. 6.19: Oben: Kupferbeschichtete Leiterplatte als Kalibriertarget. 2. von oben: Bild der linken NIR-Kamera, 3. von oben: Bild der rechten NIR-Kamera, unten: Bild der LWIR-Thermografie-Kamera. Rot markiert: erkannte Schachbrett-Ecken.[88]

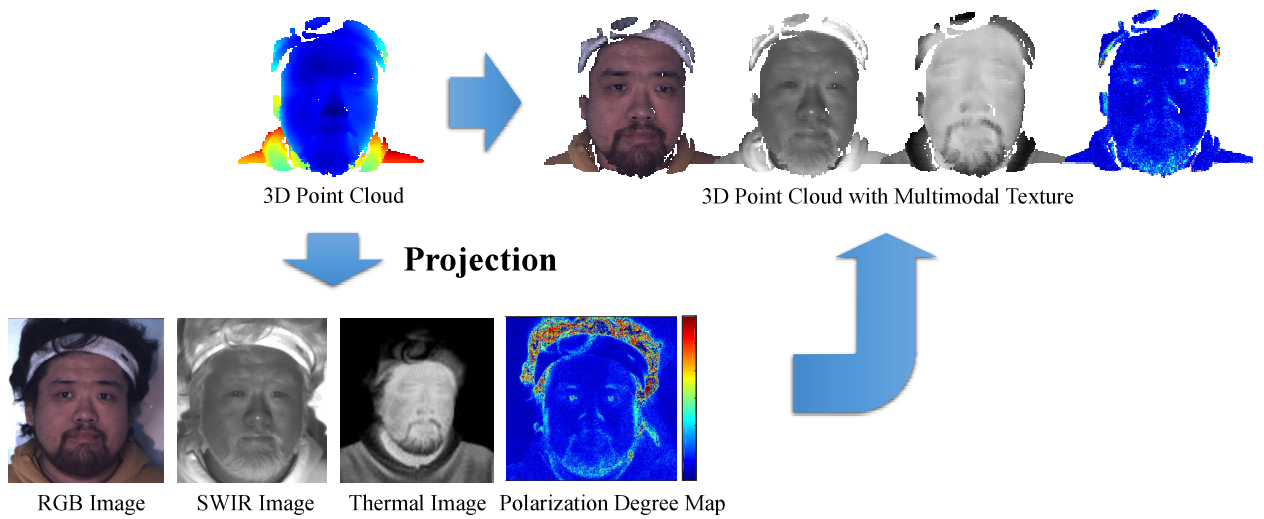


Abb. 6.20: 3D-Mapping der multimodalen Kamerainformation[88]

Ziel dieser Arbeit war es, genaue, optische 3D-Messtechnik so latenzarm zu machen, dass sie in Anwendungen eingesetzt werden kann, die eine direkte Rückkopplung innerhalb eines kurzen Zeitintervalls, z.B. innerhalb 100 ms benötigen. So soll beispielsweise bei der Strahlentherapie die Lage des Patienten kontinuierlich gemessen und mittels der Messergebnisse die Bestrahlungsanlage nachgeführt oder angehalten werden, um zu vermeiden, dass durch Bestrahlung der falschen Position gesundes Gewebe unnötig geschädigt wird.

Dafür wurde das Prinzip des GOBO-Projektor-basierten Stereo-Sensors eingesetzt, weil dieses erlaubt, mit hoher Genauigkeit zu messen, weil es außerdem bereits erfolgreich für die Messung kurzer Ereignisse wie z.B. Airbag-Entfaltungen eingesetzt wurde und es zudem mittels Infrarot-Projektor blend- bzw. irritationsfrei gemacht werden kann.

Bei diesem Verfahren wird ein sich zeitlich änderndes, aperiodisches Streifenmuster mittels eines GOBO-Projektors, d.h. eines Projektors mit rotierendem Dia-Rad, auf das Messobjekt projiziert, während zwei kalibrierte Kameras jeweils eine Bildsequenz synchron aufnehmen. Vor den Verbesserungen, die in dieser Arbeit vorgeschlagen wurden, waren Bildsequenz-Längen von $N = 10$ oder größer üblich.

Nach der Aufnahme werden innerhalb dieser Bildsequenzen korrespondierende Pixel-Paare gesucht, welche denselben Objektpunkt in den beiden Kameras zeigen. Das während der Aufnahme projizierte, zeitlich veränderliche Muster hat das Ziel, jeden Objektpunkt mit einer eindeutigen Helligkeitssequenz zu kodieren. Für die Korrespondenzsuche wird die Helligkeitssequenz eines Pixels der Referenzkamera mit der mehrerer Kandidatenpixel aus der zweiten Kamera verglichen. Das ähnlichste Pixel wird ausgewählt. Dazu kam bisher vor allem das Ähnlichkeitsmaß NCC (Normierte Kreuzkorrelation) zum Einsatz. Aus jedem Paar korrespondierender Pixel wird schließlich ein 3D-Punkt trianguliert.

In dieser Arbeit wurden zwei Teile des Sensorprinzips optimiert:

1. die Musterprojektion (Kapitel 4),
2. die 3D-Rekonstruktionsrechnung (Kapitel 3).

Ziel der Optimierung der Musterprojektion war es, die aus der Projektion resultierenden Helligkeitssequenzen so zu optimieren, dass sie selbst bei deutlich verkürzten Sequenzlängen, z.B. von $N = 6$, eine eindeutige Kodierung der Objektpunkte gewährleisten. Dazu wurden zwei Verbesserungen beschrieben und untersucht:

1. ein gepulster Projektionsmodus, der es erlaubt, dass sich das GOBO (Dia-Rad) zwischen zwei Bildaufnahmen deutlich weiterbewegt, wodurch die statistische Abhängigkeit der aufeinander folgenden Helligkeitswerte reduziert wird,
2. die numerische Optimierung des auf dem GOBO aufgetragenen Streifenmusters bezüglich der Eindeutigkeit der Helligkeitssequenzen.

Jede dieser Verbesserungen führte bei der Sequenzlänge $N = 6$ zu einer Reduktion der Messausfälle, d.h. der falschen oder fehlenden 3D-Punkte, um eine Größenordnung. Damit können mit dieser deutlich kürzeren Sequenzlänge vollständige 3D-Modelle erzeugt werden. Die Verkürzung der Sequenzlänge von $N = 10$ auf $N = 6$ bedeutet eine Reduktion der Aufnahmezeit um 40 %, d.h. bei einer Bildaufnahmezeit von 100 Hz auf 60 ms.

Um eine schnelle 3D-Rekonstruktionsrechnung zu ermöglichen, wurde ein C++-Software-Framework entwickelt, welches unter anderem den Einzug der Bilder von den Kameras und die 3D-Rekonstruktion auf der GPU mittels OpenCL umfasst. In diesem Framework wurde zunächst der rein NCC-basierte Referenzalgorithmus implementiert, womit eine 3D-Rekonstruktionszeit von 35 ms für einen Sensor mit 2 MPix Kameraauflösung erreicht wird.

Außerdem wurde ein neuer Algorithmus (BiCoS) für die Korrespondenzsuche entwickelt, der nicht mehr direkt mit den Helligkeitssequenzen arbeitet, sondern deren die Objektposition kodierende Information mittels Binäreigenschaften (Binary Futures, BF) approximiert. Für jedes Pixel wird eine Sequenz von Binäreigenschaften vorberechnet und als Bitvektoren gespeichert. Für die Korrespondenzsuche wird nun anstatt der NCC ein neues Ähnlichkeitsmaß, die BF-Ähnlichkeit verwendet: Um zwei Bitvektoren zu vergleichen, wird deren Anzahl gleicher Bits gezählt, was in zwei einzelnen GPU-Instruktionen ausgeführt werden kann. Der BiCoS-Algorithmus führt zu einer erheblichen Beschleunigung der 3D-Rekonstruktion. Für den 2 MPix-Sensor wird damit eine Rekonstruktionszeit von 17 ms erreicht.

Zusammengenommen erlauben die Verbesserungen der Musterprojektion und der 3D-Rekonstruktion eine Messzeit von 77 ms bei einer Bildeinzugsrate von 100 Hz und 2 MPix Kameraauflösung.

In Kapitel 6 wurden mehrere Anwendungen der in dieser Arbeit entwickelten Methoden vorgestellt:

Ein 3D-Sensor, der in der Abteilung HNO des Universitätsklinikums Jena eingesetzt wird, um die Symptome an Gesichtslähmung leidender Patienten zu quantifizieren und deren Veränderung durch Muskeltraining zu überwachen. Dieses System arbeitet mit einer Bildeinzugsrate von 360 Hz, einer Kameraauflösung von 1 MPix und erlaubt die Rekonstruktion von 36 3D-Modellen pro Sekunde. Es erreicht in der derzeitigen Ausbaustufe eine Messzeit von 54 ms, wobei eine weitere Verkürzung der Latenz auf 34 ms mittels eines Hardware-Updates geplant ist.

Eine weitere Anwendung ist ein Sensornetzwerk für die Überwachung von Patienten während der Strahlentherapie, das aus drei GOBO-basierten 3D-Sensoren besteht. Das System hat eine Messfeldgröße von $2 \text{ m} \times 1 \text{ m}$ und erlaubt die Erfassung der Atembewegungen des Patienten mit einer Genauigkeit von $65 \mu\text{m}$. Hier konnte, unter Anwendung der Projektions-Optimierung und der schnellen Rekonstruktion, eine Gesamt-Latenz, d.h. für die Aufnahme und Rekonstruktion aller drei 3D-Modelle, von 115 ms realisiert werden.

Die in dieser Arbeit beschriebenen Methoden bilden einerseits die Grundlage für die Auslegung zukünftiger 3D-Sensorsysteme. So können mit den für diese Arbeit entwickelten Software-Tools optimierte Projektionsmuster für unterschiedlichste Messfelder und Sensorgeometrien berechnet werden. Das Ergebnis, dass die gepulste Projektion erhebliche Vorteile für die Mustereindeutigkeit bei niedrigen Bildsequenz-Längen hat, kann zudem bei der Auslegung der Projektionslichtquellen berücksichtigt werden.

Andererseits erlaubt das entwickelte C++-Framework die praktische Verwendung der Sensoren für verschiedenste Anwendungen mit niedriger Messlatenz. Es übernimmt alle Teilaspekte der 3D-Rekonstruktion ab Bildeinzug von den Kameras bis zur Bereitstellung der 3D-Modelle für die Anwendungslogik. Dieses Framework bildet die Grundlage vieler zukünftiger Anwendungen am Fraunhofer IOF und bei Projektpartnern.

Literatur

- [1] Gavriel J. Iddan und Giora Yahav. „Three-dimensional imaging in the studio and elsewhere“. In: *Three-Dimensional Image Capture and Applications IV*. Hrsg. von Brian D. Corner, Joseph H. Nurre und Roy P. Pargas. Bd. 4298. International Society for Optics und Photonics. SPIE, 2001, S. 48–55. doi: [10.1117/12.424913](https://doi.org/10.1117/12.424913) (siehe S. 19).
- [2] S.B. Gokturk, H. Yalcin und C. Bamji. „A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions“. In: *2004 Conference on Computer Vision and Pattern Recognition Workshop*. Juni 2004, S. 35–35. doi: [10.1109/CVPR.2004.291](https://doi.org/10.1109/CVPR.2004.291) (siehe S. 19).
- [4] Simon Placht u. a. „Fast time-of-flight camera based surface registration for radiotherapy patient positioning“. In: *Medical Physics* 39.1 (2012), S. 4–17. doi: <https://doi.org/10.1118/1.3664006> (siehe S. 19).
- [5] Travis S. Taylor. *Introduction to Laser Science and Engineering*. Taylor & Francis Ltd., Aug. 2019 (siehe S. 19).
- [6] You Li und Javier Ibanez-Guzman. „Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems“. In: *IEEE Signal Processing Magazine* 37.4 (Juli 2020), S. 50–61. doi: [10.1109/MSP.2020.2973615](https://doi.org/10.1109/MSP.2020.2973615) (siehe S. 19).
- [7] Mojtaba Karimi u. a. „LoLa-SLAM: Low-Latency LiDAR SLAM Using Continuous Scan Slicing“. In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), S. 2248–2255. doi: [10.1109/LRA.2021.3060721](https://doi.org/10.1109/LRA.2021.3060721) (siehe S. 19).
- [8] Arthur P. Cracknell. *Introduction to Remote Sensing*. 2. Aufl. Taylor & Francis Inc., Apr. 2007 (siehe S. 19).
- [10] Kong-man (German) Cheung, Simon Baker und Takeo Kanade. „Shape-From-Silhouette Across Time Part I: Theory and Algorithms“. In: *International Journal of Computer Vision* 62 (3 2005), S. 221–247. doi: [10.1007/s11263-005-4881-5](https://doi.org/10.1007/s11263-005-4881-5) (siehe S. 19).
- [11] Johanna Roussel u. a. „3D Surface Reconstruction of Plant Seeds by Volume Carving: Performance and Accuracies“. In: *Frontiers in Plant Science* 7 (2016), S. 745. doi: [10.3389/fpls.2016.00745](https://doi.org/10.3389/fpls.2016.00745) (siehe S. 20).
- [12] Hanno Scharr u. a. „Fast High Resolution Volume Carving for 3D Plant Shoot Reconstruction“. In: *Frontiers in Plant Science* 8 (2017), S. 1680. doi: [10.3389/fpls.2017.01680](https://doi.org/10.3389/fpls.2017.01680) (siehe S. 20).
- [13] Jean-Marc Hasenfratz u. a. „Real-Time Capture, Reconstruction and Insertion into Virtual World of Human Actors“. In: *Vision, Video and Graphics*. Eurographics. Bath, United Kingdom: Elsevier, 2003, S. 49–56 (siehe S. 20).
- [15] *Handbuch zur industriellen Bildverarbeitung*. 3. Aufl. Michael Sackewitz, Fraunhofer-Allianz Vision, 2017 (siehe S. 20).
- [16] Albrecht Meydenbauer. „Die Photogrammetrie.“ In: *Wochenblatt des Architekten-Vereins zu Berlin* 14, 15, 16 (Apr. 1867), S. 125–126, 139–140, 149–150 (siehe S. 20).
- [17] Thomas Luhmann u. a. *Close-Range Photogrammetry and 3D Imaging*. 3. Aufl. Walter de Gruyter, 2019 (siehe S. 20).
- [18] Bill Triggs u. a. „Bundle Adjustment — A Modern Synthesis“. In: *Vision Algorithms: Theory and Practice*. Hrsg. von Bill Triggs, Andrew Zisserman und Richard Szeliski. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, S. 298–372 (siehe S. 20).
- [19] D Marr und T Poggio. „Cooperative computation of stereo disparity“. In: *Science* 194.4262 (1976), S. 283–287. doi: [10.1126/science.968482](https://doi.org/10.1126/science.968482) (siehe S. 21).
- [20] Stephen T. Barnard und Martin A. Fischler. „Computational Stereo“. In: *ACM Comput. Surv.* 14.4 (Dez. 1982), S. 553–572. doi: [10.1145/356893.356896](https://doi.org/10.1145/356893.356896) (siehe S. 21).
- [21] Chichyang Chen und Y.F. Zheng. „Passive and active stereo vision for smooth surface detection of deformed plates“. In: *IEEE Transactions on Industrial Electronics* 42.3 (Juni 1995), S. 300–306. doi: [10.1109/41.382141](https://doi.org/10.1109/41.382141) (siehe S. 21).

- [22] Daniel Scharstein und Richard Szeliski. „A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms“. In: *International Journal of Computer Vision* 47.1 (Apr. 2002), S. 7–42. doi: [10.1023/A:1014573219977](https://doi.org/10.1023/A:1014573219977) (siehe S. 21, 39).
- [23] Stefano Mattoccia und Matteo Poggi. „A Passive RGBD Sensor for Accurate and Real-Time Depth Sensing Self-Contained into an FPGA“. In: *Proceedings of the 9th International Conference on Distributed Smart Cameras*. ICDSC '15. Seville, Spain: Association for Computing Machinery, 2015, S. 146–151. doi: [10.1145/2789116.2789148](https://doi.org/10.1145/2789116.2789148) (siehe S. 21).
- [25] Kurt Konolige. „Projected texture stereo“. In: *2010 IEEE International Conference on Robotics and Automation*. Mai 2010, S. 148–155. doi: [10.1109/ROBOT.2010.5509796](https://doi.org/10.1109/ROBOT.2010.5509796) (siehe S. 21).
- [27] Zhengyou Zhang. „Microsoft Kinect Sensor and Its Effect“. In: *IEEE MultiMedia* 19.2 (Feb. 2012), S. 4–10. doi: [10.1109/MMUL.2012.24](https://doi.org/10.1109/MMUL.2012.24) (siehe S. 21).
- [28] Andreas Breitbarth u. a. „Measurement accuracy and dependence on external influences of the iPhone X TrueDepth sensor“. In: *Photonics and Education in Measurement Science 2019*. Bd. 11144. International Society for Optics und Photonics. SPIE, 2019, S. 27–33 (siehe S. 21).
- [29] Frank Forster. „A High-Resolution and High Accuracy Real-Time 3D Sensor Based on Structured Light“. In: *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*. Juni 2006, S. 208–215. doi: [10.1109/3DPVT.2006.13](https://doi.org/10.1109/3DPVT.2006.13) (siehe S. 21).
- [30] Joaquim Salvi, Jordi Pagès und Joan Batlle. „Pattern codification strategies in structured light systems“. In: *Pattern Recognition* 37.4 (2004). Agent Based Computer Vision, S. 827–849. doi: <https://doi.org/10.1016/j.patcog.2003.10.002> (siehe S. 22).
- [31] Sam Van der Jeught und Joris J.J. Dirckx. „Real-time structured light profilometry: a review“. In: *Optics and Lasers in Engineering* 87 (2016). Digital optical & Imaging methods in structural mechanics, S. 18–31. doi: <https://doi.org/10.1016/j.optlaseng.2016.01.011> (siehe S. 22).
- [32] Sai Siva Gorthi und Pramod Rastogi. „Fringe projection techniques: Whither we are?“ In: *Optics and Lasers in Engineering* 48.2 (2010). Fringe Projection Techniques, S. 133–140. doi: <https://doi.org/10.1016/j.optlaseng.2009.09.001> (siehe S. 22).
- [33] Dirk Bergmann. „New approach for automatic surface reconstruction with coded light“. In: *SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation*. Bd. 2572. 1995. doi: [10.1117/12.216931](https://doi.org/10.1117/12.216931) (siehe S. 22).
- [34] Christian Bräuer-Burchardt u. a. „High-speed three-dimensional measurements with a fringe projection-based optical sensor“. In: *Optical Engineering* 53.11 (2014), S. 1–8. doi: [10.1117/1.0E.53.11.112213](https://doi.org/10.1117/1.0E.53.11.112213) (siehe S. 22).
- [35] Yueyi Zhang, Zhiwei Xiong und Feng Wu. „Fusion of Time-of-Flight and Phase Shifting for high-resolution and low-latency depth sensing“. In: *2015 IEEE International Conference on Multimedia and Expo (ICME)*. Juni 2015, S. 1–6. doi: [10.1109/ICME.2015.7177426](https://doi.org/10.1109/ICME.2015.7177426) (siehe S. 22).
- [36] Chao Zuo u. a. „Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review“. In: *Optics and Lasers in Engineering* 85 (2016), S. 84–103. doi: <https://doi.org/10.1016/j.optlaseng.2016.04.022> (siehe S. 22, 38).
- [37] Chao Zuo u. a. „Phase shifting algorithms for fringe projection profilometry: A review“. In: *Optics and Lasers in Engineering* 109 (2018), S. 23–59. doi: <https://doi.org/10.1016/j.optlaseng.2018.04.019> (siehe S. 22, 38).
- [38] Song Zhang. „Absolute phase retrieval methods for digital fringe projection profilometry: A review“. In: *Optics and Lasers in Engineering* 107 (2018), S. 28–37. doi: <https://doi.org/10.1016/j.optlaseng.2018.03.003> (siehe S. 22, 38).
- [39] Wolfgang Schreiber und Gunther Notni. „Theory and arrangements of self-calibrating whole-body 3-D-measurement systems using fringe projection technique“. In: *Optical Engineering* 39.1 (2000), S. 159–169. doi: [10.1117/1.602347](https://doi.org/10.1117/1.602347) (siehe S. 23).
- [40] Daniel Moreno und Gabriel Taubin. „Simple, Accurate, and Robust Projector-Camera Calibration“. In: *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*. Okt. 2012, S. 464–471. doi: [10.1109/3DIMPVT.2012.77](https://doi.org/10.1109/3DIMPVT.2012.77) (siehe S. 23).

- [41] Stefan Heist u. a. „Array projection of aperiodic sinusoidal fringes for high-speed three-dimensional shape measurement“. In: *Optical Engineering* 53.11 (2014), S. 112208 (siehe S. 23).
- [45] A. Brahm u. a. „3D shape measurement of glass and transparent plastics with a thermal 3D system in the mid-wave infrared“. In: Bd. 10667. 2018. DOI: [10.1117/12.2304777](https://doi.org/10.1117/12.2304777) (siehe S. 23).
- [57] D Brown. „Decentering distortion of lenses“. In: *Inc, Eau Gallie, Florida, Tech. Rep* (1966) (siehe S. 29).
- [58] Zhengyou Zhang. „A Flexible New Technique for Camera Calibration“. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22 (Dez. 2000), S. 1330–1334. DOI: [10.1109/34.888718](https://doi.org/10.1109/34.888718) (siehe S. 29).
- [59] Richard Hartley und Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003 (siehe S. 30, 33).
- [60] Richard I. Hartley. „Theory and Practice of Projective Rectification“. In: *International Journal of Computer Vision* 35.2 (1999), S. 115–127. DOI: [10.1023/A:1008115206617](https://doi.org/10.1023/A:1008115206617) (siehe S. 31).
- [61] Fred E. Nicodemus. „Directional Reflectance and Emissivity of an Opaque Surface“. In: *Appl. Opt.* 4.7 (Juli 1965), S. 767–775. DOI: [10.1364/AO.4.000767](https://doi.org/10.1364/AO.4.000767) (siehe S. 35).
- [62] Omar Ait-Aider, Adrien Bartoli und Nicolas Andreff. „Kinematics from Lines in a Single Rolling Shutter Image“. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Juni 2007, S. 1–6. DOI: [10.1109/CVPR.2007.383119](https://doi.org/10.1109/CVPR.2007.383119) (siehe S. 36).
- [63] Simon Baker u. a. „Removing rolling shutter wobble“. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Juni 2010, S. 2392–2399. DOI: [10.1109/CVPR.2010.5539932](https://doi.org/10.1109/CVPR.2010.5539932) (siehe S. 36).
- [64] Beau Tippetts u. a. „Review of stereo vision algorithms and their suitability for resource-limited systems“. In: *Journal of Real-Time Image Processing* 11 (1 Jan. 2016), S. 5–25. DOI: [10.1007/s11554-012-0313-2](https://doi.org/10.1007/s11554-012-0313-2) (siehe S. 38).
- [65] Nalpantidis Lazaros, Georgios Christou Sirakoulis und Antonios Gasteratos. „Review of Stereo Vision Algorithms: From Software to Hardware“. In: *International Journal of Optomechatronics* 2.4 (2008), S. 435–462. DOI: [10.1080/15599610802438680](https://doi.org/10.1080/15599610802438680) (siehe S. 38, 39).
- [66] Peter Albrecht und Bernd Michaelis. „Improvement of the spatial resolution of an optical 3-D measurement procedure“. In: *IEEE Transactions on Instrumentation and Measurement* 47.1 (Feb. 1998), S. 158–162. DOI: [10.1109/19.728810](https://doi.org/10.1109/19.728810) (siehe S. 39).
- [67] Martin Schaffer, Marcus Grosse und Richard Kowarschik. „High-speed pattern projection for three-dimensional shape measurement using laser speckles“. In: *Appl. Opt.* 49.18 (Juni 2010), S. 3622–3629. DOI: [10.1364/AO.49.003622](https://doi.org/10.1364/AO.49.003622) (siehe S. 39).
- [68] Heiko Hirschmüller, Peter R. Innocent und Jon Garibaldi. „Real-Time Correlation-Based Stereo Vision with Reduced Border Errors“. In: *International Journal of Computer Vision* 47.1 (Apr. 2002), S. 229–246. DOI: [10.1023/A:1014554110407](https://doi.org/10.1023/A:1014554110407) (siehe S. 39).
- [69] Marcus Große. „Untersuchungen zur korrelationsbasierten Punktzuordnung in der stereophotogrammetrischen 3D-Objektvermessung unter Verwendung von Sequenzen strukturierter Beleuchtung“. Diss. Friedrich-Schiller-Universität Jena, Juni 2013 (siehe S. 40).
- [70] Stefan Heist u. a. „Theoretical considerations on aperiodic sinusoidal fringes in comparison to phase-shifted sinusoidal fringes for high-speed three-dimensional shape measurement“. In: *Applied optics* 54.35 (2015), S. 10541–10551 (siehe S. 40).
- [71] Aaftab Munshi. „The OpenCL specification“. In: *2009 IEEE Hot Chips 21 Symposium (HCS)*. Aug. 2009, S. 1–314. DOI: [10.1109/HOTCHIPS.2009.7478342](https://doi.org/10.1109/HOTCHIPS.2009.7478342) (siehe S. 41).
- [72] Ethan Rublee u. a. „ORB: An Efficient Alternative to SIFT or SURF“. In: *Proceedings of the 2011 International Conference on Computer Vision. ICCV '11*. Washington, DC, USA: IEEE Computer Society, 2011, S. 2564–2571. DOI: [10.1109/ICCV.2011.6126544](https://doi.org/10.1109/ICCV.2011.6126544) (siehe S. 43).
- [73] Michael Calonder u. a. „BRIEF: Binary Robust Independent Elementary Features“. In: *Eur. Conf. Comput. Vis.* Bd. 6314. Sep. 2010, S. 778–792. DOI: [10.1007/978-3-642-15561-1_56](https://doi.org/10.1007/978-3-642-15561-1_56) (siehe S. 43).

- [74] S. Leutenegger, M. Chli und R.Y. Siegwart. „BRISK: Binary Robust invariant scalable keypoints“. In: cited By 1705. 2011, S. 2548–2555. doi: [10.1109/ICCV.2011.6126542](https://doi.org/10.1109/ICCV.2011.6126542) (siehe S. 43).
- [75] Jared Heinly, Enrique Dunn und Jan-Michael Frahm. „Comparative Evaluation of Binary Features“. In: *Computer Vision – ECCV 2012*. Hrsg. von Andrew Fitzgibbon u. a. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, S. 759–773 (siehe S. 44).
- [76] Li Liu u. a. „Local binary features for texture classification: Taxonomy and experimental study“. In: *Pattern Recognition* 62 (2017), S. 135–160. doi: <https://doi.org/10.1016/j.patcog.2016.08.032> (siehe S. 44).
- [77] Louis F. Williams. „A Modification to the Half-Interval Search (Binary Search) Method“. In: *Proceedings of the 14th Annual Southeast Regional Conference*. ACM-SE 14. Birmingham, Alabama: Association for Computing Machinery, 1976, S. 95–101. doi: [10.1145/503561.503582](https://doi.org/10.1145/503561.503582) (siehe S. 46).
- [79] Anika Brahm u. a. „Fast 3D NIR systems for facial measurement and lip-reading“. In: Bd. 10220. 2017. doi: [10.1117/12.2263283](https://doi.org/10.1117/12.2263283) (siehe S. 50, 77).
- [80] Malte Dubrall. „Optimierung der Mustergenerierung eines GOBO-Projektors für die 3D-Messtechnik“. Bachelorarbeit. Ernst-Abbe-Hochschule Jena, 2020 (siehe S. 52).
- [81] Mira Finkensieper, Gerd Fabian Volk und Orlando Guntinas-Lichius. „Erkrankungen des Nervus facialis“. In: *Laryngorhinootologie* 91.2 (Feb. 2012), S. 121–142. doi: [10.1055/s-0031-1300965](https://doi.org/10.1055/s-0031-1300965) (siehe S. 75).
- [82] Christos Sagonas u. a. „A Semi-automatic Methodology for Facial Landmark Annotation“. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Juni 2013, S. 896–903. doi: [10.1109/CVPRW.2013.132](https://doi.org/10.1109/CVPRW.2013.132) (siehe S. 77).
- [83] Ralph Gross u. a. „Multi-PIE“. In: *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*. Sep. 2008, S. 1–8. doi: [10.1109/AFGR.2008.4813399](https://doi.org/10.1109/AFGR.2008.4813399) (siehe S. 77).
- [84] Henrik Voigt. „Nutzen und Auswirkungen von Feedbackschleifen auf den agilen Softwareentwicklungsprozess am Beispiel einer GUI-Entwicklung in der deklarativen Programmiersprache Qt QML“. Bachelorarbeit. Friedrich-Schiller-Universität Jena, Fakultät für Mathematik und Informatik, 2018 (siehe S. 79).
- [85] Christoph Munkelt u. a. „High performance, low latency 3D sensor network for live full object reconstruction (Conference Presentation)“. In: *Dimensional Optical Metrology and Inspection for Practical Applications VII*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 10667. International Society for Optics und Photonics. SPIE, 2018 (siehe S. 82).
- [86] Christoph Munkelt u. a. „Large-volume NIR pattern projection sensor for continuous low-latency 3D measurements“. In: *Dimensional Optical Metrology and Inspection for Practical Applications VIII*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 10991. International Society for Optics und Photonics. SPIE, 2019, S. 109–115 (siehe S. 82).
- [88] Maik Rosenberger u. a. „3D high-resolution multimodal imaging system for real-time applications“. In: *Dimensional Optical Metrology and Inspection for Practical Applications IX*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 11397. International Society for Optics und Photonics. SPIE, 2020, S. 21–30 (siehe S. 90, 91).
- [89] Kirk Shelley, Stacey Shelley und Carol Lake. „Pulse oximeter waveform: photoelectric plethysmography“. In: *Clinical monitoring* 2 (2001) (siehe S. 90).
- [90] Yan Zhang u. a. „Point Cloud Hand–Object Segmentation Using Multimodal Imaging with Thermal and Color Data for Safe Robotic Object Handover“. In: *Sensors* 21.16 (2021). doi: [10.3390/s21165676](https://doi.org/10.3390/s21165676) (siehe S. 90).

Eigene referierte Veröffentlichungen

- [43] Stefan Heist u. a. „High-speed three-dimensional shape measurement using GOBO projection“. In: *Optics and Lasers in Engineering* 87 (2016). Digital optical & Imaging methods in structural mechanics, S. 90–96. doi: <https://doi.org/10.1016/j.optlaseng.2016.02.017> (siehe S. 23, 24, 34, 50).
- [46] Martin Landmann u. a. „3D shape measurement of objects with uncooperative surface by projection of aperiodic thermal patterns in simulation and experiment“. In: *Optical Engineering* 59.9 (2020), S. 1–17. doi: [10.1117/1.0E.59.9.094107](https://doi.org/10.1117/1.0E.59.9.094107) (siehe S. 23).
- [48] Martin Landmann u. a. „High-resolution sequential thermal fringe projection technique for fast and accurate 3D shape measurement of transparent objects“. In: *Appl. Opt.* 60.8 (März 2021), S. 2362–2371. doi: [10.1364/AO.419492](https://doi.org/10.1364/AO.419492) (siehe S. 23).
- [51] Martin Landmann u. a. „High-speed 3D thermography“. In: *Optics and Lasers in Engineering* 121 (2019), S. 448–455. doi: <https://doi.org/10.1016/j.optlaseng.2019.05.009> (siehe S. 24, 34, 88).
- [54] Patrick Dietrich u. a. „BICOS—An Algorithm for Fast Real-Time Correspondence Search for Statistical Pattern Projection-Based Active Stereo Sensors“. In: *Applied Sciences* 9.16 (2019). doi: [10.3390/app9163330](https://doi.org/10.3390/app9163330) (siehe S. 25, 44, 45).
- [78] Stefan Heist u. a. „GOBO projection for 3D measurements at highest frame rates: a performance analysis“. In: *Light: Science & Applications* 7.1 (2018), S. 71. doi: [10.1038/s41377-018-0072-3](https://doi.org/10.1038/s41377-018-0072-3) (siehe S. 49, 53).

Eigene Tagungsbeiträge

- [9] David C. Tanner, Patrick Dietrich und Charlotte M. Krawczyk. „LIDAR-based outcrop characterisation-joint classification, surface and block size distribution“. In: *EGU General Assembly 2013*. Apr. 2013, EGU2013–3429 (siehe S. 19).
- [26] Christian Bräuer-Burchardt u. a. „3D reconstruction with single image pairs and structured light projection for short-term ultra-high-speed applications“. In: *Videometrics, Range Imaging, and Applications XIII*. Hrsg. von Fabio Remondino und Mark R. Shortis. Bd. 9528. International Society for Optics und Photonics. SPIE, 2015, S. 65–74. doi: [10.1117/12.2184704](https://doi.org/10.1117/12.2184704) (siehe S. 21).
- [42] Stefan Heist u. a. „Experimental comparison of laser speckle projection and array projection for high-speed 3D measurements“. In: *Optical Measurement Systems for Industrial Inspection IX*. Hrsg. von Peter Lehmann, Wolfgang Osten und Armando Albertazzi Gonçalves Jr. Bd. 9525. International Society for Optics und Photonics. SPIE, 2015, S. 282–289. doi: [10.1117/12.2184672](https://doi.org/10.1117/12.2184672) (siehe S. 23).
- [44] Anika Brahm u. a. „Non-destructive 3D shape measurement of transparent and black objects with thermal fringes“. In: *Dimensional Optical Metrology and Inspection for Practical Applications V*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 9868. International Society for Optics und Photonics. SPIE, 2016, S. 74–82. doi: [10.1117/12.2227093](https://doi.org/10.1117/12.2227093) (siehe S. 23).
- [47] Martin Landmann u. a. „Fast 3D Shape Measurement of Transparent Glasses by Sequential Thermal Fringe Projection“. In: *EPJ Web Conf.* Bd. 238. 2020, S. 06008. doi: [10.1051/epjconf/202023806008](https://doi.org/10.1051/epjconf/202023806008) (siehe S. 23).
- [50] Stefan Heist u. a. „GOBO projection-based high-speed three-dimensional shape measurement“. In: *Dimensional Optical Metrology and Inspection for Practical Applications V*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 9868. International Society for Optics und Photonics. SPIE, 2016, S. 99–107. doi: [10.1117/12.2227091](https://doi.org/10.1117/12.2227091) (siehe S. 24).

- [52] Martin Landmann u. a. „Multimodal sensor: high-speed 3D and thermal measurement“. In: *Photonics and Education in Measurement Science 2019*. Bd. 11144. International Society for Optics und Photonics. 2019, S. 1114403 (siehe S. 24, 88).
- [53] Patrick Dietrich u. a. „Efficient correspondence search algorithm for GOBO projection-based real-time 3D measurement“. In: *Dimensional Optical Metrology and Inspection for Practical Applications VIII*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 10991. International Society for Optics und Photonics. SPIE, 2019, S. 44–51. doi: [10.1117/12.2518641](https://doi.org/10.1117/12.2518641) (siehe S. 25, 44).
- [55] Patrick Dietrich u. a. „Improving the measurement robustness of active stereo 3D sensors by optimization of shifted aperiodic fringe patterns“. In: *Dimensional Optical Metrology and Inspection for Practical Applications IX*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 11397. International Society for Optics und Photonics. SPIE, 2020, S. 1–12. doi: [10.1117/12.2558583](https://doi.org/10.1117/12.2558583) (siehe S. 25, 54, 55, 62).
- [56] Patrick Dietrich u. a. „Low latency real time 3D sensor utilizing optimized aperiodic fringe patterns“. In: *Dimensional Optical Metrology and Inspection for Practical Applications X*. Hrsg. von Kevin G. Harding, Song Zhang und Beiwen Li. Bd. 11732. International Society for Optics und Photonics. SPIE, 2021, S. 19–27. doi: [10.1117/12.2586085](https://doi.org/10.1117/12.2586085) (siehe S. 25, 49, 73).
- [87] Martin Landmann u. a. „Simultaneous high-speed measurement of 3D surface shape and temperature“. In: *Dimensional Optical Metrology and Inspection for Practical Applications VIII*. Hrsg. von Kevin G. Harding und Song Zhang. Bd. 10991. International Society for Optics und Photonics. SPIE, 2019, S. 145–152. doi: [10.1117/12.2518211](https://doi.org/10.1117/12.2518211) (siehe S. 88, 89).

Patentanmeldung

- [49] Martin Landmann u. a. „Verfahren und Vorrichtung zum berührungslosen Vermessen einer Objektoberfläche“. Dt. Pat. DE 10 2020 201 536 A1. Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V. 12. Aug. 2021 (siehe S. 23).

Weitere Quellen

- [3] Nutzer „RCraig09“ via Wikimedia Commons. *Time of flight*. This file is licensed under the Creative Commons Attribution-ShareAlike 4.0 International Public License. <https://creativecommons.org/licenses/by-sa/4.0/legalcode>. 2020. URL: https://commons.wikimedia.org/wiki/File:20200501_Time_of_flight.svg (siehe S. 19).
- [14] Nutzer „UploadWizard“ via Wikimedia Commons. *Lichtschnittsensor*. 2011. URL: <https://commons.wikimedia.org/wiki/File:Lichtschnittsensor.svg> (siehe S. 20).
- [24] Nutzer „Kolossos“ via Wikimedia Commons. *Von der Kinect empfangenes Infrarotbild mit dem vom Laserprojektor ausgestrahltem Punktmuster*. This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License. <https://creativecommons.org/licenses/by-sa/3.0/deed.en>. 2011. URL: <https://en.wikipedia.org/wiki/File:Kinect2-ir-image.png> (siehe S. 21).

