

Phänomenologische Modellierung
und multistatische Simulation von Time-of-Flight
3D PMD Kameras

genehmigte

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

(Dr.-Ing.)

von

Dipl.-Ing. Valerij Peters

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen

1. Gutachter: Prof. Dr.-Ing. habil. Otmar Loffeld
2. Gutachter: Univ.-Prof. Dr. Ullrich Pietsch

Tag der mündlichen Prüfung: 14. Januar 2013

Gedruckt auf alterungsbeständigem holz- und säurefreiem Papier

Kurzfassung

Eines der zentralen Probleme der Automatisierungs-, Kontroll- und Sicherheitstechnik stellt die schnelle und präzise Erfassung von bewegten dreidimensionalen Objekten und ihrer räumlichen Orientierung dar. Herkömmliche optische Bildsensoren, die lediglich 2D-Aufnahmen mit Farbinformation liefern, stoßen hierbei schnell an ihre Grenzen. Bei konventionellen nach dem Prinzip der Stereoskopie arbeitenden Verfahren, mit deren Hilfe aus zwei oder mehr 2D-Bildern Tiefeninformation gewonnen werden kann, lässt sich das Korrespondenzproblem oftmals nur mit hohem Rechenaufwand oder nur teilweise lösen.

PMD-Sensoren, deren Funktionsprinzip auf Lichtlaufzeitverfahren basiert, liefern dagegen vektorielle dreidimensionale Messwerte eines 3D-Raums und ermöglichen somit direkt die 3D-Darstellung und Lokalisierung von Objekten. Sie sind selbstbeleuchtend im Infrarotbereich und können daher auch bei schlechteren Sichtverhältnissen (Nebel, Regen, Dunkelheit) im Gegensatz zu konventionellen optischen Systemen weitgehend zuverlässig arbeiten. Dank dieser Eigenschaften bieten PMD-Bildsensoren einen wesentlichen Vorteil gegenüber anderen Bildsensoren.

Obwohl die neueren Generationen der PMD-Sensoren bezüglich der Messgenauigkeit stark verbessert wurden, zeigen dennoch die zahlreichen Untersuchungen der letzten Jahre, dass die gemessenen Tiefenwerte, teilweise prinzipbedingt, von mehreren systematischen Fehlern verfälscht werden (Rauschen, Bewegungsartefakte, fliegende Pixel, Messfehler aufgrund des multistatischen Aufbaus der Kamera usw.). Ein sehr effektives Werkzeug bei den Untersuchungen solcher Fehler stellen Simulationen dar.

Mit ihrer Hilfe können den Anwendern, Algorithmenentwicklern und Systemplanern weitgehend unabhängig vom Entwicklungsstand und -fortschritt der PMD-Sensorhardware synthetische Referenzdaten erstellt sowie Erkenntnisse gewonnen werden, die in den Bereichen der Entwicklung dazu beitragen können, die Fehler zu klassifizieren und die existierenden Prozessierungs- und Koregistrierungsansätze weiterhin zu verbessern und zu optimieren. Mit synthetischen Testdaten können Algorithmen vorab auf ihre Funktionstüchtigkeit untersucht, Systemkonfigurationen auf ihre Eignung für eine spezielle Anwendung überprüft und Anwendern der PMD-Technologie vor dem realen Einsatz bereits ein Bild von der Leistungsfähigkeit, bzw. den Leistungsgrenzen einer bestimmten PMD-Konfiguration vermittelt werden.

Diese Arbeit befasst sich mit der Modellierung und Entwicklung eines solchen universell einsetzbaren PMD-Simulators.

Abstract

One of the central problems of automation, control and safety technology is fast and accurate detection of moving objects and their spatial orientation. Conventional optical sensors providing only 2D images with color information are not fully suitable for this. In conventional methods, according to the principle of stereoscopy, two images from the field of interest from different viewpoints are required to capture the object surface. An efficient real-time capable solution of the correspondence problem associates with high computational costs and is for some object types only partially solved.

In contrast, PMD sensors, whose operating principle is based on the measurement of the light propagation time, provide without further processing three-dimensional vector measurements of the environment and thus allow a direct 3D visualization and localization of objects. Another advantage over other image sensors is due to the self-illuminating construction the ability to suppress the most of the background light coming from artificial lighting or the sun and to increase thus the dynamic range.

The newer generations of PMD sensors have been greatly improved in terms of the measurement accuracy. However, the numerous studies have shown that the measured distance values are superposed with systematic errors (noise, motion artifacts, flying pixels, measurement error due to the multistatic constellation of the camera, etc.). Simulations are a very effective tool in the investigation of such errors. They can support algorithm developers and system designers for generating synthetic reference data of the PMD sensor hardware. The results can help to classify the errors and to improve the existing processing and coregistration algorithms. By using synthetic test data algorithms can previously be tested for their ability to function and system designs can be appraised for their suitability for a particular application.

This work deals with the modeling and development of such a universal PMD simulator.

Danksagung

Die vorliegende Dissertationsschrift entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter im Zentrum für Sensorsysteme (ZESS) an der Universität Siegen. An dieser Stelle möchte ich die Gelegenheit ergreifen und mich bei den Menschen bedanken, die diese Arbeit ermöglicht oder sie um einiges erleichtert haben.

Mein ganz besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr.-Ing. habil. Otmar Löffeld, der nicht nur die wissenschaftliche Betreuung meiner Dissertationsarbeit übernommen und sie mit seiner sehr hohen fachlichen Kompetenz, wertvollen Anregungen und zahlreichen und konstruktiven Diskussionen bereichert und vorangetrieben hat, sondern mich auch im Hinblick auf meine fachliche, berufliche und persönliche Weiterentwicklung stets gefördert hat.

Herrn Prof. Dr. Ullrich Pietsch danke ich herzlich für die fachliche wissenschaftliche Betreuung und Übernahme des Korreferats.

Bedanken möchte ich mich auch bei allen derzeitigen und ehemaligen Kollegen des Zentrums für Sensorsysteme, nicht nur für die hilfreiche und kompetente Unterstützung, sondern auch für das nette und harmonische Miteinander und das in allen Belangen tolle und einmalige Arbeitsklima im ZESS.

Ein großer Dank gilt meinem Kollegen Dr.-Ing. Holger Nies, der mir seit meiner Diplomarbeit mit seinen Ideen, Ratschlägen und Anmerkungen stets zur Seite stand. Herrn Dr.-Ing. Klaus Hartmann, Herrn Dipl.-Ing. Omar Löprrich, Herrn Dipl.-Ing. Arnd Sluiter und Herrn Dr. rer. nat. Rainer Bornemann möchte ich für die fachliche Unterstützung und wertvolle Ratschläge danken. Herrn Dipl.-Ing. Sven Stark danke ich für seine Hilfe bei der Aufnahme realer PMD-Daten. Mein Dank gilt auch meinen ehemaligen Kollegen Dr.-Ing. Ulrich Gebhard und Dr.-Ing. habil. Stefan Knedlik für die Unterstützung und die angenehme Zusammenarbeit während meiner Zeit im ZESS.

Meinen tiefsten Dank möchte ich meiner Frau und meinen Eltern aussprechen. Ohne ihre Liebe, Geduld, moralische und tatkräftige Unterstützung wäre diese Arbeit unmöglich gewesen. Auch unseren kleinen Kindern möchte ich dafür danken, dass sie mich in der Endphase meiner Promotion so liebevoll motiviert und unterstützt haben.

Inhaltsverzeichnis

Inhaltsverzeichnis	IX
Verzeichnis häufig verwendeter Abkürzungen	XII
1 Einleitung	1
1.1 Motivation	2
1.2 Zielsetzung	3
1.3 Gliederung der Arbeit	5
2 PMD Grundlagen	7
2.1 Laufzeitverfahren Time-of-Flight	7
2.2 Aufbau und Funktionsweise des Photogate-PMD	9
2.3 Überführung in eine PMD-Kamera	16
2.4 Aufbau einer PMD-Kamera	17
2.5 Leistungsgrenzen eines PMD-Systems	18
3 Beleuchtung – Modellierung und Simulation	21
3.1 Aufgabenstellung der Lichtsimulationskomponente (Light Propagation Module)	21
3.2 Physikalische Betrachtungen	22
3.2.1 Lichtquellen	23
3.2.2 Lichtausbreitung und Ausbreitungsmedium	25
3.2.3 Simulation der Rückstreuungseigenschaften	25
3.3 Softwaretechnische Umsetzung	33
3.3.1 Modellierung einer Lichtquelle	36
3.3.2 Modellierung eines Beleuchters	38
3.3.3 Modellierung der Modulation	40
3.3.4 Ablaufsteuerung	46
3.3.5 Testszenario Beleuchtung	49
4 Koordinatensysteme	58
4.1 Weltkoordinatensystem	58
4.2 Kamerafestes Koordinatensystem	59
4.3 Trajektorienkoordinatensystem	59
4.4 Sensorkoordinatensystem	60
4.5 Optikkoordinatensystem	61
4.6 LED-Koordinatensystem	61
4.7 Beleuchter-Koordinatensystem	62
4.8 Koordinatentransformationen	62
5 Modellierung und Simulation einer PMD-Kamera	70
5.1 Objektumgebung und ihre Definition	70

5.2	Modellierung des PMD-Sensors	75
5.3	Modellierung eines Pixels	76
5.4	Lochkameramodell	77
5.5	Geometrische Modellierung der PMD-Kamera	80
5.6	Berechnung des zum Punktziel korrespondierenden Pixels	81
5.7	Berechnung der theoretischen Antwort des Sensors	83
	5.7.1 Berücksichtigung des 4-Phasenalgorithmus	87
5.8	Betrachtung der Bewegungsunschärfe	89
5.9	Quantisierung der Messwerte	91
6	Sichtbarkeitsanalyse	93
6.1	Sichtbarkeit: Voranalyse für einen Sensor	95
6.2	Sichtbarkeit: Voranalyse für eine Lichtquelle	98
6.3	Sichtbarkeit: Nachanalyse für polygonale Daten	99
6.4	Sichtbarkeit: Nachanalyse für Objekte in Punkteform	102
6.5	Benchmark Test	105
7	Linsenverzeichnung	107
7.1	Klassische Linsenverzeichnungen	107
7.2	Modellierung der Linsenverzeichnungen	109
8	Simulatorarchitektur	111
8.1	Frame Control Unit (FCU)	111
8.2	Visibility Check Module	112
8.3	Light Propagation Module	113
8.4	PMD-Modul	114
8.5	Visualisierung	115
8.6	Dynamikkomponente	115
8.7	Simulationsablauf	116
	8.7.1 Einlesevorgang und Parametrierung	116
	8.7.2 Initialisierung	117
	8.7.3 Simulation	117
	8.7.4 Zustand-Update	118
	8.7.5 Ergebnisse und Visualisierung	118
9	Beispielszenarien	119
9.1	Bi- bzw. Multistatischer Fehler	119
9.2	Simulation der Bewegungsunschärfe	127
9.3	Simulation: Abstandsmessung an schrägen Oberflächen	130
10	Zusammenfassung und Ausblick	133
10.1	Zusammenfassung	133
10.2	Ausblick	136
11	Literaturverzeichnis	139

12	Anhang.....	147
12.1	Alternativer Ansatz zur Berechnung der Pixelzugehörigkeit eines Punktes.....	147
12.2	Homogene Koordinaten.....	150
12.3	Betrachtung von dünnen bikonvexen Linsen.....	152

Verzeichnis häufig verwendeter Abkürzungen

c_0	Lichtgeschwindigkeit im Vakuum
τ_L	Laufzeit des Signals
T_m	Periode des modulierten Signals
I	Strahlungsintensität
R_L	Reflektionsgrad
T_L	Transmissionsgrad
A_L	Absorptionsgrad
Φ	Strahlungsleistung, Strahlungsfluss
Q	Strahlungsenergie
Ω	Raumwinkel
L	Strahlungsdichte
ϕ	Azimutwinkel
θ	Polarwinkel
E	Beleuchtungsstärke, Bestrahlungsstärke
T_{KW}	Transformationsmatrix vom KKS ins WKS
T_{WK}	Transformationsmatrix vom WKS ins KKS
\vec{r}_{sp}	Schnittpunktvektor
b_{PMD}	Breite des PMD Sensors
h_{PMD}	Höhe des PMD Sensors
φ_d	Phasenverschiebung
T_m	Periode des modulierten Signals
$k(\tau)$	Korrelationsfunktion
A_{eff}	Effektive Fläche
ε_1	Abstrahlwinkel
ε_2	Reflektionswinkel
\vec{e}	Einheitsvektor
\vec{n}	Normalvektor
\vec{P}, \vec{P}_{obj}	Im Allgemeinen Punktziel
\vec{P}^{proj}	Projektion des Punktzieles auf den Sensor

\vec{O}	Vektor zum Optikzentrum
σ_φ	Standardabweichung der Phase
σ_R	Standardabweichung des Abstandes
χ_{Sim}	Simulationskoeffizient
λ	Wellenlänge
I_0	Strahlungsintensität in Hauptrichtung
f_{mod}	Modulationsfrequenz
A_{sim}	Simulierte Amplitude des Pixels
R_{sim}	Simulierter Abstand des Pixels
\vec{v}_K	Vektor \vec{v} im kamerafesten Koordinatensystem
\vec{v}_S	Vektor \vec{v} im Sensor-Koordinatensystem
\vec{v}_L	Vektor \vec{v} im LED-Koordinatensystem
H	Grauwert eines Pixels
\vec{v}_W	Vektor \vec{v} im Weltkoordinatensystem
M_{WK}	Rotationsmatrix der Transformation aus dem KKS ins WKS
M_{KW}	Rotationsmatrix der Transformation aus dem WKS ins KKS
\vec{T}_{WK}	Verschiebungsvektor der Transformation aus dem KKS ins WKS
\vec{T}_{KW}	Verschiebungsvektor der Transformation aus dem WKS ins KKS
M_{TS}	Rotationsmatrix der Transformation aus dem SKS ins TKS
M_{KS}	Rotationsmatrix der Transformation aus dem SKS ins KKS
M_{SK}	Rotationsmatrix der Transformation aus dem KKS ins SKS
3D	Dreidimensional
BKS	Beleuchter-Koordinatensystem
CCD	Charge-Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CW	Continuous Wave (Modulation)
f	Brennweite
F	Anzahl der Zwischenframes
FCU	Frame Control Unit
GUI	Graphical User Interface (Benutzeroberfläche)
HSR	Hidden Surface Removal
KKS	Kamerafestes Koordinatensystem

LED	Light-emitting Diode (Lichtdiode)
LKS	LED-Koordinatensystem
LPM	Light Propagation Module
M	Anzahl der Zeilen od. Amplitude der Korrelationssignale
N	Anzahl der Spalten
OKS	Optik-Koordinatensystem
PMD	Photonic Mixer Device
SKS	Sensor-Koordinatensystem
TKS	Trajektorien-Koordinatensystem
TOF	Time-of-Flight
VCM	Visibility Check Module
WKS	Weltkoordinatensystem
ZESS	Zentrum für Sensorsysteme

1 Einleitung

Die Entwicklungen und das Interesse an berührungslosen, echtzeitfähigen Verfahren zur Vermessung von 3D-Szenen gewinnen in den letzten Jahren in vielen Bereichen rasant an Bedeutung. Mit Ausnahme von ein paar Spezialanwendungen, findet man heute im industriellen Bereich vorwiegend drei physikalisch unterschiedliche Messprinzipien: Triangulation, Interferometrie und Time-of-Flight Entfernungsmessung.

Triangulationsverfahren, welche prinzipiell dem menschlichen Sehen nachempfunden sind, lassen sich in zwei große Gruppen unterteilen – passive Stereoaufnahmen und aktive Punkt- und Lichtschnittprojektionsverfahren. Dabei wird die zu vermessene Szene bei passiven Verfahren aus mindestens zwei unterschiedlichen Richtungen mit Hilfe von lichtempfindlichen Detektoren (üblicherweise 2D CCD Sensoren) aufgenommen. Bei aktiven Verfahren wird ein Muster auf das Objekt projiziert, welches von einem positionsempfindlichen Detektor erfasst wird. Die Entfernung lässt sich dann auf der Grundlage der Disparität in den aus unterschiedlichen Winkeln aufgenommen Bildern bestimmen.

Punktschnittprojektionsverfahren bieten in der Regel eine hohe Genauigkeit und können auch bei schlecht reflektierenden und hoch texturierten Oberflächen zuverlässig arbeiten, die zu vermessene Szene muss aber Punkt für Punkt abgetastet und aufgenommen werden, was zu langen Aufnahmezeiten führt. Daher eignen sich solche Verfahren schlecht für bewegliche Szenen. Zur Steigerung der Geschwindigkeit kann zum Vermessen ein großflächiges Punktmuster genommen werden. Dieses Prinzip wird auch bei Lichtschnittprojektionsverfahren verwendet, bei denen die Szene mit Hilfe von flächenhaften Streifen- bzw. Weißlichtmustern abgetastet wird. Dadurch kann die Anzahl der verfügbaren Messpunkte pro Messvorgang signifikant gesteigert werden.

Bei passiven Triangulationsverfahren (Stereo Vision) wird die zu vermessene Bildszene mit möglichst diffusem (Tages)Licht erhellt, es wird oft keine spezielle Beleuchtung gebraucht. Durch parallele Aufnahme aus zwei versetzten Richtungen sind solche Systeme oft echtzeitfähig. Das zu lösende Korrespondenzproblem ist allerdings je nach Form und Beschaffenheit der Szene nicht immer trivial und teilweise nur mit hohem technischem Aufwand oder bei zu wenig Szenenkontrast gar nicht lösbar. Hinzu kommt noch ein weiterer für alle Triangulationsverfahren geltender Nachteil. Die Auflösung der Tiefeninformation hängt bei einer konstanten lateralen Auflösung des Detektors und einer konstanten Triangulationsbasis quadratisch von der zu messenden Distanz ab. Eine ausreichend gute Auflösung im Messbereich einiger Meter erfordert daher eine größere Triangulationsbasis sowie eine höhere Auflösung der Sensoren, was letztendlich die Ausmaße und den Preis des Systems negativ beeinflusst. Der typische Arbeitsbereich solcher Systeme liegt daher im Bereich einiger weniger Meter.

Interferometrieverfahren, insbesondere Weißlichtinterferometrie, beruhen auf der Messung eines vom Objekt reflektierten Lichtstrahls mit einem Referenzstrahl. Aus den so gemessenen Interferenzmustern, die aus Überlagerung von mindestens zwei monochromatischen Wellen mit einer festen Phasenverschiebung entstehen, wird die Entfernung anhand der Phasendifferenz bestimmt. Die Messgenauigkeit hängt dabei grundsätzlich von der Wellenlänge des verwendeten kohärenten Lichtes ab. Bei entsprechend kurzen Kohärenzlängen kann die Genauigkeit im Bereich weniger Nanometer liegen. Der Nachteil dieser Systeme besteht in der langen Messzeit, weswegen diese Methoden in der Messtechnik meistens für exakte Messungen und Visualisierung von Oberflächeneigenschaften verwendet werden.

Bei Time-of-Flight Verfahren wird die Entfernung aus der Laufzeit des eigens gesendeten Signals bestimmt, wobei die Laufzeit entweder direkt (Pulslaufzeitverfahren) oder indirekt (Phasenkorrelationsverfahren) bestimmt wird. Obwohl die ersten Anwendungen bereits Mitte letzten Jahrhundert entwickelt wurden und auch teilweise bis heute in der Form existieren, beschränkten sich die meisten davon eine lange Zeit auf einzelne Entfernungsmessungen. Erst mit den rasanten Entwicklungen im Bereich der Mikroelektronik kamen Ende des Jahrhunderts neue Verfahren zum Vermessen einer ganzen Szene. Eine der fortschreitendsten Neuentwicklungen im letzten Jahrzehnt im Bereich der Time-of-Flight Technik stellen 3D Kameras dar, bei denen die Auswertung der Entfernung direkt im Sensor durchgeführt wird. Dank dieser Eigenschaft können 3D Bilder mit einer Bildrate von aktuell bis zu 100 Bildern pro Sekunde aufgenommen werden, wodurch solche Systeme besonders in Bereichen Automatisierungstechnik, Robotik, Automotiv und Sicherheit zunehmend attraktiver werden.

Die in dieser Arbeit betrachtete und im Zentrum für Sensorsysteme entwickelte PhotoMixer Device (PMD) Kamera stellt eine solche Kamera dar.

1.1 Motivation

Der Fokus der vorliegenden Arbeit richtet sich auf das Teilgebiet der phänomenologischen Modellierung und Simulation einer 3D PMD-Kamera. Betrachtet man die einzelnen in der PMD-Technik wichtigen Teilprozesse, beginnend bei der Ansteuerung des Sensors und der Beleuchter, die Lichtgenerierung in den LED's, die Lichtausbreitung zum Messobjekt hin und zurück, die Rückstreuungseigenschaften, den Prozess der Ladungsgenerierung in den Photogates durch einfallendes moduliertes Licht, über die Auswertung der Ladungsdifferenz, bis hin zur abschließenden digitalen Ausgabe der Entfernungs- und Amplitudenbilder, so wird schnell ersichtlich, dass eine vollständige, realitätsnahe und dennoch überschaubare physikalische Modellierung dieser Prozesse in vollem Umfang kaum möglich ist. Bei solchen komplizierten Vorgängen bieten phänomenologische Modelle große Vorteile. Das Ziel der absoluten Korrektheit jedes Teilmodells steht dabei nicht so sehr im Vordergrund wie eine systemtheoretische Betrachtung.

tung und Nachbildung der einzelnen wichtigsten Teilkomponenten und -prozesse, wodurch eine „scheinbare Korrektheit“ in einem bestimmten Simulationsprozess erzielt werden kann.

Neben den bei CMOS-Sensoren typischen Grund-, Temperatur- und Schrottrauschen, kommt bei allen PMD-Kamera-Anordnungen physikalisch bedingt der sogenannte multistatische Fehler mehr oder weniger stark zum Tragen. Zusätzlich gibt es eine deutliche Abhängigkeit der Messunsicherheit von der empfangenen Lichtmenge. Die Genauigkeit der Messwerte hängt, ähnlich wie bei konventionellen bildgebenden Digitalkameras, sehr stark von den Einsatzbedingungen und den gewählten Betriebseinstellungen der Kamera ab. Mit Hilfe von diversen technischen Weiter- und Neuentwicklungen konnte in den letzten Jahren der Einsatzbereich der Kamera zwar deutlich erweitert werden, doch selbst mit diesen Verbesserungen gibt es keinen universellen, unter allen Bedingungen optimalen Betrieb der Kamera. Dieser hängt stark von der Fragestellung der Anwendung ab. Aufgrund der Überlagerung der genannten Effekte und der oft unmöglichen Trennung der einzelnen Rauscharten, stellt sich eine objektive Einschätzung der Größenordnung der einzelnen Einflüsse als sehr schwierig heraus. Hier können Simulationen oft als Analyseinstrument wichtig sein. Sie können außerdem entscheidende Anhaltspunkte bei der Entwicklung eines neuen Designs der Kamera, Dimensionierung und Parametrierung neuer Beleuchter, sowie bei Verifizierung und Neuentwicklung von Verarbeitungsalgorithmen liefern und somit die technologische Weiterentwicklung eines Systems aktiv unterstützen.

1.2 Zielsetzung

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines Simulationskonzeptes, welches aus einer in 3D gegebenen Ansammlung von Punkten und/oder Polygonen in relativen Sensorkoordinaten beschriebene synthetische Helligkeits- und Entfernungsbilder generiert, welche ein PMD-Sensor, installiert auf einer innerhalb dieser Szene bewegten Plattform, von dieser liefern würde.

Der in dieser Arbeit entwickelte software-basierte Simulator soll unterschiedliche Sensorgeometrien und konstruktive Designs berücksichtigen können, flexibel parametrierbar sein, relevante physikalische Effekte und auftretende Störphänomene realitätsgetreu darstellen, sowie Test- und Referenzdaten erzeugen können, mit denen in weiteren Schritten 3D-Bildverarbeitungs- und Koregistrierungsansätze auf ihre Leistungsfähigkeit hin untersucht werden können.

Komplementär zu der Simulation eines PMD-Sensors, soll im Rahmen des Simulationskonzeptes eine Simulationsumgebung für Beleuchter realisiert werden, in der die senderseitige Lichtausbreitung des modulierten Lichts und Beleuchtung der Szene simuliert werden kann.

Aufgrund des, bedingt durch die Kamerakonstruktion, multistatischen¹ Messaufbaus der Empfänger und Beleuchter ergibt sich bei der Laufzeitmessung eine Verzerrung durch unterschiedliche Hin- und Rückwege von gesendetem und empfangenem Signal. Dieser Effekt führt insbesondere im Nahbereich der Kamera, wo das Verhältnis des Abstandes zwischen dem Beleuchter und Empfänger zu dem Abstand zum Messobjekt noch relativ groß ist, zu signifikanten Messfehlern. Zur Steigerung der Genauigkeit der realen PMD Kameras muss dieser Effekt berücksichtigt und nach Möglichkeit kompensiert werden. In welcher Weise und wie stark sich dieser Effekt äußert, lässt sich allerdings nur mit Hilfe von Simulationen, bei denen äußere Störfaktoren über die ganze Testreihe konstant gehalten werden können, genauer feststellen.

Es existieren viele Simulatoren bzw. Anwendungen, die sich mit dem Thema Lichtgebung, Lichtausbreitung und Lichtreflektionen befassen. Einige wenige bekannte Simulatoren gibt es auch im Bereich 3D Simulationen bildgebender Systeme. Die meisten davon befassen sich mit Simulationen starrer bildgebender Systeme oder mit Systemen, die auf einem anderen Messprinzip als bei einer PMD-Kamera basieren. Der in dieser Arbeit vorgestellte Simulator für multistatische Beleuchter-PMD-Kamera Anordnungen bildet somit eine komplette Neuentwicklung in diesem Sektor. Folgende grundlegende Anforderungen werden an den Simulator gestellt:

- Es soll möglich sein, Sensoren unterschiedlicher Auflösung, Chip- und Pixelgröße zu simulieren.
- Das zugrunde liegende Simulationskonzept soll auf einem Lochkameramodell aufbauen. Einbindung der Möglichkeit anderer austauschbarer „Optik-Module“ bietet größere Freiheitsgrade bei der Auswahl der Optiken. Lösung → Modularer Ansatz zur Simulation der Optik.
- Definition und Simulation von Beleuchtern unterschiedlicher Form und Abstrahlcharakteristik ermöglichen die Berücksichtigung der örtlichen Gegebenheiten bei Signalausbreitung und liefern die für realitätsnahe Simulation notwendigen Informationen zur Szenenausleuchtung.
- Der Simulator soll weitgehend plattformunabhängig und ohne spezielle Anforderungen an die Hardware lauffähig sein (aktuelle Windows-, Linux- und Unix-Betriebssysteme). Der Einsatz eines 64-bit Betriebssystems wird aufgrund der oft großen Datenmengen als gegeben und optimal angenommen.
- 3D Umgebungsmodelle sollen aus Datensätzen eingelesen und objektweise beliebig zusammengestellt werden können. Die Simulation unterschiedlich definierter Szenenobjekte, insbesondere in Punkteform und Polygonform, soll möglich sein. Die Objekte können dabei dynamisch sein, d.h. ihre Form oder Position im Raum verändern können.

¹ Man spricht von einer multistatischen Anordnung, wenn sich die Sender und Empfänger an unterschiedlichen Positionen befinden. Fallen ihre Positionen zusammen, dann handelt es sich um eine monostatische Anordnung.

- Synthetisch generierte 3D/2D Datensätze einer Szenenaufnahme sollen aufgezeichnet und framewise analysiert werden können.

1.3 Gliederung der Arbeit

Zum besseren Verständnis der entwickelten Modellierungsansätze und Software-Algorithmen beginnt das Kapitel „*PMD Grundlagen*“ mit einer Einführung in die Grundlagen der Time-of-Flight Messverfahren. Es gibt einen Einblick in den grundsätzlichen Aufbau eines PMD-Pixels sowie in das zugrunde liegende Funktionsprinzip eines PMD-Sensors. Abgeschlossen wird mit dem Aufbau einer PMD-Kamera sowie mit den Leistungsgrenzen des Systems.

PMD-Systeme sind selbstbeleuchtend im Infrarotbereich. Das nachfolgende Kapitel „*Beleuchtung – Modellierung und Simulation*“ beschäftigt sich mit der Modellierung und Simulation verschiedener variabler Beleuchterausführungen. Es beginnt mit der Modellierung frei definierbarer Lichtquellen mit der anschließenden Überführung dieser in geschlossene Beleuchter. Das für die Berechnung der Lichtausbreitung zuständige „*Light Propagation Module*“ wird vorgestellt.

Im Kapitel „*Koordinatensysteme*“ werden die für die Modellierung und die systemtheoretische Beschreibung des PMD-Sensors notwendigen Koordinatensysteme und Transformationen behandelt.

Das darauf folgende Kapitel beschäftigt sich mit der Fragestellung der „*Modellierung und Simulation einer PMD-Kamera*“. Es beginnt mit der Definition der Objektumgebung, gefolgt von der geometrischen Modellierung eines PMD-Sensors sowie der eines Pixels. Zur Beschreibung der Kamera und der darauf aufbauenden optischen Abbildung wird das verwendete Lochkameramodell vorgestellt. Nach der Analyse der Grundkomponenten wird das übergreifende Simulationskonzept vorgestellt und die Berechnung der theoretischen Antwort eines Pixels systemtheoretisch betrachtet und hergeleitet. Zur Nachbildung der systembedingten physikalischen Effekte einer PMD-Kamera wird das Konzept um weitere Modellierungsansätze ausgeweitet.

Gegenstand des nachfolgenden Kapitels ist die „*Sichtbarkeitsanalyse*“. Mit der Sichtbarkeitsanalyse wird berechnet, welche Bereiche einer digital vorliegenden 3D Szene von einem definierten Punkt aus sichtbar sind. Die Kenntnis dieser Bereiche ist für nachfolgende Bestimmung der möglichen Signalausbreitung sowie bei der Berechnung der synthetischen Sensordaten von zentraler Bedeutung. Für unterschiedliche Datenformen wurden zwei Berechnungsmethoden entwickelt und in dem *Visibility Check Module* implementiert.

Das Kapitel „*Linsenverzeichnung*“ beschäftigt sich mit der Fragestellung der Modellierung der klassischen Linsenverzerrungen. Im Anschluss behandelt das Kapitel

„*Simulatorarchitektur*“ die Entwicklung, die Aufgaben und das Zusammenwirken der wichtigsten Teilkomponenten. Es gewährt einen tieferen Einblick in die wichtigsten Schritte eines Simulationsprozesses. Im darauffolgenden Kapitel wird anhand einiger „*Beispielszenarien*“ ein Vergleich mit realen Daten vorgenommen.

Eine abschließende Zusammenfassung der Ergebnisse erfolgt in Kapitel 10. Dabei wird ein kurzer Ausblick auf zukünftige Arbeiten gegeben.

2 PMD Grundlagen

Im nachfolgenden Kapitel werden die Grundlagen der PMD-Technik vorgestellt. Es werden nur die für die Arbeit wichtigen Zusammenhänge und Aspekte betrachtet. Einen tiefergehenden Einblick in das Thema PMD vermitteln die zahlreichen unter Literatur aufgeführten Arbeiten.

2.1 Laufzeitverfahren Time-of-Flight

Bei Laufzeitverfahren wird die Entfernung anhand der Zeit bestimmt, die das vom Sender ausgesandte Licht für die Strecke Sender-Objekt-Empfänger braucht. Dabei wird davon ausgegangen, dass der Sender und der Empfänger dicht beieinander liegen. Die Laufzeit kann entweder direkt gemessen (z.B. mit Hilfe von Korrelationsverfahren) oder aus der Phasenverschiebung bestimmt werden, die das ausgesandte modulierte Signal auf seinem Ausbreitungsweg erfahren hat. Für die Laufzeit im Vakuum folgt dann:

$$\tau_L = \frac{2R}{c_0} \text{ bzw. } R = \frac{\tau_L \cdot c_0}{2} \quad (2.1)$$

$c_0 = 299.792.458 \text{ m/s}$ – Lichtgeschwindigkeit im Vakuum

R – Abstand Sender-Empfänger

Es wird ersichtlich, dass bei Pulsmessverfahren basierenden Messungen mit einer Auflösung von einem cm sehr hohe Anforderungen an das Empfängermodul gestellt werden. Die empfangenen Pulse müssen im Pikosekundenbereich ausgewertet werden.

Alternativ dazu kann ein kontinuierlich amplitudenmoduliertes Licht als Trägersignal ausgesendet werden. Aus der Phasenverschiebung des gesendeten und empfangenen Signals lässt sich die Entfernung zum Objekt bestimmen. Sei T_m – die Periode des amplitudenmodulierten Sendesignals, dann folgt für die Phasenverschiebung nach der Laufzeit τ_L :

$$\tau_L = T_m \cdot \frac{\varphi_d}{2\pi} \quad (2.2)$$

mit φ_d – Phasenverschiebung. Setzt man die Laufzeit in die Gleichung (2.1) ein, so erhält man für den gesuchten Abstand:

$$R = \frac{c_0 T_m}{2} \cdot \frac{\varphi_d}{2\pi} \quad (2.3)$$

D.h. die Phasenverschiebung ist direkt proportional zur Entfernung. Das Problem bei dieser Methode ist, dass die Phasenmessung nur innerhalb einer Periode eindeutig ist, denn nach einer Periode 2π besitzt das Signal die gleiche Phasenverschiebung

$$\frac{c_0 T_m}{2} \cdot \frac{\varphi_d \pm 2\pi}{2\pi} = \frac{c_0 T_m}{2} \cdot \frac{\varphi_d}{2\pi}$$

und der Abstand kann nicht eindeutig bestimmt werden, bzw. nur mit Hilfe von Phase-Unwrapping Algorithmen. Der Eindeutigkeitsbereich hängt somit von der Periode des modulierten Signals wie folgt ab:

$$0 < R < \frac{c_0 T_m}{2} \quad (2.4)$$

Die Tabelle 2.1 zeigt beispielhaft die Eindeutigkeitsbereiche für drei unterschiedliche Modulationsfrequenzen.

<i>Modulationsperiode</i>	<i>Modulationsfrequenz</i>	<i>Eindeutigkeitsbereich</i>
1000 ns	1 MHz	150 m
100 ns	10 MHz	15 m
10 ns	100 MHz	1,5 m

Tabelle 2.1: Eindeutigkeitsbereiche in Abhängigkeit von der Modulationsfrequenz

Das Messprinzip der optisch inkohärenten Phasenlaufzeitmessung ist in der Abbildung 2.1 dargestellt.

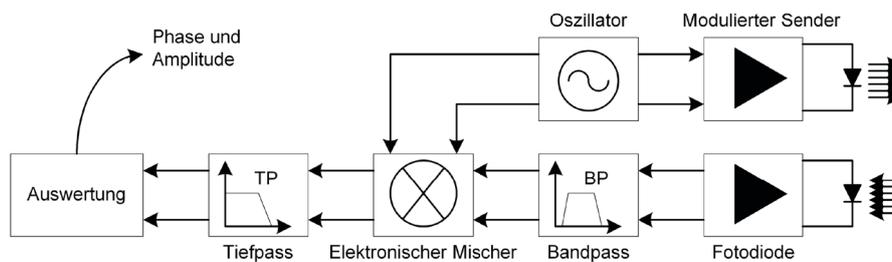


Abbildung 2.1: Prinzip des konventionellen Modulationsinterferometrieverfahrens ohne Referenzkanal [Schn 03]

Ein Sender strahlt das amplitudenmodulierte Lichtsignal in die zu untersuchende Szene aus. Dieses wird von ihr reflektiert und das reflektierte Licht von einer Fotodiode erfasst und anschließend gefiltert. Die Modulationsfrequenz muss dabei im Hochfrequenzbereich liegen damit geeignete Interferogramme mit Wellenlängen im Meterbereich dargestellt werden können [Schn 03]. Das empfangene gefilterte Messsignal, welches gegenüber dem Sendesignal phasenverzögert² ist, wird anschließend in einem Mischer mit dem ursprünglichen Oszillatorsignal gemischt. Das Ergebnis des Mischprozesses ist die

² Genau genommen ist das empfangene Signal nicht nur phasenverzögert. Die Amplitude des Signals fällt mit der Entfernung ebenfalls ab und es ist verrauscht.

Korrelation zwischen dem gesendeten und dem empfangenen Signal, aus der Informationen zur Entfernung abgeleitet werden können.

Der konventionelle Aufbau dieser Messmethode beinhaltet mit dem Sender, der Photodiode, dem Bandpass, dem breitbandigen Verstärker, dem elektrischen Mischer und dem Tiefpass eine Reihe von Komponenten, die nicht nur wegen der aufwendigen hochfrequenten Signalverarbeitung zu einer kostspieligen Lösung führen, sondern auch in der Summe, wie auch einzeln, störanfällig sind. Außerdem wird die Möglichkeit eingeschränkt, mehrere Punkte gleichzeitig zu messen.

Eine Alternative bietet die Integration der eingangsseitigen Komponenten in einen speziell entwickelten opto-digitalen Baustein, bei dem während der Detektion eines einfallenden intensitätsmodulierten Lichtsignals die Mischung des Fotostroms mit einem elektrischen Gegentaktsignal (Referenzsignal) direkt im fotoempfindlichen Bereich durchgeführt wird [Zha 03]. Einen solchen Sensor stellt der an der Universität Siegen entwickelte PMD-Sensor dar, dessen Funktionsweise im folgenden Kapitel näher betrachtet werden soll.

2.2 Aufbau und Funktionsweise des Photogate-PMD

Ein PMD Sensor (Photonic Mixer Device) ist, wie der Name schon sagt, nicht nur ein Photodetektor, sondern zugleich ein optoelektronischer Mischer. Er vereint in sich als integriertes Bauelement mehrere Funktionen des konventionellen Modulationsinterferometrieverfahrens. Die Beschreibung des Grundprinzips findet sich in den Publikationen [Schw 96] und [SwXu 96]. In Abbildung 2.2 ist der schematische Aufbau eines PMD Pixels dargestellt. Sie zeigt die Aufsicht und den Querschnitt durch eine PMD-Zelle sowie den durch die Modulation mit dem Referenzsignal erzeugten Potentialverlauf im aktiven Bereich des Sensorelementes. Ein PMD-Pixel besteht aus zwei symmetrisch angeordneten sich im lichtempfindlichen Detektorbereich befindlichen transparenten Photogates am und bm (auch Modulationsgates genannt). In direkter Nachbarschaft befindet sich jeweils eine Auslesediode, welche als Schnittstelle zur nachgeschalteten pixelzugehörigen Ausleseschaltung gilt [Hein 01]. Die beiden Photogates sind mit der Vorspannung U_0 und einer Gegentakt-Modulationsspannung $\pm u_m(t)$ vorgespannt. Auf diese Weise entsteht unter den lichtempfindlichen Photogates im Halbleiter ein zusätzliches Potentialgefälle, wodurch je nach Polarität der Modulationsspannung $u_m(t)$ die von dem einfallenden Licht generierten Ladungsträger (Elektronen) abwechselnd in den einen oder anderen Potentialtopf (Speicher) der Auslesedioden transportiert werden. Dieser sich entsprechend der Modulationsspannung zeitlich ändernde Verlauf des Potentialgefälles wird als „Elektronenschaukel“ bezeichnet.

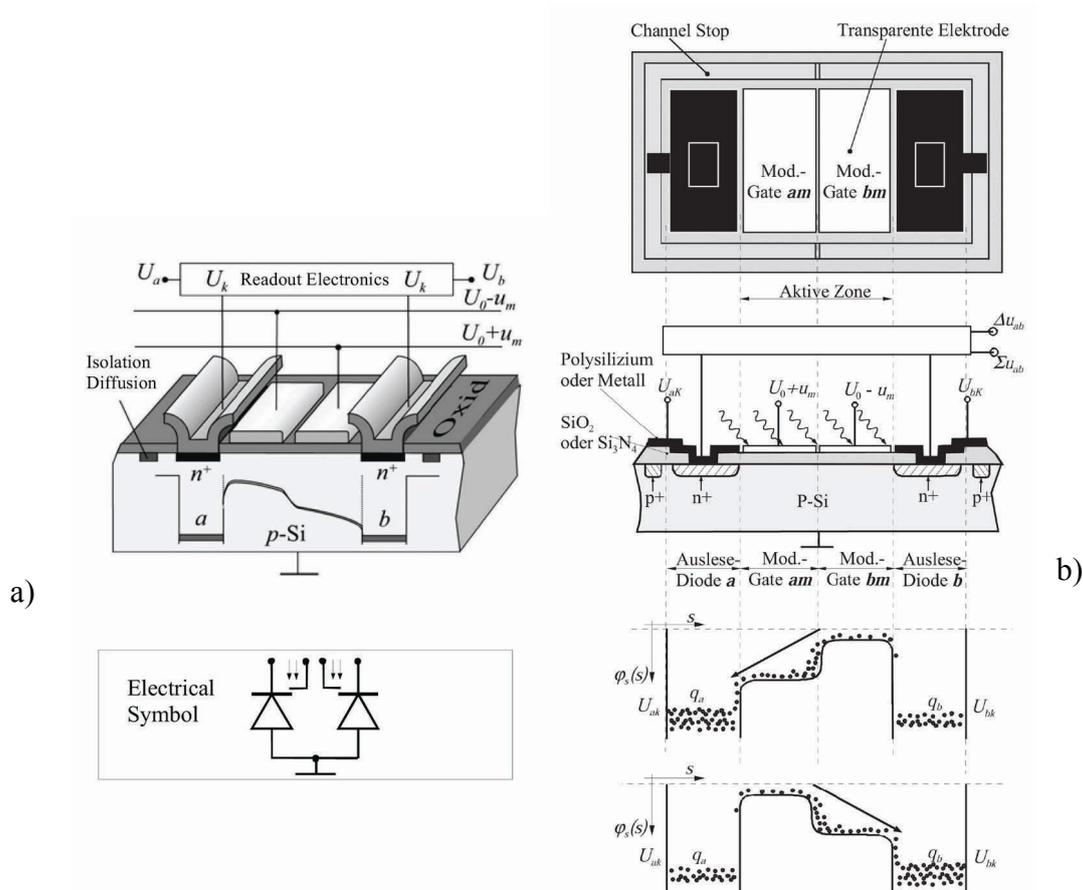


Abbildung 2.2: a) Schematischer Aufbau eines PMD-Pixels [Luan 01] und das elektrische Symbol eines PMD-Sensors, b) schematische Darstellung eines Schnittes durch eine PMD-Zelle und Veranschaulichung des Potentialverlaufs in Abhängigkeit von der angelegten Modulationsspannung

Sie bewirkt den gewünschten Mischeffekt, wenn die Modulation und das einfallende modulierte Licht einen definierten Phasenzusammenhang aufweisen. Die Ladungsträger wandern dann gezielt in eine Richtung. Eine phasenneutrale Hintergrundbeleuchtung (unmoduliertes Licht) führt zur wechselnden aber auf beiden Seiten gleichen Erhöhung der Ladung in beiden Potentialtöpfen. Es findet also eine korrelierte Ladungsträgerseparation statt.

Nun wird die Entfernungsmessung mittels PMD unter Ausnutzung der Phasenmessung und unter Annahme einer CW-Modulation signaltheoretisch betrachtet. Das Sendesignal $s(t)$ sei eine periodische Kosinus bzw. Sinusschwingung mit der Grundfrequenz ω_0 , der Amplitude A und dem Gleichanteil A_0 . Es wird angenommen, dass das empfangene Signal $g(t)$ mit dem Sendesignal bis auf die zeitliche Verschiebung τ_d bzw. den Phasenunterschied φ_d identisch ist, es gilt also:

$$\begin{aligned} s(t) &= A_0 + A \cdot \cos(\omega_0 t - \varphi_1) \\ g(t) &= A_0 + A \cdot \cos(\omega_0 t - \varphi_2) \end{aligned} \quad (2.5)$$

mit $\varphi_2 = \varphi_1 + \varphi_d$

Die Kreuzkorrelation zweier beliebiger Leistungssignale bzw. periodischer Signale ist mathematisch wie folgt definiert:

$$k_{sg}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T s(t) \cdot g(t + \tau) dt \quad (2.6)$$

Setzt man nun die Signale $s(t)$ und $g(t)$ in die Gleichung ein, dann erhält man:

$$k_{sg}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T [A_0 + A \cdot \cos(\omega_0 t - \varphi_1)] \cdot [A_0 + A \cdot \cos(\omega_0 (t + \tau) - \varphi_2)] dt \quad (2.7)$$

Mit $\cos(\alpha + \beta) = \cos(\alpha) \cdot \cos(\beta) - \sin(\alpha) \cdot \sin(\beta)$

und $\alpha = \omega_0 t - \varphi_2$ und $\beta = \omega_0 \tau$ folgt für die Kreuzkorrelation:

$$\begin{aligned} k_{sg}(\tau) &= \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T [A_0 + A \cdot \cos(\omega_0 t - \varphi_1)] \cdot [A_0 + A \cdot \cos(\omega_0 t - \varphi_2) \cdot \cos(\omega_0 \tau) \\ &\quad - A \cdot \sin(\omega_0 t - \varphi_2) \cdot \sin(\omega_0 \tau)] dt \\ &= \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \left[A_0^2 + \underbrace{AA_0 \cdot \cos(\omega_0 t - \varphi_1)}_{\text{Das Teilintegral verschwindet}} + \underbrace{AA_0 \cdot \cos(\omega_0 t - \varphi_2) \cdot \cos(\omega_0 \tau)}_{\text{Das Teilintegral verschwindet}} + \right. \\ &\quad + A^2 \cdot \cos(\omega_0 t - \varphi_1) \cdot \cos(\omega_0 t - \varphi_2) \cdot \cos(\omega_0 \tau) \\ &\quad - \underbrace{AA_0 \cdot \sin(\omega_0 t - \varphi_2) \cdot \sin(\omega_0 \tau)}_{\text{Das Teilintegral verschwindet}} \\ &\quad \left. - A^2 \cdot \cos(\omega_0 t - \varphi_1) \cdot \sin(\omega_0 t - \varphi_2) \cdot \sin(\omega_0 \tau) \right] dt \end{aligned} \quad (2.8)$$

Das zweite, dritte und fünfte Teilintegral liefern keinen Beitrag aufgrund der Integration einer harmonischen Funktion über eine ganze Periode.

Unter Ausnutzung von

$\cos(\alpha) \cdot \cos(\beta) - \sin(\alpha) \cdot \sin(\beta) = \cos(\alpha + \beta)$ lässt sich schreiben:

$$\begin{aligned}
k_{sg}(\tau) &= \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \left[A_0^2 + A^2 \cdot \cos(\omega_0 t - \varphi_1) \cdot \left\{ \cos(\omega_0 t - \varphi_2) \cdot \cos(\omega_0 \tau) \right. \right. \\
&\quad \left. \left. - \sin(\omega_0 t - \varphi_2) \cdot \sin(\omega_0 \tau) \right\} \right] dt \\
&= \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \left[A_0^2 + A^2 \cdot \cos(\omega_0 t - \varphi_1) \cdot \cos(\omega_0 t - \varphi_2 + \omega_0 \tau) \right] dt
\end{aligned} \tag{2.9}$$

Und nach einer weiteren Umformung mit

$$\cos(\alpha) \cdot \cos(\beta) = \frac{1}{2} (\cos(\alpha - \beta) + \cos(\alpha + \beta))$$

$$\text{und } \alpha = \omega_0 t - \varphi_1 \quad \text{und } \beta = \omega_0 t - \varphi_2 + \omega_0 \tau \quad \Rightarrow$$

$$\begin{aligned}
k_{sg}(\tau) &= \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \left[A_0^2 + A^2 \cdot \frac{1}{2} \cos(\omega_0 t - \varphi_1 - \omega_0 t + \varphi_2 - \omega_0 \tau) + \right. \\
&\quad \left. A^2 \cdot \frac{1}{2} \cos(\omega_0 t - \varphi_1 + \omega_0 t - \varphi_2 + \omega_0 \tau) \right] dt = \\
&= \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \left\{ A_0^2 + \frac{A^2}{2} \cos(\omega_0 \tau + \varphi_1 - \varphi_2) \right. \\
&\quad \left. + \underbrace{\frac{A^2}{2} \cos(2\omega_0 t - \varphi_1 - \varphi_2 + \omega_0 \tau)}_{\text{Das Teilintegral verschwindet}} \right\} dt
\end{aligned} \tag{2.10}$$

Die Kreuzkorrelationsfunktion des Sendesignals mit dem Empfangssignal lässt sich also wie folgt bestimmen:

$$\begin{aligned}
k_{sg}(\tau) &= \lim_{T \rightarrow \infty} \frac{1}{2T} \cdot 2T \cdot \left[A_0^2 + \frac{A^2}{2} \cos(\omega_0 \tau + \varphi_1 - \varphi_2) \right] = A_0^2 + \frac{A^2}{2} \cos(\omega_0 \tau - \varphi_d) \\
&= A_0^2 + \frac{A^2}{2} \cos(\omega_0 (\tau - \tau_d))
\end{aligned} \tag{2.11}$$

$$\text{mit } \varphi_d = \varphi_2 - \varphi_1 = \omega_0 \tau_d$$

Gleichung (2.11) erhält drei Unbekannte – A_0 , A und φ_d bzw. τ_d . Um sie bestimmen zu können, müssen mindestens drei verschiedene Messungen der KKF zu unterschiedlichen Verschiebungszeiten τ durchgeführt werden. Üblicherweise werden in einer PMD-Kamera bei jeder Messung zwei Messdaten aus den beiden Kanälen geliefert, die aufgrund der Gegentaktmodulation zwei Messpunkten mit einem Intervall von $\Delta \tau = \frac{T_0}{4}$

entsprechen [Zha 03], mit $T_0 = \frac{2\pi}{\omega_0}$. Werden die vier Messpunkte äquidistant gewählt,

z.B. bei 0° , 90° , 180° und 270° , dann lassen sich die spezifischen Größen der Korrelationsfunktion wie folgt bestimmen:

$$\begin{aligned}
 1) \quad \tau_1 = 0 & \Rightarrow k_{sg}(\tau_1) = A_0 + \frac{A^2}{2} \cos(\varphi_d) \\
 2) \quad \tau_2 = \frac{T_0}{4} & \Rightarrow k_{sg}(\tau_2) = A_0 + \frac{A^2}{2} \sin(\varphi_d) \\
 3) \quad \tau_3 = \frac{T_0}{2} & \Rightarrow k_{sg}(\tau_3) = A_0 - \frac{A^2}{2} \cos(\varphi_d) \\
 4) \quad \tau_4 = \frac{3T_0}{4} & \Rightarrow k_{sg}(\tau_4) = A_0 - \frac{A^2}{2} \sin(\varphi_d)
 \end{aligned} \tag{2.12}$$

Aus der Summe der vier Gleichungen lässt sich A_0 bestimmen:

$$A_0^2 = \frac{1}{4} (k_{sg}(\tau_1) + k_{sg}(\tau_2) + k_{sg}(\tau_3) + k_{sg}(\tau_4)) \tag{2.13}$$

Subtrahiert man von der ersten Gleichung die dritte und von der zweiten die vierte, so lässt sich nach Quadrieren und anschließendem Summieren der beiden Ausdrücke A und durch Dividieren die Differenzphase φ_d bestimmen:

$$A^2 = \sqrt{[k_{sg}(\tau_1) - k_{sg}(\tau_3)]^2 + [k_{sg}(\tau_2) - k_{sg}(\tau_4)]^2} \tag{2.14}$$

$$\varphi_d = \arctan\left(\frac{k_{sg}(\tau_2) - k_{sg}(\tau_4)}{k_{sg}(\tau_1) - k_{sg}(\tau_3)}\right) \tag{2.15}$$

In Abbildung 2.3 sind die entsprechenden Sende- und Empfangssignale mit ihrer Korrelationsfunktion beispielhaft dargestellt (mit $\varphi_d = 0$ und $\tau_d = T_0/6$).

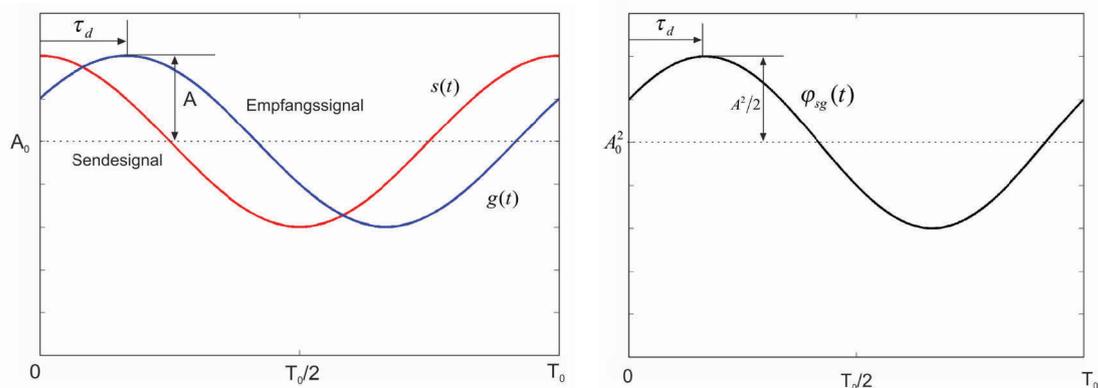


Abbildung 2.3: Verlauf eines Sende- und Empfangssignals (links) und ihre Korrelationsfunktion (rechts)

Mit den in (2.5)-(2.15) erzielten Ergebnissen lässt sich somit nachweisen, dass die Korrelation als Ergebnis des Mischprozesses in einem Zusammenhang mit der zu bestimmenden Entfernung steht.

Betrachtet man die Strahlungsleistung auf einer Fotodiode oder einem Photogate allgemein, dann setzt sich diese aus einem durch die Lichtausbreitung in der Amplitude veränderten, harmonisch modulierten Sendesignal einerseits, und aus dem zusätzlich einfallenden Hintergrundlicht andererseits zusammen [Schn 03].

$$\Phi(t, \varphi) = G_0 + R \cdot \sin(\omega_0 t + \varphi_d) \quad (2.16)$$

mit φ_d – entfernungsabhängige Phasenverschiebung
 G_0 – unmodulierter Gleichanteil
 R – Remissionsamplitude

Wird nun das Demodulationssignal mit der Vorspannung U_0 und der Gegentakt-Modulationsspannung $\pm u_m(t)$ betrachtet,

$$s_{a,b}(t, \psi) = U_0 \pm u_m(t) = U_0 \pm U \cdot \sin(\omega_0 t + \psi) \quad (2.17)$$

dann lässt sich mit der bereits oben durchgeführten Rechnung folgender Zusammenhang für die Korrelation der beiden Funktionen herleiten:

$$\begin{aligned} k_a(\varphi_d, \psi) &= H + M \cos(\varphi_d - \psi) \\ k_b(\varphi_d, \psi) &= H - M \cos(\varphi_d - \psi) \end{aligned} \quad (2.18)$$

mit dem nach [Schn 03] modulierten Signalanteil

$$M = f(U, R, n, T_p),$$

dem unmodulierten Hintergrundanteil

$$H = f(U_0, G_0, n, T_p),$$

der wählbaren Phasenverschiebung zwischen Modulation und Demodulation ψ und mit der Entfernung R zum Objekt zusammenhängenden Phasenverschiebung $\varphi_d = \frac{2\omega_0 R}{c}$.

Werden nun die Gegentaktmodulationssignale im symmetrischen „Lock-in“-Verfahren zu den zwei Zeitpunkten bei 0° Inphase und 90° Quadraturphase gemessen, dann lässt sich daraus die Phasenverzögerung analog zu (2.15) bestimmen:

$$\varphi_d = \arctan\left(\frac{k_2 - k_4}{k_1 - k_3}\right) \quad (2.19)$$

mit den vier Korrelationssignalen

$$\begin{aligned}
k_1(\varphi_d, 0) &= H + M \cos(\varphi_d) \\
k_3(\varphi_d, 0) &= H - M \cos(\varphi_d) \\
k_2\left(\varphi_d, \frac{\pi}{2}\right) &= H + M \cos\left(\varphi_d - \frac{\pi}{2}\right) = H + M \sin(\varphi_d) \\
k_4\left(\varphi_d, \frac{\pi}{2}\right) &= H - M \cos\left(\varphi_d - \frac{\pi}{2}\right) = H - M \sin(\varphi_d)
\end{aligned} \tag{2.20}$$

Durch Summieren der vier Korrelationswerte erhält man den Offset der Korrelationsfunktion, welcher dem Grauwert jedes Pixels entspricht.

$$H = \frac{1}{4}(k_1 + k_2 + k_3 + k_4) \tag{2.21}$$

Analog zu (2.14) folgt für Amplitude der Korrelationssignale:

$$M = \frac{1}{2} \sqrt{(k_1 - k_3)^2 + (k_2 - k_4)^2} \tag{2.22}$$

Diese repräsentiert die Qualität des gemessenen Signals, also die Aussage darüber wie sicher aus den einzelnen Abtastwerten der Korrelationsfunktion die Phasenverschiebung und der entsprechende Entfernungswert bestimmt werden kann.

Betrachtet man einen realen Verlauf der Korrelationsfunktion, dann ist dieser durch viele physikalische Störungen und Einflüsse beeinträchtigt, wie z.B. thermisches Rauschen, Schrotrauschen, Fixed Pattern Noise, Abtastdauer und nicht zuletzt infolge der digitalen Verarbeitung durch die interne Quantisierung bei der Signalverarbeitung.

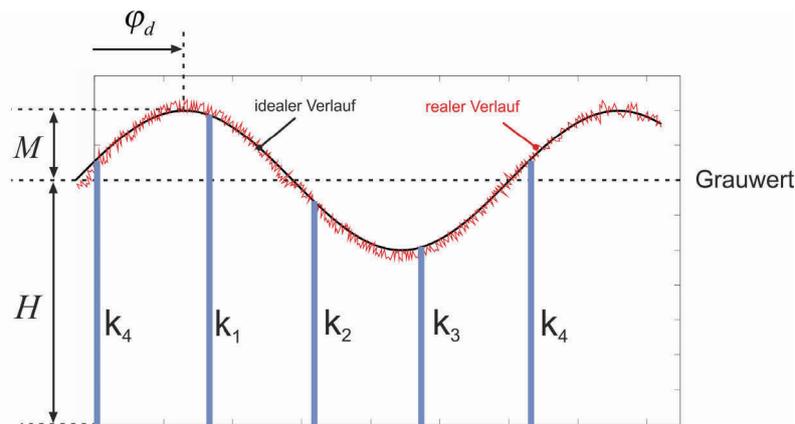


Abbildung 2.4: Abtastung der Korrelationsfunktion für die Bestimmung der Phasenverschiebung

Werden diese Störungen in der Summe zu stark bzw. wird das empfangene Signal im Verhältnis dazu zu schwach, dann nimmt mit dem sinkenden Signal-zu-Rausch-Verhältnis auch die Genauigkeit der Messungen ab. Das Verhältnis der Amplitude des Korrelationssignals zum Grauwert ist also maßgeblich für die Qualität der Messung.

Der eigentliche Entfernungswert ergibt sich dann aus der Phasenverschiebung zu:

$$R = \frac{c_0}{2 \cdot f_{\text{mod}}} \cdot \left(\frac{\varphi_d}{2\pi} + N \cdot 2\pi \right), \text{ mit } N = 0, 1, 2, 3, \dots \quad (2.23)$$

Um den Eindeutigkeitsbereich nicht zu verlassen, muss bei der Messung der einzelnen Korrelationssignale darauf geachtet werden, dass die Modulationsfrequenz entsprechend so gewählt wird, dass alle Messungen innerhalb einer Periode durchgeführt werden.

2.3 Überführung in eine PMD-Kamera

Der in Kapitel 2.2 vorgestellte Photogate-PMD Sensor bietet dank seiner technisch günstig zu realisierenden Funktionsweise nicht nur eine kostengünstige Alternative zu konventionellen Modulationsinterferometrierverfahren, sondern aufgrund seiner integrierten Form auch die Möglichkeit, ähnlich wie bei digitalen Kameras, kostengünstig eine Matrix von Sensoren als Pixel kompakt nebeneinander unterzubringen. Dieses Konzept kann vollständig in einer Standard-CMOS-Technologie realisiert werden, wodurch neben den enormen Kostenvorteilen noch breitere technische Möglichkeiten zur Verfügung stehen, z.B. Implementierung weiterer integrierter Schaltungen und Komponenten, wie Ausleseelektronik, Filter oder der patentierten SBI-Schaltung³ direkt „on chip“. Die letztere ermöglicht weitgehend die Unterdrückung des Hintergrundlichtes. Das Hintergrundlicht hat dann prinzipiell keinen Einfluss auf die Spannungsdifferenz der beiden Auslesedioden. Ein Problem entsteht, wenn die Ausleseschaltung schnell in die Sättigung geht. Bei hellem Umgebungslicht ist das Verhältnis des eingefangenen modulierten Lichts zum Hintergrundlicht besonders bei größeren Entfernungen zum Messobjekt sehr klein. D.h. nur ein geringer Teil der vom einfallenden Licht generierten Ladungen korrespondiert mit dem gesendeten Signal, was eine Verschlechterung der Dynamik des Sensors und die damit verbundene schlechtere Entfernungsauflösung zur Folge hat. Mit einer zusätzlichen aktiven Schaltung auf dem Chip kann die Sättigung durch Einwirkung von Fremdlicht minimiert werden. Auf diese Weise erreichen aktuelle PMD-Sensoren mit SBI eine Dynamik von > 120 dB.

Ein weiterer Vorteil des in dem vorhergehenden Kapitel vorgestellten Konzeptes besteht in der Möglichkeit neben den reinen Tiefeninformationen den Grauwert des gesamten detektierten Lichtes, sowie die Amplitude der Korrelationssignale – auch Signalstärke oder einfach Amplitude genannt – als Qualitätsmaß jeder einzelnen Messung zu erhalten. Unter Ausnutzung dieser Amplitudenwerte können z.B. in nachgeschalteten Bildverarbeitungsalgorithmen stark gestörte Pixel herausmarkiert bzw. „stabilisiert“ werden.

Mit dem für jedes Pixel vorliegenden Entfernungs- und Grauwert und der bekannten Abbildung über die gemeinsame Optik kann die Umgebung im Kamerakoordinatensys-

³ SBI – Suppression of Background Illumination

tem dreidimensional abgetastet werden. Dabei wird die zu vermessende Szene mit inkohärentem, moduliertem Licht bestrahlt. Dieses wird von einem oder mehreren, meistens aus vielen einzelnen LED's bestehenden, Beleuchtern erzeugt. Das reflektierte Licht wird gleichzeitig von allen Pixeln aufgenommen. Je nach eingestellter Integrationszeit (vergleichbar mit der Belichtungszeit bei einer CCD Kamera) können dabei mit aktuellen PMD-Kameras Bildraten von bis zu 100 Frames/Sek erzielt werden. Abbildung 2.5 zeigt zwei reale Aufnahmen einer PMD Kamera.

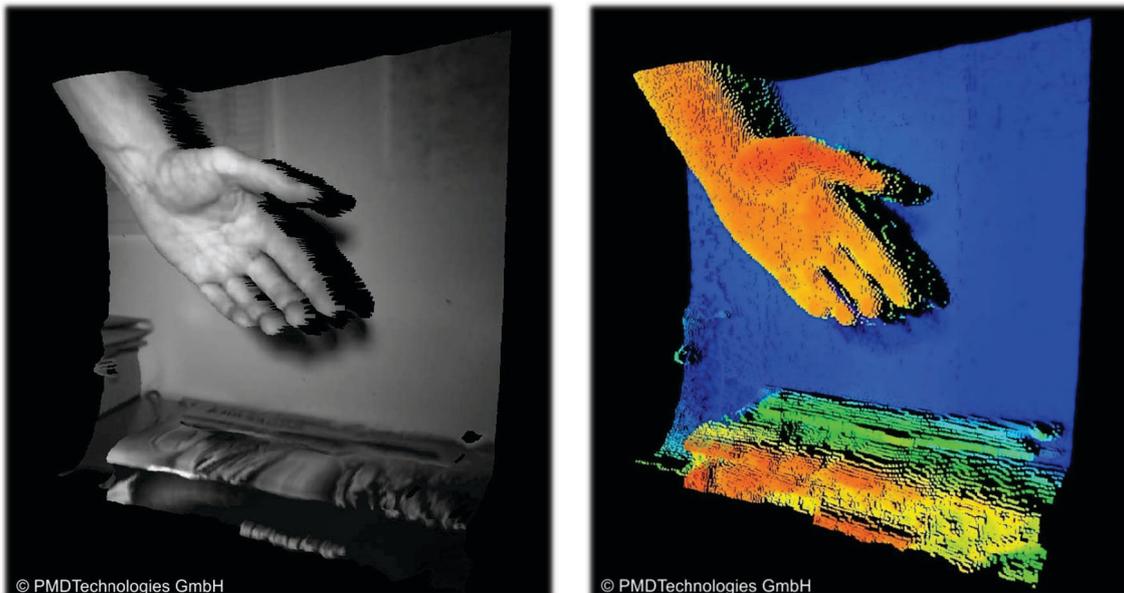


Abbildung 2.5: Aufnahmen einer realen PMD-Kamera, links – Grauwertbild, rechts – farbkodiertes Entfernungsbild

Quelle: © PMDTechnologies GmbH

2.4 Aufbau einer PMD-Kamera

Aufgrund unterschiedlicher Anforderungen und kontinuierlich fortschreitender Entwicklungen gibt es einige Ausführungen von PMD-Systemen. Alle dieser Ausführungen bestehen prinzipiell aus folgenden Komponenten:

- Aktive(r) Beleuchter

PMD Kameras sind selbstbeleuchtend. Das für die Messung benötigte modulierte Licht wird über die interne Ansteuerung mittels FPGA von den Beleuchtern erzeugt. Aufgrund der recht hohen benötigten Lichtstrahlung wird das Licht üblicherweise entweder mit Hilfe von Hochleistungs-LED's oder durch eine Reihe von LED's erzeugt. Üblicherweise wird die Beleuchtung im für das menschliche Auge unsichtbaren Infrarotbereich realisiert.

- PMD-Sensor und Beschaltung

Der PMD-Sensor – das Herzstück einer PMD-Kamera – wird mit der notwendigen Beschaltung und Schnittstellen meistens in einem eigenen Gehäuse untergebracht.

- Optik

Die verwendete Optik bestimmt im Wesentlichen die Abbildungseigenschaften und ist bei vielen Ausführungen austauschbar.



Abbildung 2.6: Aufbau verschiedener PMD-Kameras, links – PMD[vision]® CamCube, rechts – ZESS 2D/3D MultiCam, unten – PMD 41k-S mit parabolischen Reflektoren zur Erhöhung der Reichweite

Abbildung 2.6 zeigt unterschiedliche Ausführungen von PMD-Systemen. Links im Bild ist die PMD[vision]® CamCube mit zwei am Kameragehäuse seitlich angebrachten Beleuchtern dargestellt. Das rechte Bild zeigt die im Zentrum für Sensorsysteme entwickelte MultiCam. Die in einem gemeinsamen Gehäuse untergebrachten 2D CMOS Sensor und ein 3D PMD Sensor, die über eine gemeinsame Optik betrieben werden, vereinen die Vorteile der beiden Systeme. Unten im Bild ist der Aufbau eines Projektes zur Erhöhung der Reichweite der Beleuchter mittels parabolischen Reflektoren abgebildet.



2.5 Leistungsgrenzen eines PMD-Systems

Wie bei allen anderen optischen Entfernungsmesssystemen, wird die Messgenauigkeit eines PMD-Systems in erster Linie durch das System selbst (hardwareseitige Realisierung), und in zweiter Linie durch die Menge des einfallenden modulierten Lichtes (physikalische Gegebenheiten) beeinflusst. Zu wenig einfallendes Licht bewirkt starkes Rauschen, bei zu viel Licht kommt es zur Übersättigung, welche die Messunsicherheit

ebenfalls stark erhöht. Daher gibt es keine PMD-Lösungen, die in allen Bereichen zu besten Ergebnissen führen, oft muss ein PMD-System an eine Anwendung angepasst werden um den besten Kompromiss zwischen den gestellten Anforderungen und der Realisierbarkeit zu finden.

Betrachtet man die hardwareseitige Seite, so hängt, wie bereits erwähnt, die erzielbare Tiefengenauigkeit primär von der Menge des einfallenden Lichtes ab. Diese hängt wiederum, neben der Messentfernung, einerseits mit der Größe der lichtempfindlichen Pixel und andererseits mit der Strahlungsstärke des eingesetzten Beleuchters zusammen. Mit oft gewünschter Erhöhung der lateralen Auflösung des Chips sinkt oft mit steigender Pixeldichte gleichzeitig die Größe und somit die Empfindlichkeit jedes Pixels. Typische Pixelgrößen aktueller 2D CCD Kameras liegen im Bereich einiger μm , die der aktuellen 3D ToF-Kameras – im Bereich 40-500 μm . Bei diesen Pixelgrößen würde ein 3D Megapixel-Chip bereits einige 10 cm^2 groß sein, was nicht nur hohe Herstellungskosten (auch nicht zuletzt wegen der teureren Optik) mit sich brächte, sondern auch die Handlichkeit und Verwendbarkeit einschränken würde. Mögliche Ansätze wären hier:

- Erhöhung der Lichtempfindlichkeit des Sensors
- Niedrigeres Eigenrauschen des Systems durch Optimierung des Herstellungsprozesses und Ausnutzung neuer Technologierkenntnisse
- Erhöhung der Effizienz und Lichtstärke der Beleuchtung (z.B. durch Kombination mehrerer Beleuchter, Einsatz lichtstarker Beleuchter, Bündelung der Lichtkeulen usw.)
- Verbesserung der Genauigkeit durch Einsetzen optimierter Bilderzeugungs- und Bildverarbeitungsalgorithmen.

Unterschiedliche Objekte und Materialien reflektieren das Licht unterschiedlich stark. Während glatte metallische Oberflächen für elektromagnetische Strahlung hohe Reflektivität aufweisen und die einfallende Strahlung in eine bevorzugte Richtung wieder zurückspiegeln, absorbieren raue Oberflächen die Strahlung meist viel stärker und geben sie diffus ab. Einige Materialien besitzen die Eigenschaft die Strahlung regelrecht komplett zu verschlucken. Eine Messung und ihre Genauigkeit hängen somit auch sehr stark von den physikalischen Gegebenheiten der zu vermessenden Szene ab.

Messunsicherheit eines PMD-Systems

Die Messunsicherheit dR eines PMD-Systems kann nach [Ring 07] und [Win 09] wie folgt berechnet werden

$$dR = \frac{1}{\sqrt{N_{\text{phase}}}} \cdot \frac{1}{k_{\text{tot}} \cdot \frac{S}{N}} \cdot \frac{\lambda_{\text{mod}}}{\pi\sqrt{8}} \quad (2.24)$$

mit

N_{phase} – Anzahl der Rohwert-Messungen bei sequentielltem „phase shift“

k_{tot} – Mischkontrast nach der Demodulation, der aus dem Produkt des (De)modulationskontrastes des PMD-Empfängers mit dem Modulationskontrast der aktiven Beleuchtung entsteht. Der maximale Wert für k_{tot} beträgt 1: $0 \leq k_{tot} \leq 1$

S – Anzahl der Signalelektronen des aktiven Lichts

N – Anzahl der Rauschelektronen (beinhaltet Schrottrauschen aller im PMD empfangenen, optisch und thermisch generierten Ladungsträger, Reset-, Verstärker- und Quantisierungsrauschen und ADC Rauschen)

λ_{mod} – Wellenlänge des Modulationssignals

Ist die Anzahl der Signalelektronen viel größer als die Anzahl der Rauschelektronen, dann tendiert die Messunsicherheit gegen Null: $S/N \rightarrow \infty \Rightarrow dR \rightarrow 0$. Dementsprechend sind große Entfernungen zum Objekt hin und Materialien, die schlecht reflektieren, ein grundsätzliches Problem für PMD-Systeme. Mit abnehmender Anzahl an Signalelektronen wird die Messunsicherheit größer.

Standardabweichung der Phasenverzögerung und Entfernungsstandardabweichung

Die Verteilung der Phasenverzögerung aufgrund des Poisson-Rauschens⁴ kann nach [Schn 03] wie folgt berechnet werden:

$$\sigma_{\varphi}^2 = \frac{H}{2M^2} \quad (2.25)$$

mit H – Grauwert eines Pixels und M – Amplitude der Korrelationssignale.

Für die daraus resultierende Entfernungsstandardabweichung folgt dann:

$$\sigma_R = \frac{\partial R}{\partial \varphi} \cdot \sigma_{\varphi} = \frac{c}{2\omega} \sqrt{\frac{H}{2M^2}} \quad (2.26)$$

Wie man aus den Gleichungen erkennen kann, ist die Anzahl der von dem modulierten Licht generierten Ladungen, also die Menge des Nutzlichtes auf dem Sensor, maßgeblich für die Entfernungsmessgenauigkeit.

⁴ Jedem Lichtsignal ist aufgrund der Photonenstatistik das Poisson-Rauschen überlagert. Treten innerhalb einer bestimmten Zeit im Mittel N Photonen auf, so beträgt die Standardabweichung $\sigma = \sqrt{N}$.

3 Beleuchtung – Modellierung und Simulation

Das nachfolgende Kapitel beschäftigt sich mit der Fragestellung realitätsnaher Modellierung und Simulation verschiedener Beleuchterausführungen und deren möglichen Kombinationen. Es befasst sich mit der Betrachtung der möglichen Lichtquellen und mit der Ausbreitung von Licht im Raum ausgehend von diesen Quellen. Für die Berechnung der Lichtausbreitung wurde im Rahmen dieser Arbeit ein eigenständiges Modul – das *Light Propagation Module* – entwickelt, welches dafür verantwortlich ist, die Lichtintensität für jeden einzelnen Punkt der Szene für jeden beliebigen Zeitpunkt fragementweise zu berechnen.

3.1 Aufgabenstellung der Lichtsimulationskomponente (*Light Propagation Module*)

Wie bereits erwähnt, besteht die Hauptaufgabe des *Light Propagation Module* in der Berechnung der Lichtverhältnisse an einem bestimmten Punkt im Raum zu einer bestimmten Zeit. Folgende Freiheitsgrade mussten dabei realisiert und folgende Probleme gelöst werden:

Freiheitsgrade für den Benutzer:

- Definition einer einzelnen Lichtquelle (Punktstrahler, LED usw.).
Um dem Anwender die Möglichkeit zu geben, seine Simulationen möglichst nahe an die realen Anforderungen anzupassen, sollte es möglich sein, die Charakteristika der einzelnen Lichtquellen vorzugeben.
- Konfiguration von Arbeitsparametern der Lichtquellen (Modulationsfrequenz, Betriebsart (an/aus) usw.).
- Kombination von einzelnen Lichtquellen zu einem Beleuchter.
Reale Beleuchter bestehen in der Regel aus einer festen Anordnung von Infrarot-LED's, die das modulierte Licht (idealerweise) phasengenau ausstrahlen. Desweiteren haben solche Beleuchter Komponenten gleicher Güte mit gleichen Spezifikationen. Die Anordnung kann somit als vordefiniert in die Beleuchter-Bibliothek aufgenommen und aufgerufen werden.
- Einsatz mehrerer frei definierbarer Beleuchter.
Es muss möglich sein, mehrere Beleuchter zu definieren. Die technischen Eigenschaften dieser Beleuchter können variieren.

- Frei veränderbare Position und die Orientierung der Beleuchter im Raum.

Im Hinblick auf die Untersuchungen der bi- bzw. multistatischen Einflüsse muss es möglich sein, vordefinierte Trajektorien der Beleuchter „abfahren“ zu lassen.

Theoretische und Simulationstechnische Herausforderungen:

- Betrachtung der Abschwächung der Lichtintensität zwischen der Lichtquelle und Punktziel.

Die Lichtintensität nimmt nicht nur mit dem steigenden Abstand zur Lichtquelle ab, die Rückstreuung des ausgesandten Lichtes hängt auch von weiteren Faktoren ab, wie Objekt- und Oberflächeneigenschaften, Einfallswinkel usw. Es muss ein Ansatz gewählt werden, der einerseits eine realitätsnahe Simulation mit Berücksichtigung der wichtigsten Faktoren ermöglicht, und andererseits die Simulationszeiten nicht unverhältnismäßig ansteigen lässt.

- Interaktion mit dem *Visibility Check Module*

Die Lichtsimulationskomponente muss mit dem *Visibility Check Module*, welches für die Berechnung der Schattenseiten einer Szene verantwortlich ist, aktiv interagieren. Die hier gewonnenen Informationen darüber, welche Bereiche der 3D-Szene beleuchtet werden können, müssen später einer weiteren Sichtbarkeitsüberprüfung aus der Sicht der Kamera unterworfen werden. Es ist sehr wichtig an dieser Stelle ein gutes Datenmanagement zu realisieren, da diese beiden Berechnungen den größten Beitrag an der gesamten Rechenzeit ausmachen.

- Optionale Betrachtung von möglichen Interferenzerscheinungen von Lichtwellen unterschiedlicher Lichtquellen beim Einsatz kohärenter Lichtquellen.

3.2 Physikalische Betrachtungen

Für eine realitätsnahe Modellierung müssen die Rückstreuungseigenschaften der beleuchteten Szene berücksichtigt werden. Betrachtet man eine Infrarotlichtfront, die von einem Punktstrahler (siehe 3.2.1) erzeugt wird, auf ihrem Ausbreitungsweg zum Objekt hin und zurück, so findet nicht nur eine Abnahme der Lichtintensität mit dem Quadrat des Abstandes statt. Trifft diese Front auf ein Objekt auf, so wird nur im Falle eines idealen Rückstrahlers die gesamte Energie wieder zurückgestrahlt. Ideale Rückstrahler wären interessant für eventuelle Kalibrierungen der Kamera oder Koregistrierung der Daten, wo man bestimmte Objekte durch höhere Helligkeitswerte in der Szene „markieren“ könnte. Im allgemeinen Fall wären Simulationen mit solchen Objekten aber realitätsfremd. Aus diesem Grund wurden im Simulator für den Anwender einige Möglichkeiten realisiert die Rückstreuungseigenschaften zu definieren. Zunächst betrachten wir jedoch den allgemeinen Fall der Lichtausbreitung.

Was passiert mit der Lichtstrahlung auf ihrem Ausbreitungsweg vom Beleuchter zum Objekt?

Im Laufe der letzten vier Jahrhunderte entwickelte man unterschiedliche Theorien, ob Licht Wellen- oder Teilchencharakter aufweist. Im 17. Jahrhundert kam Newton zu der Feststellung, dass einzelne Lichtstrahlen unveränderliche Eigenschaften haben, was ihn zu der Überzeugung brachte, dass Licht aus atomähnlichen Lichtteilchen bestünde. Allerdings waren mit dieser Theorie solche Phänomene wie Doppelbrechung oder Interferenz nicht erklärbar. Mit der Formulierung der Maxwell-Gleichungen Ende des 19. Jahrhunderts und des sich daraus ergebenden zusätzlichen elektromagnetischen Charakters des Lichtes wurde der Grundstein des modernen Wellen-Teilchen-Dualismus gelegt.

Der Teilchencharakter der Lichtwelle spielt im Bereich der lichtempfindlichen p-Siliziumschicht des PMD-Sensors eine entscheidende Rolle, wo durch die einfallenden Photonen Elektronen aus ihren Bindungen gelöst und somit Elektron-Loch Paare generiert werden. Bei der Ausbreitung des von den Beleuchtern erzeugten Lichtes spielen die elektromagnetischen Eigenschaften die primäre Rolle. Das Licht wird also in Simulationen als elektromagnetische Welle betrachtet.

3.2.1 Lichtquellen

In der Computergrafik unterscheidet man grob vier verschiedene Arten von Lichtquellen. Simulationen dieser Lichtquellen stellen unterschiedlich starke Anforderungen an die berechnende Hardware.

Umgebungslicht (ambient light)

Man spricht vom Umgebungslicht, wenn das Licht

- keiner Quellposition zugeordnet werden kann
- keine feste Ausbreitungsrichtung hat
- überall im Raum die gleiche Intensität aufweist

Solche Lichtquellen gibt es in der Natur nicht. Näherungsweise entsteht dieses Licht durch Reflektionen des Lichtes einer anderen Quelle an matten Oberflächen. Das Licht in einer weißen Papierkugel, die von außen von vielen Seiten angestrahlt wird, könnte man näherungsweise als Umgebungslicht beschreiben.

Gerichtetes Licht (directed light, infinite light)

Man spricht vom gerichtetem Licht, wenn das Licht

- keiner Quellposition zugeordnet werden kann

- eine feste Ausbreitungsrichtung hat
- überall im Raum die gleiche Intensität aufweist

Das Licht von weit entfernten Lichtquellen, wie das der Sterne oder Sonnenlicht, kann als gerichtetes Licht angesehen werden. Die Lichtstrahlen in großer Entfernung verlaufen dann näherungsweise parallel zueinander.

Punktlichtquelle (point light)

Man spricht von einer Punktlichtquelle, wenn das Licht

- einer Quellposition zugeordnet werden kann
- keine bevorzugte Ausbreitungsrichtung hat
- Anfangsintensität I_0 hat
- Intensität nimmt mit der Entfernung ab

Ist die Lichtquelle im Vergleich zum Objekt relativ klein und strahlt in alle Richtungen gleichmäßig ihr Licht ab, so kann die Quelle als Punktquelle betrachtet werden.

Strahler (spot light)

Man spricht von einem Strahler, wenn

- die Lichtquelle eine Position besitzt
- das Licht eine Lichtrichtung (Abstrahlwinkel) hat
- das Licht die Anfangsintensität I_0 hat
- die Lichtintensität mit der Entfernung abnimmt

Aufgrund der zu stark abnehmenden Lichtintensität der von Beleuchtern erzeugten Lichtstrahlen im Fernbereich beschränken sich die meisten relevanten PMD-Kamera-Anwendungen auf den Bereich bis zu 100 m (meistens deutlich weniger). Weitere Distanzen wären nur bei Einsatz stark gerichteter Beleuchtung möglich. D.h. für typische Beleuchter-Szene Interaktionen kommt die Betrachtung der ersten beiden Lichtquellen nicht in Frage. Der typische Beleuchter erfüllt Strahlereigenschaften, er hat eine feste Position und eine feste Abstrahlrichtung. Bei einigen Testsimulationen hat sich im Hinblick auf Reduzierung der Simulationszeiten bewährt, Punktlichtquellen anstelle der Strahler einzusetzen. Befindet sich das Zielobjekt stets mittig in der Strahlungskeule, sind die Unterschiede im Vergleich zu Simulationen mit Strahlern vergleichsweise klein. Diese Art Beleuchter wurde vorzugsweise während der Entwicklung und schneller Tests eingesetzt und wurde auch nach der Erweiterung der Algorithmen beibehalten. Der Anwender hat die Möglichkeit, jeden Strahler zu einer Punktlichtquelle zu deklarieren (dabei wird die Lichtintensität gleich der in Hauptstrahlrichtung gesetzt), der umgekehrte Weg funktioniert wegen fehlender Informationen nicht.

3.2.2 Lichtausbreitung und Ausbreitungsmedium

Breitet sich Strahlung in einem homogenen Medium aus, welches die Ausbreitung dämpft, so findet eine Intensitätsabnahme statt. Dabei wandelt die Energie der Strahlung in andere Energieformen um (z.B. Temperatur, chemische Reaktionen usw.). Sei I_0 die Intensität der eindringenden Strahlung und x die Koordinate im Ausbreitungsmedium parallel zur Ausbreitungsrichtung. Dann wird an einer Stelle x vorhandene Intensität $I(x)$ um einen Betrag $dI = -\alpha I(x)dx$ vermindert, wenn die Welle sich von x nach $x + dx$ fortbewegt. Integriert man nun die beiden Seiten über die Schichtdicke x , so erhält man das Lambert-Beersche Gesetz:

$$I(x) = I_0 \cdot e^{-\alpha x} \quad (3.1)$$

Der Absorptionskoeffizient α beschreibt dabei die Materialeigenschaften des durchquerten Mediums. Dieser hängt u.a. von der Wellenlänge der Strahlung, von der Dichte und der chemischen Zusammensetzung des Absorbermaterials ab.

3.2.3 Simulation der Rückstreuereigenschaften

Trifft Licht auf eine Oberfläche auf, so wird es je nach seiner Eigenschaft und Materialeigenschaften von dieser reflektiert, durchgelassen (transmittiert) und/oder verschluckt (absorbiert). Dabei können der Reflektionsgrad R_L , der Transmissionsgrad T_L und der Absorptionsgrad A_L theoretisch Werte von 0 bis 1 bzw. von 0 bis 100% annehmen. Aus dem Energieerhaltungsgesetz folgt, dass die Summe der einzelnen Anteile 1 bzw. 100% beträgt:

$$R_L + T_L + A_L = 1 \quad (3.2)$$

In den Abbildungen 3.1 und 3.2 sind mögliche Reflektions- und Absorptionsarten skizzenhaft dargestellt. Wird die einfallende Welle komplett in eine Richtung reflektiert, so spricht man von einer gerichteten Reflektion. Wird die Welle ungleichmäßig reflektiert, handelt es sich um gestreute Reflektion. Da gerichtete Reflektionen in der Realität nur ganz selten vorkommen (ideal spiegelnde Flächen) und gestreute Reflektionen schwer und nur mit großem Rechenaufwand berechenbar sind, werden die Oberflächen der Testobjekte als Lambert-Reflektoren, also diffus reflektierend, angenommen. Diese entspricht zwar nicht ganz der in der Natur am häufigsten vorkommenden Reflektion⁵, bietet jedoch eine gute Näherung.

⁵ Die meisten Materialien sind keine perfekten Lambert-Reflektoren. Bei bestimmten Einfallswinkeln wird das einfallende Licht nicht mehr in alle Richtungen gleich gestreut, sondern reflektiert bevorzugt in einem bestimmten Bereich.

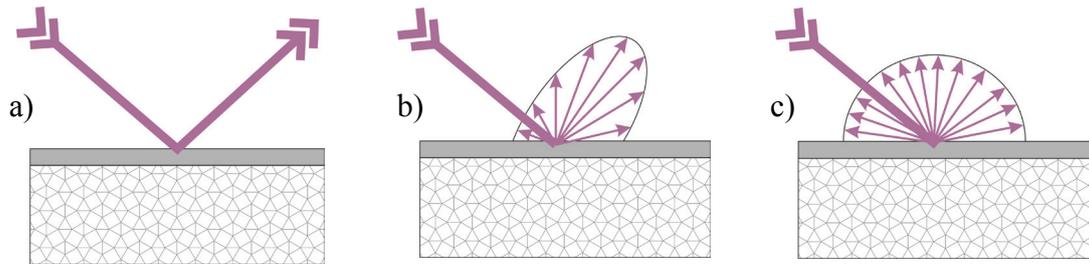


Abbildung 3.1: Reflektionsarten:

a) gerichtete Reflexion, b) gestreute Reflexion, c) diffuse Reflexion (Lambert-Reflektor)

Analog zu den oben genannten Reflektionsarten gibt es auch, je nach Material, auf das die Welle auftrifft, vollständige, gerichtete und diffuse Absorption.

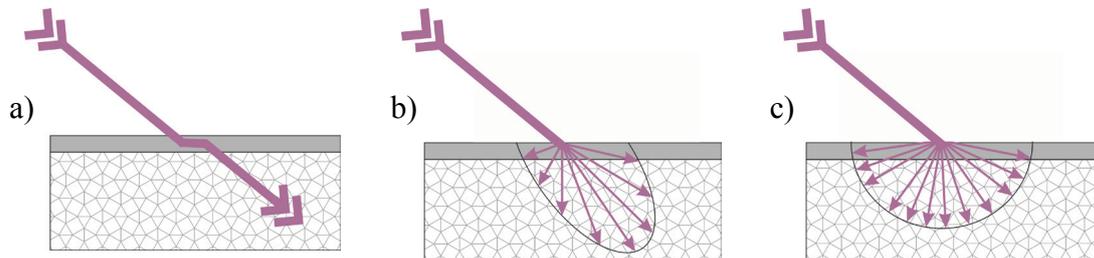


Abbildung 3.2: Absorptionsarten:

a) vollständiges Durchlassen des Lichtes, b) gerichtetes Durchlassen des Lichtes, c) diffuses Durchlassen des Lichtes

Bei der Berechnung der theoretischen Antwort des PMD Sensors wird die zurückgestrahlte Welle berücksichtigt, denn nur sie kann von dem Sensor detektiert werden. Daher ist es wichtig zu bestimmen, welcher Anteil der Gesamtenergie bzw. -intensität zurückgestrahlt wird. Auf welche Art das Licht dabei im Material absorbiert bzw. transmittiert wird, spielt für die Simulation keine Rolle. Die Tabelle 3.1 gibt einen kleinen Überblick über Reflektionsgrade verschiedener Materialien im Infrarotbereich. Wie man deutlich erkennen kann, hängt der Anteil des reflektierten Lichtes sehr stark vom Objektmaterial ab.

<i>Material</i>	<i>Reflektionsgrad in %</i>
Verputz	65-80
Zement, Beton	20-30
Holz	20-40
Klarglas	6-8
Weißes Papier	75-85
Schwarzes Plastik	10-20

Tabelle 3.1: Reflektionseigenschaften verschiedener Materialien im IR Bereich

Für möglichst realitätsnahe Simulationen war es also wichtig, die Möglichkeit zu schaffen, die Reflektionseigenschaften der Objekte in der Objektumgebung vorgeben und beeinflussen zu können. Je nachdem ob die Simulationsumgebung in Form von Punkten, Polygonen oder gar gemischt aufgebaut wird, wurden unterschiedliche Ansätze

realisiert. Doch bevor diese näher erläutert werden, wird erst mal der allgemeine Fall betrachtet – Reflektion von einem Lambert-Reflektor.

Strahlungsphysikalische Größen

Strahlung ist eine Form von Energie. Wird von einem Objekt elektromagnetische Strahlung ausgestrahlt, so kann diese Arbeit verrichten und durch Messung entsprechender physikalischer Effekte detektiert werden. Strahlungsenergie wird in Photometrie mit Q [in Joule] bezeichnet. Die Strahlungsleistung (auch Strahlungsfluss genannt) ist als Gesamtenergie pro Zeiteinheit definiert [Heri 08]:

$$\Phi = \frac{dQ}{dt}, \quad [\text{in J/s, } 1\text{J/s}=1\text{W}=1\text{Watt}] \quad (3.3)$$

Die Strahlungsleistung in einem beliebigen Punkt im Raum, die von einem Strahler abgegeben wird, hängt von seiner Fläche und seinem Abstand zu diesem Punkt ab.

Strahlt eine Punktquelle Licht in alle Richtungen gleichermaßen (isotrop) aus [vgl. 3.2.1] und es gibt keinen Energieverlust, so folgt aus dem Energiebilanzsatz, dass die Beleuchtungsstärke mit dem Quadrat des Abstandes zur Lichtquelle abnimmt: $E \propto \frac{1}{R^2}$

(siehe auch (3.8)). Dies folgt aus dem Grund, dass sich die gesamte Strahlungsenergie Q bei einer isotropen Ausbreitung auf der Oberfläche einer Kugel mit dem zeitlich zunehmenden Radius R verteilt.

Betrachtet man die räumliche Verteilung der Strahlung in einem Kegelausschnitt, so wird die Strahlungsenergie Q_1 in einem Abstand r_1 gleich der Strahlungsenergie Q_2 in einem Abstand r_2 sein (s. Abbildung 3.3)

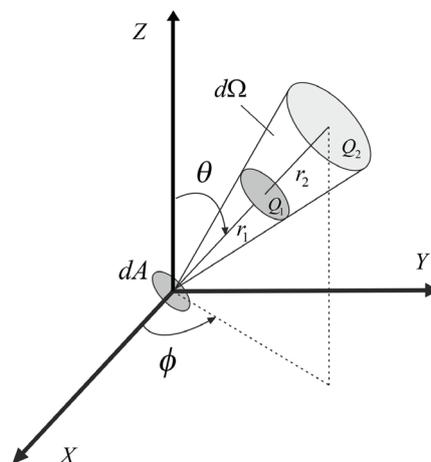


Abbildung 3.3: Definition des Raumwinkels

In diesem Zusammenhang wird der Begriff Raumwinkel eingeführt, der durch das Verhältnis der Fläche A auf der Kugel zum Abstand definiert ist:

$$d\Omega = \frac{dA}{R^2} \quad (3.4)$$

$$\text{Maßeinheit Steradian: } 1[\text{sr}] = 1 \left[\frac{\text{m}^2}{\text{m}^2} \right] \hat{=} \left[\frac{\text{Fläche}}{\text{Radius}^2} \right]$$

Der größte mögliche Raumwinkel wäre dementsprechend $4\pi[\text{sr}]$ bei einer ganzen Kugel.

Die von einer Lichtquelle ausgestrahlte Strahlungsleistung pro Raumwinkel wird als Strahlungsstärke bzw. Strahlungsintensität bezeichnet:

$$I = \frac{d\Phi}{d\Omega}, \text{ in } \left[\frac{\text{W}}{\text{sr}} \right] \quad (3.5)$$

Betrachtet man die Sendefläche A unter dem Winkel ε_1 zur Flächennormalen (siehe Abbildung 3.4), dann reduziert sich die sichtbare Sendefläche. Die wirksame Fläche ist dann gleich $A \cos(\varepsilon_1)$ (siehe auch (3.9)) und die Strahlungsintensität

$$dI(\varepsilon_1) = L \cdot dA \cos(\varepsilon_1) \quad (3.6)$$

mit L - Strahlungsdichte. Strahlungsdichte (Strahldichte in Photometrie) gibt an, welche Strahlungsleistung von einem gegebenen Punkt des Strahlers in die durch den Azimutwinkel ϕ und den Polarwinkel θ gegebene Richtung pro wirksames Flächenelement $dA \cos(\varepsilon_1)$ und pro Raumwinkelelement $d\Omega$ ausgestrahlt wird.

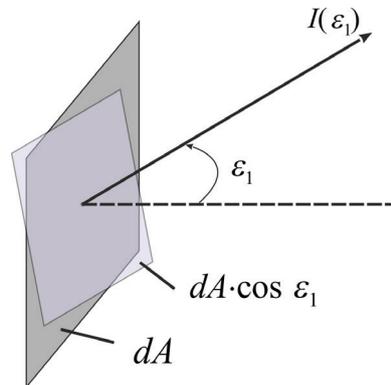


Abbildung 3.4: Definition der Strahlungsintensität

$$L = \frac{d^2\Phi}{dA \cos(\varepsilon_1) \cdot d\Omega}, \text{ in } \left[\frac{\text{W}}{\text{m}^2 \cdot \text{sr}} \right] \quad (3.7)$$

Die Strahlungsintensität I als Funktion des Abstrahlwinkels kann experimentell bestimmt werden (siehe Abbildung 3.5). Viele Hersteller geben diese Daten in ihren Datenblättern an. In Abbildung 3.6 ist ein Ausschnitt aus einem solchen Datenblatt dargestellt.

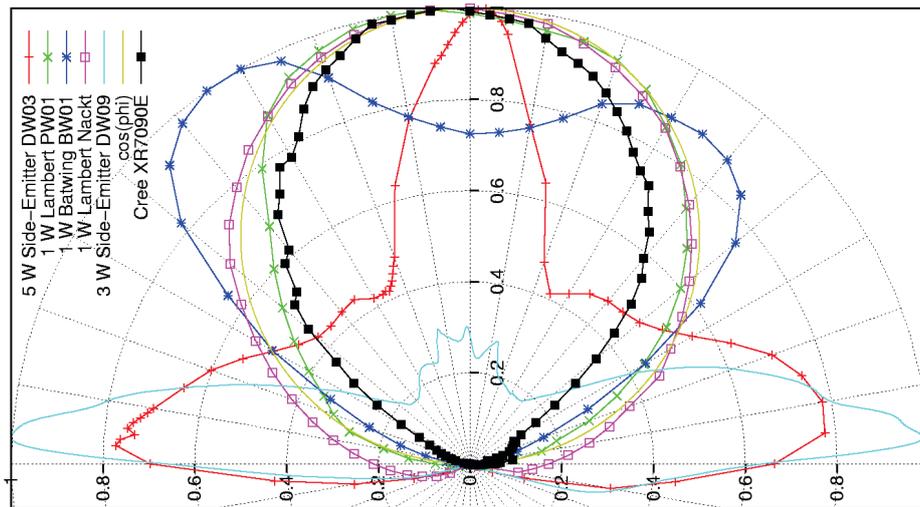


Abbildung 3.5: Gemessene normierte Abstrahlcharakteristika unterschiedlicher LED's in Polarkoordinaten

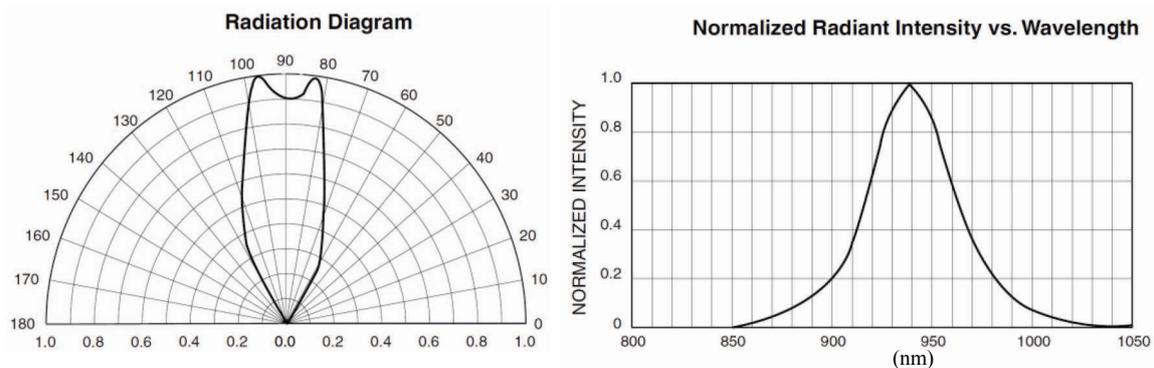


Abbildung 3.6: Beispiel: Abstrahlcharakteristik von QED233/234 von Fairchild Semiconductor Links – die Abstrahlcharakteristik (Lichtstärkeverteilungskurve, LVK), rechts – Lichtintensität als Funktion der Wellenlänge

Neben der ausgestrahlten Strahlleistung, die ein Maß dafür ist, wie viel Strahlung pro Sekunde in einen bestimmten Raumwinkel abgegeben wird, spielt der Begriff Beleuchtungsstärke (photometrische Bestrahlungsstärke) eine wichtige Rolle. Sie gibt an welche Strahlungsleistung auf dem Empfänger mit der Fläche A letztendlich ankommt.

$$E = \frac{d\Phi}{dA_{\text{eff}}} = \frac{I \cdot d\Omega}{dA_{\text{eff}}} = \begin{cases} \frac{I}{R^2}, & \text{für } \varepsilon_2 = 0 \\ \frac{I}{R^2} \cdot \cos(\varepsilon_2), & \text{für } \varepsilon_2 \neq 0 \end{cases}, \text{ in } \left[\frac{W}{m^2} \right] \quad (3.8)$$

mit A_{eff} – effektive Querschnittsfläche



Abbildung 3.7: Veranschaulichung der Beleuchtungsstärke bei senkrechtem (links) und schrägem (rechts) Lichteinfall

Die Beleuchtungsstärke ist also ein Maß für die Helligkeit an einem bestimmten Ort. Diese nimmt, wie bereits erwähnt, mit dem Quadrat des Abstandes zur Lichtquelle ab.

Lambert-Strahler

Analog zu einem Lambert-Reflektor, bezeichnet man als Lambert-Strahler eine Lichtquelle, die aus allen Richtungen gleich hell aussieht ($L = const$). In der Natur kann man die Sonne und den Mond näherungsweise als Lambert-Strahler bezeichnen, die Verdunkelung am Rand dieser Strahlkörper ist vergleichsweise gering. Mit der Bedingung, dass die Strahlungsdichte konstant sein soll, folgt mit (3.6)

$$L = \frac{dI}{dA_{eff}} = const \quad \Rightarrow \quad dI(\varepsilon_1) = L \cdot dA \cos(\varepsilon_1) \quad (3.9)$$

$$I(\varepsilon_1) = I_0 \cos(\varepsilon_1) \text{ – Lambertsches Kosinusetz, mit } -\pi/2 \leq \varepsilon_1 \leq \pi/2$$

Als Lambert-Strahler wird also ein Strahler bezeichnet, bei dem die Strahlungsstärke der von einem Flächenelement ausgestrahlten Strahlung mit größer werdendem Abstrahlwinkel kontinuierlich abnimmt. Seitlich des Strahlers ist die Strahlungsstärke gleich Null, da die effektive Strahlfläche Null ist. Viele Körper mit rauer diffus reflektierender Oberfläche sind in guter Näherung Lambert-Strahler.

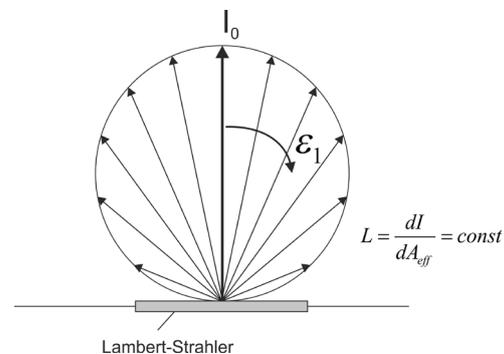


Abbildung 3.8: Abstrahlcharakteristik eines Lambert-Strahlers

Photometrisches Entfernungsgesetz

Nun wird eine lambertsche Sender-Empfänger Kombination betrachtet (siehe Abbildung 3.9). Angenommen, dass der Sender mit der Fläche A_1 die Strahlungsdichte L_e in die Richtung senkrecht zur der Senderfläche ausstrahlt, folgt für die Strahlungsstärke mit (3.4) und (3.5):

$$I_e(\varepsilon_1) = \frac{d\Phi_e}{d\Omega} = L_e \cdot A_1 \cos(\varepsilon_1) \quad (3.10)$$

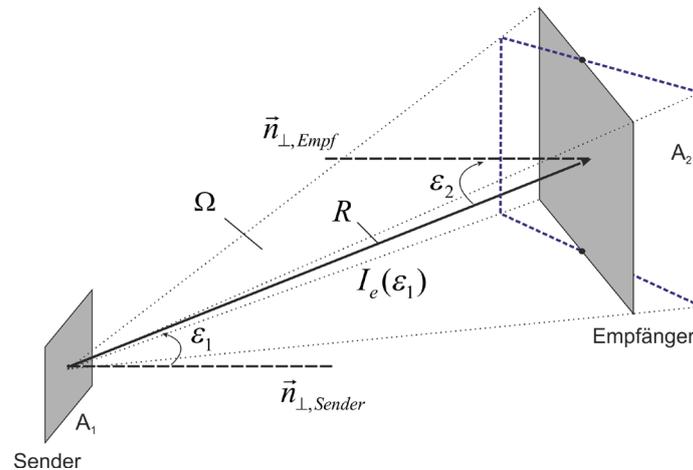


Abbildung 3.9: Photometrisches Grundgesetz

Diese Strahlung wird von einem Empfänger mit der Fläche A_2 aufgenommen, deren Normale mit dem Vektor $\vec{I}_e(\varepsilon_1)$ einen Winkel ε_2 aufspannt. Der wirksame Raumwinkel ist dann gegeben durch:

$$d\Omega = \frac{dA_2 \cos(\varepsilon_2)}{R^2} \quad (3.11)$$

Somit folgt für die Strahlungsleistung und die Beleuchtungsstärke

Photometrisches Grundgesetz:

$$d^2\Phi_e = d\Omega \cdot L_e \cdot dA_1 \cos(\varepsilon_1) = L_e \cdot \frac{dA_1 \cos(\varepsilon_1) \cdot dA_2 \cos(\varepsilon_2)}{R^2} \quad (3.12)$$

Photometrisches Entfernungsgesetz:

$$E_e = \frac{I_e(\varepsilon_1)}{R^2} \cdot \cos(\varepsilon_2) \quad (3.13)$$

Das photometrische Gesetz sagt aus, dass die Beleuchtungsstärke mit zunehmendem Abstand umgekehrt proportional zum Quadrat des Abstandes und proportional zum Kosinus des Lichteinfallwinkels ist. Dieses Gesetz gilt aber nur für Abstände größer als die photometrische Grenzentfernung.

Photometrische Grenzentfernung und Approximation

Die in (3.6)-(3.13) gewonnenen Erkenntnisse gelten für punktförmige Strahler und Empfänger. Betrachtet man einen Strahler sehr geringer Ausmaße, so bleibt der Ausstrahlwinkel ε_1 in Richtung eines Empfängers ebenfalls geringer Ausmaße näherungsweise konstant. Sind die Ausmaße des Strahlers groß, so weisen unterschiedliche Berei-

che des Strahlers zum Empfänger hin unterschiedliche Abstrahlwinkel auf. Das gleiche gilt bei verhältnismäßig großen Empfängern in Relation zur Ausbreitungsdistanz Strahler-Empfänger. In solchen Fällen sind die Abweichungen aufgrund der großen Unterschiede in der Geometrie nicht zu vernachlässigen. Die Abbildung 3.10 veranschaulicht das Problem. Betrachtet man einen Punktstrahler im Ursprung des Koordinatensystems, so stellen Kugeln um diesen Punkt herum Flächen gleicher Beleuchtungsstärke dar. Die Strahlungsleistung in jedem Punkt auf der Kugeloberfläche ist gleich. Möchte man jedoch die Beleuchtungsstärke auf einem ebenen Objekt, und nichts anderes sind die einzelnen Polygone einer beleuchteten Simulationsszene, ausrechnen, so wird ersichtlich, dass die Beleuchtungsstärke am Rand des Polygons eine andere ist als die Beleuchtungsstärke in der Mitte (siehe Abbildung 3.10, rechts).

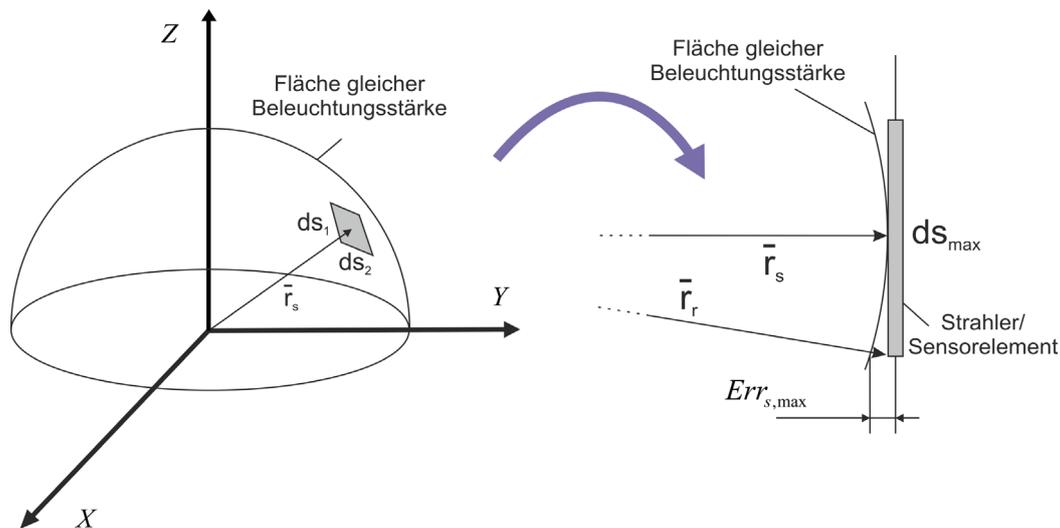


Abbildung 3.10: Veranschaulichung des Näherungsfehlers bei Unterschreitung der photometrischen Grenzentfernung

Der Grund dafür können nicht nur die unterschiedlichen Entfernungen $|\vec{r}_s| \neq |\vec{r}_r|$ sein, sondern im Falle eines Strahlers (z.B. Lambert-Strahler) auch die unterschiedlichen Richtungen und somit auch die unterschiedlichen Beleuchtungsstärken in Richtung der beiden Richtungsvektoren.

In diesem Zusammenhang spricht man von der sog. photometrischen Grenzentfernung – der Entfernung zwischen Lichtquelle und Empfänger, bei der ein vorgegebener Messfehler, verursacht durch die endliche Ausdehnung des Strahlers und Empfängers, nicht überschritten wird. Faktoren, die Einfluss auf den Fehler haben sind u.a.:

- Das Verhältnis der größten Ausdehnung des Senders (aktiver Strahler) zum Abstand Sender-Empfänger
- Das Verhältnis der größten Ausdehnung des Empfängers (aktiver Empfänger) zum Abstand Sender-Empfänger
- Eigenschaften des Strahlers/Empfängers
- Strahlungsstärkeverteilung usw.

Nach DIN5032 gilt für einen kreisförmigen Lambert-Strahler und für einen kosinusgetreu bewertenden Empfänger als photometrische Grenzentfernung ein Abstand, der 10-mal so groß ist wie die größte Ausdehnung von Sender bzw. Empfänger, wenn der Fehler kleiner als 0,25% sein soll [Heri 05].

Betrachtet man die Ausmaße der üblichen Infrarot LED's in gängigen PMD-Beleuchtern, dann liegen sie im Bereich einiger mm. Der Fehler hervorgerufen durch Annahme eines Strahlers gleicher Strahlcharakteristik mit verschwindend kleinen Dimensionen wäre sehr gering und zu vernachlässigen. Achtet man dagegen bei der Auswahl der Simulationstestkörper nicht auf ausreichend feine Polygongröße, so kann es vorkommen, dass ab einer bestimmten Polygongröße besonders im Nahbereich des Beleuchters bei Approximation des Helligkeitsverlaufs auf dem Polygon durch eine konstante Größe nicht ausreichend hohe Genauigkeit geboten werden kann. Dieses Problem

kann nicht nur bei schlecht aufgelösten Objekten vorkommen, sondern auch bei Objekten, die nur mit Hilfe einiger Polygone unendlich genau definiert werden können. Zur Definition eines Quaders sind z.B. 6 viereckige Polygone (Quads) ausreichend, für eine ebene Wand sogar nur einer.

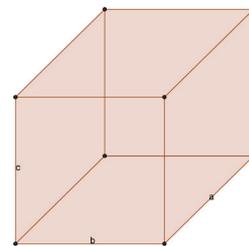


Abbildung 3.11: Quader bestehend aus 6 Polygonen

Für eine hinreichend genaue Berechnung der Helligkeitsverteilung auf dem Objektkörper, muss das Verhältnis des Simulationstrios – die Beleuchtergröße, die Empfängergröße und der Abstand zwischen den beiden – überwacht werden. Sind die Bedingungen nicht erfüllt, dann müssen sie differentiell formuliert und über die Beleuchter und die Empfängerfläche integriert werden.

3.3 Softwaretechnische Umsetzung

Nach der Betrachtung des Lichtes als elektromagnetische Welle und der Grundlagen der photometrischen Lichtausbreitung in einem homogenen Medium, wird in diesem Abschnitt das Simulationskonzept des *Light Propagation Module* vorgestellt. In der Aufgabenstellung (siehe Kapitel 3.1) sind die Hauptziele und die gewünschten Freiheitsgrade für den Anwender bereits behandelt worden, hier wird nun der grundlegende Entwurf zur softwaretechnischen Umsetzung vorgestellt.

Wie bereits in Kapitel 1 erläutert, werden bei einer PMD Kamera die Messobjekte von dem/n Beleuchter/n in Form von Lichtimpulsen angeleuchtet. Das von der Szene reflektierte Licht wird anschließend von dem Sensor ausgewertet. Für die Berechnung der Strahlungsleistung, die auf ein bestimmtes PMD Pixel zurückgeworfen wird, ist die Rückstrahlleistung derjenigen Teile der Oberfläche der beleuchteten Szene relevant, die genau mit diesem Pixel korrespondiert. Die Zuordnung der Strahlungsleistung

zu den einzelnen Punkten einer 3D Szene stellt bei der Berechnung der theoretischen Antwort des PMD-Sensors (wird in Kapitel 5 behandelt) die Hauptaufgabe des *Light Propagation Module* dar. Dies muss jedoch im Zusammenspiel mit der Simulationskomponente des PMD Sensors geschehen. Als Endergebnis sind nicht nur die Informationen über die jeweiligen Strahlungsleistungen wichtig. Bereits in diesem Schritt kann ein großer Teil der Simulationszeit eingespart werden, wenn man die Teile der gesamten Szene, die nicht beleuchtet werden können, im Simulationsprozess des PMD Sensors aus der Szene herausnimmt. Denn was nicht angestrahlt wird, kann nichts zurückstrahlen. Daher muss das *Light Propagation Module* über externe Schnittstellen mit weiteren Simulatorkomponenten aktiv agieren, Daten anfordern, Berechnungen durchführen, Daten sichern und letztendlich einen Teil der Ergebnisse visuell darstellen können.

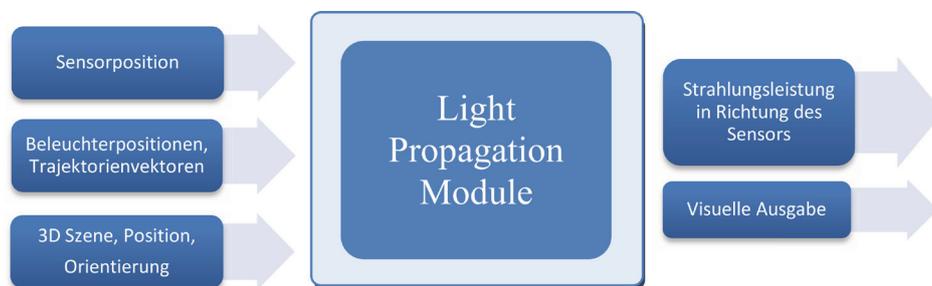


Abbildung 3.12: Das Prinzip des Light Propagation Module beschränkt auf Input/Output (ohne Schnittstellen)

In Abbildung 3.12 ist das Prinzip des *Light Propagation Module* (LPM) schematisch dargestellt. Auf die einzelnen Schnittstellen zu anderen Komponenten (die in der Abbildung nicht dargestellt sind) und ihre Bedeutung wird in späteren Kapiteln eingegangen. Das LPM wurde als eine eigenständige Komponente entwickelt und kann bei Einhaltung bestimmter Datenvorgabe und -formate vom eigentlichen PMD Simulator getrennt ausgeführt werden. Die Schnittstellen müssen jedoch in diesem Modus nach außen hin „geschlossen“ werden (Independent Mode), weil das Modul sonst auf Eingabebefehle von der *Frame Control Unit* (FCU) wartet (wird in Kapitel 8.1 näher beschrieben). Dieser Datenfluss stellt eingangsseitig die Informationen dar, die das LPM zur Berechnung der Strahlungsleistungen braucht. Dies sind:

- ***Sensorposition im Raum***

Das *Light Propagation Module* rechnet Strahlungsleistungen direkt in Richtung des Sensors aus, so dass diese später ohne weitere Umrechnungen als zu dem Pixel korrespondierende Werte ausgewertet werden können. Dazu wird die relative Sensorposition bezüglich der Simulationsszene und der Beleuchterpositionen benötigt.

- ***Beleuchterpositionen, Trajektorienvektoren***

Um die Ausrichtung der Beleuchter bezüglich der Simulationsszene zu bestimmen, werden neben der Position der Beleuchter von der FCU Trajektorienvektoren übermittelt. Zusammen bilden sie später Ausrichtungsvektoren im Raum. Mit der Position, der Ausrichtung der Beleuchter und der Position des Sensors wird im Zusammenspiel mit der Simulationsszene eine im Raum vollständig beschriebene Beleuchter-Punkt-Sensor „Zelle“ gebildet – die kleinste zu simulierende Einheit.

- **3D Szene, Position und Orientierung**

Die Position und Ausrichtung der Simulationsszene bezüglich der Beleuchter und Sensoren ist maßgebend bei der Berechnung der Strahlungsleistung, die von dieser Szene in Richtung der Sensoren zurückgeworfen wird.

Ausgangsseitig stellt das *Light Propagation Module* die Daten zur Strahlungsleistung jedes einzelnen Punktes der Szene in Richtung des Sensors bereit. Diese werden im PMD-Simulator zur Berechnung der Helligkeits- und der Entfernungsbilder benötigt. Wird die LPM Komponente unabhängig von dem PMD-Simulator ausgeführt (Independent Mode), dann ist es möglich, sich die Strahlungsleistungsverteilung einer Szene aus einem beliebigen Punkt anzuschauen. Wird dies nicht explizit über die Schnittstelle angesteuert, so wird die reflektierte Strahlungsleistung immer in die Richtung des Strahlers berechnet – Kamera fällt dann mit dem Beleuchter in einem Punkt zusammen (monostatische Konstellation).

Nun wird die Modellierung und die softwaretechnische Umsetzung des *Light Propagation Module* etwas detaillierter betrachtet. Wie bereits erwähnt, stellt das LPM im Prinzip ein eigenständiges Tool dar, welches über Schnittstellen nach Vorgabe der Ausgangssituation angesteuert werden kann. Nach der Berechnung werden die Ergebnisse über eine eigene Datenschnittstelle an die *Frame Control Unit* übergeben und der Erfolg bzw. Misserfolg (z.B. bei internen Fehlermeldungen, Fehlermeldungen wegen Verletzung der Formatvorlagen usw.) der Berechnung quittiert. Erst nach dieser Quittierung werden im Simulator weitere Schritte eingeleitet. Die Hauptaufgabe dieser Komponente besteht also darin, in einem Zusammenspiel mit dem PMD-Simulator einen Teil der Berechnungen (Methoden) nach außen hin abgekapselt durchzuführen. Der Übersichtlichkeit wegen werden einige (meistens programmiertechnisch als Klassen ausgeführte) Komponenten des gesamten Simulators hier erklärt, obwohl sie im Programm teilweise in der *Frame Control Unit* integriert sind. Diese stellt (siehe Kapitel 8) eine zentrale Überwachungskomponente sowie eine Datenbank an Daten und Informationen dar, die von einigen Teilen des Simulators gemeinsam genutzt werden und aus diesem Grund und aus dem Grund der Übersichtlichkeit in externen Komponenten untergebracht wurden.

3.3.1 Modellierung einer Lichtquelle

Eine Lichtquelle stellt die kleinste Grundfunktionalität dar. Sie besitzt einen Ort und eine Strahlungscharakteristik. Prinzipiell wird jede Lichtquelle mit einem temporären Koordinatensystem „ausgestattet“, in dem diese für die Zeit der Berechnung definiert und in welchem die Ausbreitung der Lichtwellen beschrieben wird. Nach der Berechnung der Interaktion „Lichtquelle (i, j) – Objekt (k, m) “ wird das Koordinatensystem an das nächste Paar „vererbt“. Dabei ist i – Nummer des Beleuchters, j – Nummer der Lichtquelle, k – Nummer des Objektes in der Simulationsszene und m – Nummer des Punktes im Objekt. Auf diese Weise besteht einerseits die volle Kontrolle über jede Interaktion, da über die zuständigen Indizes jedes „Lichtquelle-Objekt“-Paar einzeln manipuliert/verändert werden kann und somit das Paar einzeln einstellbar bleibt, andererseits bleibt aber auch das Datenmanagement im Programm vergleichsweise schlank. Die Koordinaten und Richtungsvektoren müssen lediglich einmal geladen und abgearbeitet werden und können somit das Selbstständigkeitsprinzip des *Light Propagation Module* nicht gefährden. Ein weiterer Vorteil dieser Methode ist die relativ einfache Umsetzung der Markierung einzelner Punkte oder Bereiche der Simulationsszene. Diese kann z.B. sehr nützlich sein, wenn man bestimmte Bereiche der Szene in den simulierten Daten treffsicher wiederfinden soll (z.B. durch Beeinflussen der Rückstreuungseigenschaften bestimmter Punkte).

Unterschieden wird zwischen zwei Lichtquellen:

- Punktlichtquelle – homogene Lichtausstrahlung in alle Richtungen mit einer frei definierbaren Anfangslichtintensität I_0
- Strahler bzw. LED mit einer frei definierbaren Abstrahlcharakteristik und einer festen Anfangslichtintensität I_0 in die Hauptrichtung der Strahlungskeule

Bei Abstrahlcharakteristika zu LED's, die man den Datenblättern entnehmen kann, handelt es sich in der Regel um eine Intensitätsaufnahme in einer Ebene durch die Strahlhauptachse der LED (siehe als Beispiel Abbildung 3.14, links). Dreidimensionale Aufnahmen der Intensitätsverteilung sind nur bei speziellen Ausführungen wichtig, wo die Lichtführung in der Diode z.B. durch im Glas integrierte kleine Mikrolinsen beeinflusst werden kann. In den klassischen Anwendungen der PMD-Technik werden in PMD-Beleuchtern LED's mit rotationssymmetrischen Abstrahlcharakteristika⁶ eingesetzt. Im Simulator können daher, zumindest im aktuellen Entwicklungsstadium, nur rotationssymmetrische LED's betrachtet werden. Eine Asymmetrie kann jedoch trotzdem leicht erreicht werden, wenn eine Lichtquelle in mehrere Teillichtquellen zerlegt wird, deren Hauptachsen in unterschiedliche Richtungen zeigen.

⁶ Aufgrund der Herstellertoleranzen können die Abstrahlcharakteristika natürlich nur näherungsweise als rotationssymmetrisch betrachtet werden.

Unabhängig davon ob Abstrahlcharakteristika in Polar- oder in kartesischen Koordinaten vorliegen (siehe die Abbildungen unten), müssen diese Daten im Simulator tabellarisch aufbereitet werden.

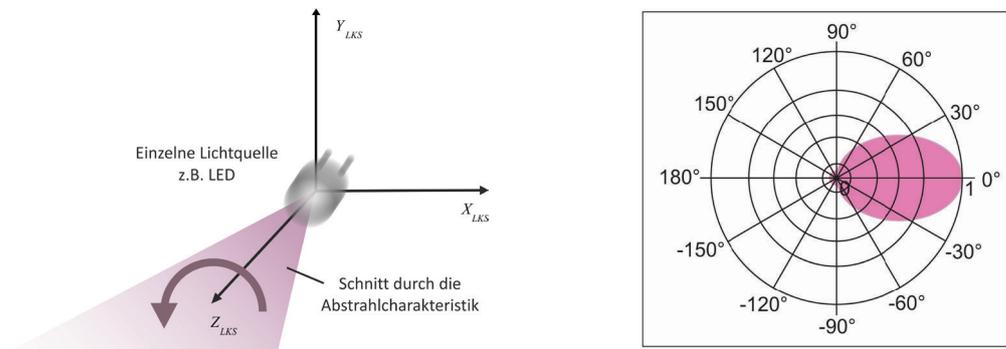


Abbildung 3.13: Veranschaulichung der Definition der Abstrahlcharakteristik (links) und Abstrahlcharakteristik in Polarkoordinaten (rechts)

Betrachtet man einen typischen Verlauf der Abstrahlcharakteristik (siehe Abbildung 3.14, links und Abbildung 3.5), so stellt man fest, dass die auf eine Ebene reduzierte Abstrahlcharakteristik oft keine absolute Symmetrie bezüglich der Hauptstrahlrichtung aufweist:

$$I_{LED}(-\alpha) \neq I_{LED}(+\alpha) \tag{3.14}$$

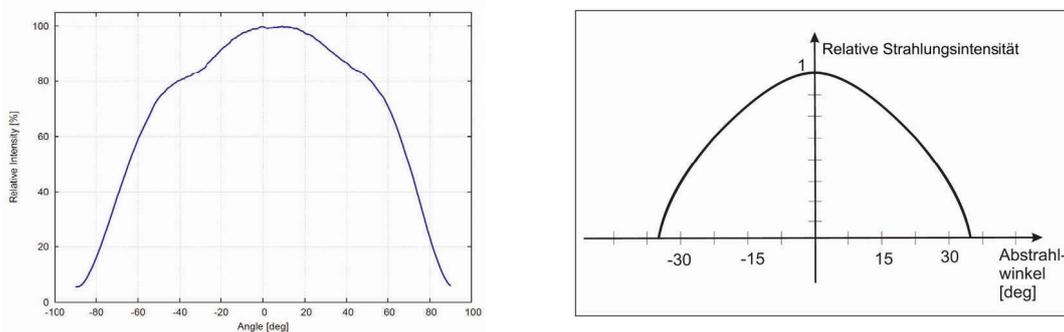


Abbildung 3.14: Beispiele: Abstrahlcharakteristik einer LED aus einem Datenblatt (links), rotationssymmetrische räumliche relative Strahlungsintensität einer LED (rechts)

Wird von dem Benutzer eine solche Vorgabe der Abstrahlcharakteristik gemacht, dann wird anstelle des eigentlichen Wertes ein Mittelwert aus beiden Werten symmetrisch an der Haupttrichtung ($\alpha = 0$, mit α – Abstrahlwinkel) gebildet. Mit anderen Worten, die

Vorgabe im Simulator ist: $I_{LED}(-\alpha) = I_{LED}(+\alpha)$

Wenn $I_{LED}(-\alpha) \neq I_{LED}(+\alpha)$, dann \Rightarrow

$$I_{LED}(-\alpha)_{NEU} = I_{LED}(+\alpha)_{NEU} = \frac{I_{LED}(-\alpha) + I_{LED}(+\alpha)}{2}$$

Strahlwinkel in [deg]	-90	-85	-80	-75	-70	...	70	75	80	85	90
Relative Lichtstärke in [%]	4,2	7,2	17,3	24,7	34,1	...	51,1	34,9	23,0	12,3	4,7



Strahlwinkel in [deg]	90	85	80	75	70	...
Relative Lichtstärke in [%]	4,45	9,75	20,15	29,8	42,1	...

Tabelle 3.2: Veranschaulichung der Erzeugung der Strahlungsintensität-Tabelle als Input für die Lichtsimulationskomponente

Zusammenfassend: Wird von dem Benutzer keine Abstrahlcharakteristik vorgegeben, so nimmt das Programm automatisch einen Lambert-Strahler an. Wird eine Abstrahlcharakteristik im Bereich $-\pi/2 \leq \alpha \leq \pi/2$ vorgegeben, dann wird mit den Daten eine rotationssymmetrische Abstrahlcharakteristik für einen Abstrahlwinkel $0 \leq \alpha \leq \pi/2$ berechnet. Die Zwischenwerte können wahlweise linear (Spline-Interpolation, schneller) oder kubisch (c^2 -Spline Interpolation, etwas langsamer) interpoliert werden.

Neben der Abstrahlcharakteristik werden für jede LED ihre Position im Raum und der Ausrichtungsvektor vorgegeben. Die Position entspricht der Lage des Schwerpunktes einer LED. Die LED selbst wird durch einen Punktstrahler mit der gleichen Abstrahlcharakteristik simuliert (dies entspricht einer LED mit verschwindend geringen Ausmaßen). Der Ausrichtungsvektor zeigt mittig in Richtung der Strahlhauptkeule.

Ein weiterer Freiheitsgrad ist die Betriebsart – eine Art Schalter, mit dem jede LED im Simulator ein- und ausgeschaltet werden kann.

So stellt jede Lichtquelle bzw. LED programmtechnisch ein *Objekt* dar, welches nach außen hin eine geschlossene Struktur mit frei definierbaren aber während eines Frames nicht veränderbaren Eigenschaften ausweist. Zwischen den einzelnen Frames sind *Betriebsart*, *Position* und der *Ausrichtungsvektor* veränderbar, jedoch nicht die *Abstrahlcharakteristik*.

3.3.2 Modellierung eines Beleuchters

Werden zwei oder mehr LED's zu einem starren System verbunden, dann bildet diese Zusammensetzung einen Beleuchter. Ein Beleuchter besteht also aus mindestens zwei Lichtquellen, die in einer festen Relation zueinander stehen. Die Hauptmerkmale sind:

- Die Anzahl der Lichtquellen $N \geq 2$.
- Alle Lichtquellen eines Beleuchters besitzen die gleiche Abstrahlcharakteristik.
- Alle Lichtquellen eines Beleuchters werden auf die gleiche Weise angesteuert und senden deshalb zur gleichen Zeit das gleiche Modulationssignal aus. Die einzige Ausnahme ist die *Betriebsart* der Lichtquelle. Das Ein- und Ausschalten einer Lichtquelle in einem Beleuchter ist im Simulator in der Zeit zwischen zwei Frames möglich, jedoch nicht während der Berechnung eines Frames.

- Die relative Position aller Lichtquellen im Beleuchter ist fest und während der Simulation nicht veränderbar.
- Die Hauptrichtungen der Strahlungskeulen sind bei allen Lichtquellen parallel zueinander, d.h. die zeigen alle in die gleiche Richtung auf einen Punkt im Unendlichen.
- Ähnlich wie bei Lichtquellen verfügt auch jeder Beleuchter über die Funktionalität *Betriebsart*. Zwischen zwei Frames kann ein Beleuchter ein- bzw. ausgeschaltet werden. Wird im gleichen Schritt auch die *Betriebsart* der Lichtquelle verändert, so wirkt sich diese Veränderung vorrangig aus, da sie im Programm eine höhere Priorität besitzt (sonst könnte eine defekte Lichtquelle versehentlich eingeschaltet werden). Ansonsten besteht zwischen den beiden *Betriebsarten* eine logische „AND“-Operation.

In Abbildung 3.15 ist die Simulationsanordnung einer CamCube dargestellt. Die CamCube (links im Bild) besitzt zwei rechteckförmige Beleuchter seitlich der Kamera. Das Simulationsmodell dieser Anordnung besteht somit aus zwei Beleuchtern mit der gleichen Anzahl an LED's wie im realen Beleuchter.

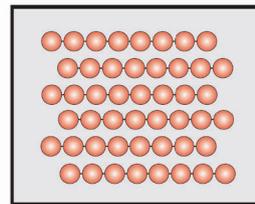


Abbildung 3.15: Rechteckförmige Beleuchtung einer PMD[vision]® CamCube 3.0 (links) und das dazugehörige Simulationsmodell eines der beiden Beleuchter (rechts)

Abbildung 3.16 zeigt einen anderen, ringförmigen Beleuchter einer 2D/3D MultiCam.



Abbildung 3.16: Ringförmige Beleuchtung der PMD Kamera einer 2D/3D MultiCam (links) und das dazugehörige Simulationsmodell (rechts)

Um einen Beleuchter im 3D Raum zu definieren, werden zunächst die Koordinaten aller seiner LED's bezüglich eines fest definierten Punktes benötigt. Daher besitzt jeder Beleuchter sein eigenes Koordinatensystem (siehe Kapitel 4.7), in dem die Positionen der einzelnen LED's definiert werden. Es existieren Trajektorien-Koordinatensysteme für

jeden Beleuchter einer Simulationsumgebung, in denen die Lagewinkel, die Position und die Geschwindigkeitsvektoren beschrieben werden. Diese bilden ihrerseits ein Bindeglied zwischen dem Beleuchterkoordinatensystem und dem Weltkoordinatensystem.

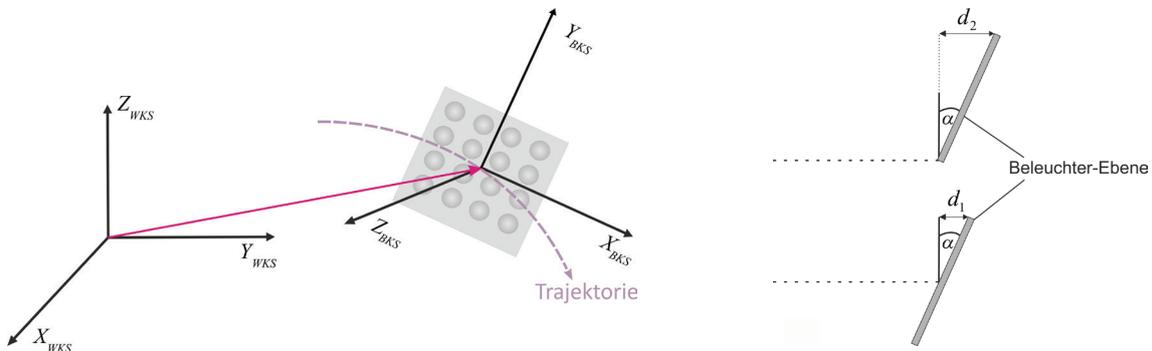


Abbildung 3.17: Der geometrische Modellierungsansatz eines Beleuchters

Es empfiehlt sich, die Koordinaten der einzelnen LED's achsensymmetrisch zum Ursprung des Beleuchterkoordinatensystems festzulegen. Für die gängigen PMD-Beleuchter wird es erreicht, wenn der Ursprung des Koordinatensystems im Massenschwerpunkt der meistens planaren Anordnung der LED's gelegt wird (siehe Abbildung 3.17). Das hat den Hintergrund, dass alle Drehungen des Beleuchters im Weltkoordinatensystem bezüglich dieses einen Punktes gemacht werden und, wie die Abbildung rechts veranschaulicht, stellt eine Drehung bezüglich der Mitte des Beleuchters eine andere Drehung dar, als wenn man um einen am Rand liegenden Koordinatenursprung drehen würde.

Für Beleuchter gängiger PMD-Kameras liegen die Koordinaten der LED's bezüglich des Schwerpunktes bereits im entsprechenden MATLAB-Format vor und können direkt eingebunden werden. Wird vom Benutzer eine neue Tabelle angegeben, dann überprüft

das Programm für planare Beleuchter selbstständig ob das Kriterium $\left| \sum_i \vec{r}_{BKS,i} \right| < \varepsilon$ er-

füllt ist und warnt den Benutzer vor, falls dies nicht der Fall ist. Mit ε – der kleinste Abstand zwischen zwei benachbarten LED's und $\vec{r}_{BKS,i}$ – der Richtungsvektor zur LED i .

3.3.3 Modellierung der Modulation

Wie bereits in Kapitel 2 erwähnt, wird bei klassischen PMD-Anwendungen zur Beleuchtung der Umgebung inkohärentes Licht eingesetzt. Die einzelnen Infrarot-LED's werden zwar in einem Beleuchter synchron angesteuert, die senden aber relativ breitbandig. D.h. das Spektrum einer solchen LED enthält neben der für diese LED spezifischen Wellenlänge auch Frequenzanteile, die um ein paar Prozent von dem spezifischen Wert abweichen. Kohärenzlängen typischer Infrarot LED's liegen im μm Bereich. Somit setzt sich das von den Lichtquellen emittierte Licht aus vielen einzelnen Wellenzü-

gen zusammen und es entsteht kein Interferenzmuster im typischen Sinne. Überlagerung inkohärenter Strahlung führt daher zu örtlich statistischen Verstärkungen (konstruktive Interferenz) und Auslöschungen (destruktive Interferenz), es entsteht kein festes zeitliches Muster. Bei Überlagerung inkohärenter Wellen beobachtet man demzufolge das statistische Mittel aus den einzelnen Intensitäten.

Um die Möglichkeit der Erstellung synthetischer Daten mit kohärenten Lichtquellen dennoch zu ermöglichen, wurde im *Light Propagation Module* die Möglichkeit geschaffen, die Art der Lichtquellen zu bestimmen. Im Folgenden werden die Modellierungsansätze für kohärente und inkohärente Lichtquellen vorgestellt.

Betrachtet man ein für Licht undurchsichtiges Objekt, welches von einer Punktlichtquelle beleuchtet wird, dann wirft das Objekt hinter sich bekanntermaßen einen Schatten. Vernachlässigt man evtl. Lichtbeugungen an den Kanten und Lichtstreuung von anderen Objekten, dann lässt sich eine Szene klar in zwei Bereiche unterteilen – den beleuchteten Bereich und den (Kern)Schattenbereich. Wird die Lichtquelle ausgedehnt, oder es existieren mehrere Lichtquellen, dann kann es zusätzlich Bereiche in der Szene geben, die nur teilweise beleuchtet werden – die s.g. Halbschatten-Bereiche (siehe Abbildung 3.18).

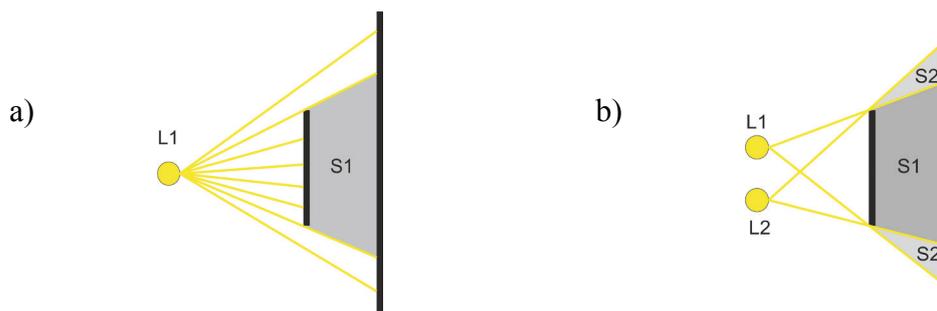


Abbildung 3.18: Lichtstrahlen und Schattenentstehung (S1 – Kernschatten, S2 – Halbschatten); a) Schattenentstehung bei einer Lichtquelle L1, b) Schattenentstehung bei zwei Lichtquellen L1 und L2

Es werden zunächst inkohärente Lichtquellen betrachtet.

Inkohärente Lichtquellen

Die gesamte Beleuchtungsstärke als Funktion der Zeit in einem beliebigen Punkt P der Szene setzt sich additiv zusammen aus den Intensitätswerten der einzelnen Lichtquellen in diesem Punkt.

$$I_{ges,P}(t) = \sum_i I_{i,P}(t) \quad (3.15)$$

$I_{i,P}(t)$ beschreibt dabei die Zeitfunktion der Beleuchtungsintensität des Beleuchters I_i in Richtung des Punktes P . Abbildung 3.19 veranschaulicht das Grundprinzip der Modellierung. Es muss berücksichtigt werden, dass nicht jeder Punkt der Szene von jeder Position aus beleuchtet werden kann. In dem in der Abbildung links abgebildeten Fall

wird z.B. der Punkt P von den Beleuchtern 1, 5 und 6 aufgrund der Lage außerhalb des Öffnungswinkels (der Lichtkeule) nicht beleuchtet. Betrachtet man die Frontpartie des Zielobjektes, dann liegt sie komplett im Halbschattenbereich, kein Bereich davon wird von allen Lichtquellen erreicht oder gar nicht beleuchtet.

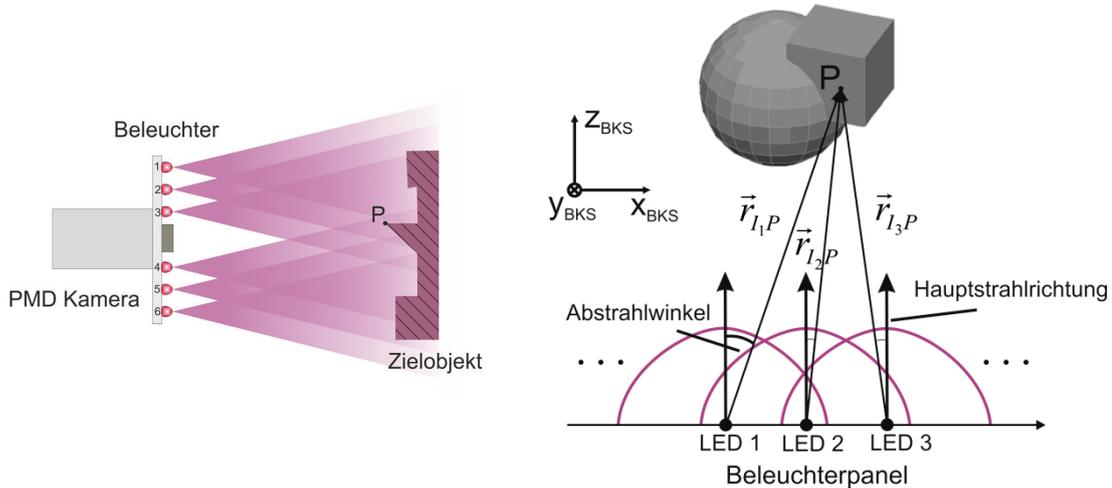


Abbildung 3.19: Modellierung: Ringförmige Beleuchtung der PMD Kamera einer 2D/3D MultiCam (links) und das allgemein geltende Simulationsmodell (rechts)

Die Interaktion aller Punkte P_i mit $i \in [1, N]$ einer Szene mit allen Lichtquellen I_k mit $k \in [1, L]$ kann mit (3.6) und (3.8) als Produkt der entsprechenden Lichtintensitäten in die Richtung des jeweiligen Punktes mit der Gewichtungsmatrix \underline{G} interpretiert werden. Dabei beschreibt die Gewichtungsmatrix die durch die unterschiedlichen Abstände hervorgerufenen Verluste der Strahlungsleistung infolge der Entfernung $r_{I_k P_i}$, sowie die Abschwächung der Leistungsintensität beim Durchqueren des Ausbreitungsmediums.

$$\underline{I} \circ \underline{G} = \begin{pmatrix} I_{I_1 P_1} & I_{I_1 P_2} & \dots & I_{I_1 P_N} \\ I_{I_2 P_1} & I_{I_2 P_2} & \dots & I_{I_2 P_N} \\ \dots & \dots & \dots & \dots \\ I_{I_L P_1} & I_{I_L P_2} & \dots & I_{I_L P_N} \end{pmatrix} \circ \begin{pmatrix} G_{I_1 P_1} & G_{I_1 P_2} & \dots & G_{I_1 P_N} \\ G_{I_2 P_1} & G_{I_2 P_2} & \dots & G_{I_2 P_N} \\ \dots & \dots & \dots & \dots \\ G_{I_L P_1} & G_{I_L P_2} & \dots & G_{I_L P_N} \end{pmatrix} \quad (3.16)$$

◦ – *elementweise Matrizenmultiplikation*

Durch spaltenweise Aufsummierung erhält man die Überlagerung aller partieller Intensitäten – die gesamte Lichtintensität in den jeweiligen Punkten der Szene. Die Punkte der Szene, die von einer Lichtquelle aufgrund ihrer Lage nicht beleuchtet werden können, bekommen den Intensitätswert Null zugewiesen. Für jedes Lichtquelle-Punktziel-Paar muss deswegen das HSR-Problem (Hidden Surface Removal) – die Bestimmung der Sichtbarkeit von der Lichtquelle aus – aufs Neue gelöst werden.

Kohärente Lichtquellen

Bei kohärenten Lichtquellen stehen die Lichtwellen in einer festen Phasenbeziehung zueinander. Laufen zwei solche Wellen in einem Punkt oder auf einem Objekt zusammen, so entsteht, im Gegensatz zu kohärentem Licht, aufgrund der festen Phasenbeziehungen ein bestimmtes festes (bei räumlicher Kohärenz) oder sich zeitlich wiederholendes (bei zeitlicher Kohärenz) Muster – Interferenzbild. Es kann in einem solchen Fall daher keine einfache Überlagerung der Lichtintensitäten vorgenommen werden, die gesamte Lichtintensität muss zeitlich und räumlich unter Berücksichtigung der Phasenverhältnisse bestimmt werden.

Interferenz zweier Wellen

Es werden zwei Wellen gleicher Frequenz mit unterschiedlicher Amplitude und Phase betrachtet.

$$s_1(t) = A_1 \cdot \sin(2\pi ft + \varphi_1) \quad s_2(t) = A_2 \cdot \sin(2\pi ft + \varphi_2) \quad (3.17)$$

Treffen diese Wellen in einem Punkt aufeinander, dann können sie nach dem Superpositionsprinzip überlagert werden. Für die Überlagerung der beiden Wellen gilt dann:

$$s_1(t) + s_2(t) = A \cdot \sin(2\pi ft + \varphi) \quad (3.18)$$

mit

$$A = A_1^2 + A_2^2 + 2A_1A_2 \cdot \cos(\varphi_1 - \varphi_2) \quad (3.19)$$

$$\varphi = \arctan\left(\frac{A_1 \cdot \sin(\varphi_1) + A_2 \cdot \sin(\varphi_2)}{A_1 \cdot \cos(\varphi_1) + A_2 \cdot \cos(\varphi_2)}\right)$$

Sollen drei Wellen überlagert werden, so lässt sich die resultierende Welle als Überlagerung zweier von denen mit der dritten darstellen usw. Man merkt, dass mit steigender Anzahl der Überlagerungen die Anzahl der erforderlichen Berechnungen proportional zunimmt. Ein weiteres Problem stellt die arctan-Funktion dar, die nur im Bereich $(-\pi/2, \pi/2)$ definiert ist. D.h. für eine korrekte Berechnung werden noch weitere Abfragen bezüglich der räumlichen Orientierung der Vektoren benötigt. Eine viel einfacher zu realisierende und vor allem weniger störungsanfällige Lösung bietet die Betrachtung der Überlagerung der amplitudenmodulierten harmonischen Schwingungen in komplexer Form. Diese bietet außerdem den Vorteil, dass die Lichtwellen mit Hilfe der Vektorrechnung beschrieben werden und die Überlagerung allgemein schneller durchgeführt werden kann.

Vektorielle Wellenüberlagerung

Nach Euler kann jeder rotierende Zeiger vektoriell als Summe seiner Projektionen auf die beiden Achsen beschrieben werden.

$$e^{j\theta} = \cos \theta + j \sin \theta \quad \text{bzw.} \quad (3.20)$$

$$\underline{z} = r \cdot e^{j\theta} = r \cos \theta + jr \sin \theta$$

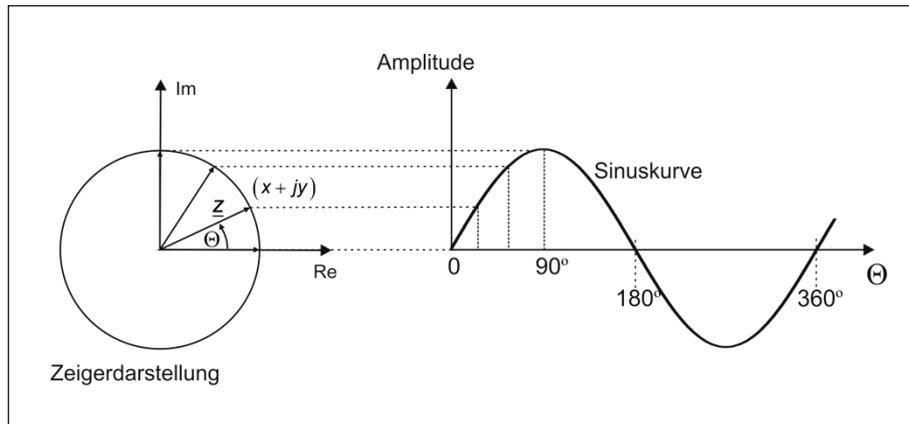


Abbildung 3.20: Zeigerdarstellung einer Sinus-Funktion

Im Folgenden soll das Grundprinzip der Wellenüberlagerung am Beispiel eines amplitudenmodulierten Signals vorgestellt werden. Bei Amplitudenmodulation wird die Amplitude einer hochfrequenten Trägerschwingung f_T abhängig von einem zu übertragenen niederfrequenten Nutzsinal $s(t)$ verändert.

$$m(t) = a_0 \cdot [1 + \mu_{AM} \cdot s(t)] \cdot \cos(2\pi f_T t) \quad (3.21)$$

mit a_0 – Amplitude des unmodulierten Trägersignals, μ_{AM} – Amplitudenmodulationsgrad. In Abbildung 3.21 ist ein Beispiel für ein AM moduliertes Signal mit einem Nutzsinal $s(t) = \cos(2\pi f_s t)$ gezeigt.

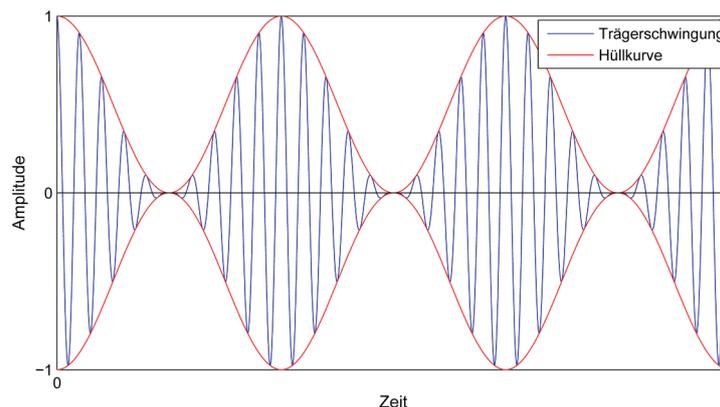


Abbildung 3.21: Amplitudenmodulierte cos-Schwingung ($\mu_{AM} = 1$)

In der Zeigerdarstellung kann eine solche Modulation mit Hilfe von einem Vektor \vec{v}_T der Trägerfrequenz und zwei zusätzlichen Seitenfrequenzdrehvektoren $\frac{1}{2}\vec{v}_s$ realisiert werden [Hecht 89]. Die Seitenfrequenzdrehvektoren stehen auf dem Träger, haben eine

konstante Länge, die der halben Länge des maximalen Amplitudenausschlages entspricht, und drehen sich mit der Frequenz $\omega_s = 2\pi f_s$ in jeweils entgegengesetzte Richtung. Zusätzlich müssen sie in ihrer Richtung angepasst werden, so dass die Summe der beiden Vektoren immer in die gleiche Richtung wie der Vektor der Trägerfrequenz zeigt.

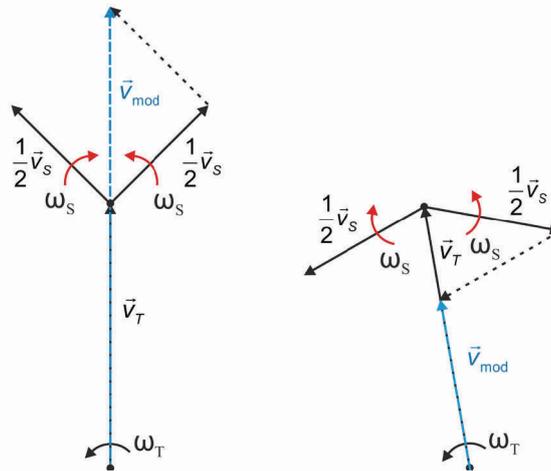


Abbildung 3.22: Darstellung der Amplitudenmodulation in Zeigerform

Die Vektoranteile der Seitenfrequenzdrehvektoren in Richtung senkrecht zum Vektor \vec{v}_T heben sich in der Summe gegenseitig auf. Auf diese Weise wird der Modulationsvektor – die Summe der Seitenfrequenzdrehvektoren mit dem Trägerfrequenzvektor – entsprechend den Modulationsparametern in seiner Länge und seiner Drehposition beeinflusst.

Um den Drehzeiger einer sich ausbreitenden Welle an einem Punkt zu bestimmen, werden die Momentanphase ψ_T der Trägerschwingung und die Momentanphase ψ_s des Nutzsignals benötigt.

$$\psi_T = 2\pi f_T t \quad \psi_s = 2\pi f_s t \quad (3.22)$$

Aus den Phasenwinkeln lassen sich die jeweiligen Vektoren bestimmen. Bei der Berechnung der Interferenz aller Wellen müssen somit lediglich die einzelnen Modulationsvektoren aufsummiert werden:

$$\vec{v}_{res} = \sum_L \vec{v}_{mod,L} \quad (3.23)$$

Betrachtet man lichtintensitätsmodulierte Signale, so stellen die analog zu (3.16) gewichtete Beträge der Modulationsvektoren die gesuchte Lichtintensität dar.

$$I_{res,P_N} = \left| \sum_L \left(\vec{I}_{mod,L,P_N} \circ \underline{G}_{I_{LP_N}} \right) \right| \quad (3.24)$$

3.3.4 Ablaufsteuerung

Die am Anfang des Kapitels behandelten Anforderungen und die gegebenen theoretischen Modellierungsansätze wurden in dem im Rahmen dieser Arbeit entwickelten *Light Propagation Module* umgesetzt. Abbildung 3.23 zeigt die wichtigsten Ablaufschritte innerhalb des Moduls. Die einzelnen Blöcke dienen zur besseren Verständlichkeit des Ablaufs und stellen kein Flussdiagramm im programmiertechnischen Sinne dar.

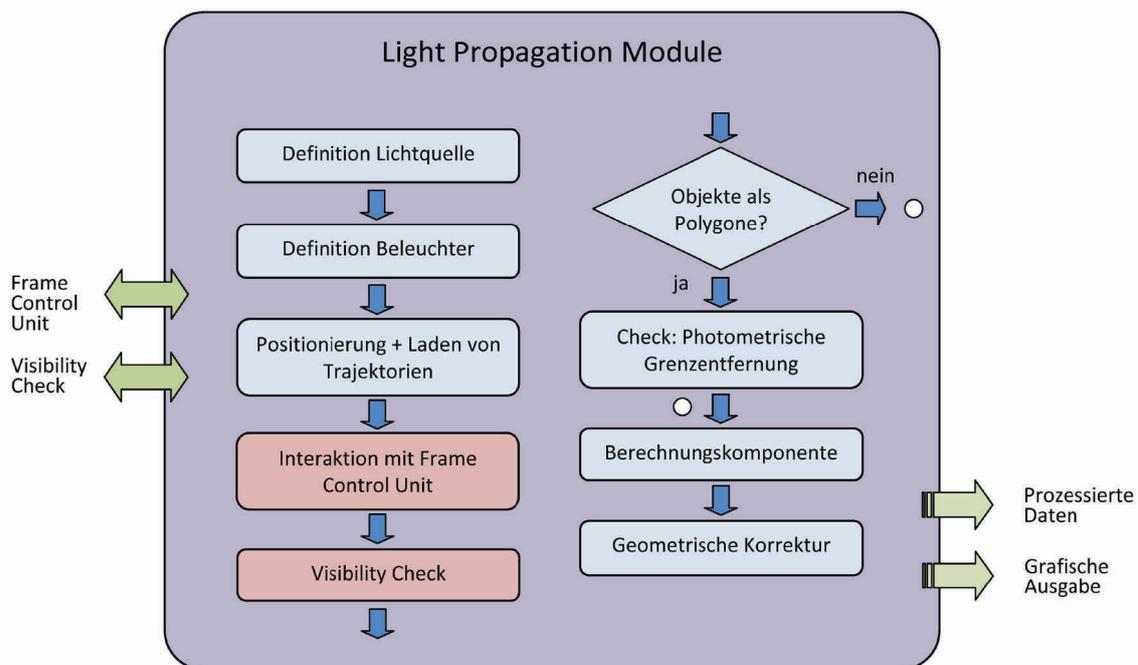


Abbildung 3.23: Schemenhafte Darstellung der Ablaufsteuerung des Light Propagation Moduls

1. Initialisierung

Der Programmablauf beginnt mit der Definition der Simulationsszene. Es werden die Abstrahlcharakteristika der Lichtquellen tabellarisch oder funktional festgelegt und daraus starre Beleuchteranordnungen modelliert. Die Beleuchter werden an ihre Ausgangsposition gesetzt. Anhand der vordefinierten Vorgaben für die Startanordnung werden Beleuchter-Koordinatensysteme definiert und mit dem Weltkoordinatensystem verankert. Bei Simulationen dynamischer Beleuchter-Anordnungen werden für die einzelnen Beleuchter die Trajektoriendateien geladen. Von der *Frame Control Unit* (FCU) werden die Datensätze der Szene angefordert. Im Independent Mode (siehe Anfang des Kapitels) werden die Szenen „von Hand“ geladen, die Schnittstelle zu FCU wird geschlossen. Mit der Definition der Position der Kamera – die Richtung in welche die Strahlungsstärke bestimmt werden soll – ist die Initialisierung abgeschlossen und die eigentliche Simulation kann gestartet werden.

2. Berechnung

Anhand der Positionen der einzelnen Lichtquellen wird mit Hilfe der Sichtbarkeitsanalyse berechnet, welche Teile der Szene von der jeweiligen Lichtquelle beleuchtet werden können. Dabei wird unterschieden, ob Objekte in Polygonform oder als Punktwolke definiert vorliegen. Liegen die Punkte in Polygonform vor, so bietet der Simulator im Falle der Unterschreitung der photometrischen Grenzentfernung die Funktionalität an, die Genauigkeit der Berechnung zu erhöhen, indem die nicht mehr ausreichend fein aufgelösten Szenenobjekte interpoliert werden. Dabei wurde der Schwerpunkt auf die Realisierung einer möglichst einfachen und daher schnellen Interpolation gelegt – lineare Interpolation. Abbildung 3.24 veranschaulicht das Prinzip der stattfindenden linearen Interpolation.

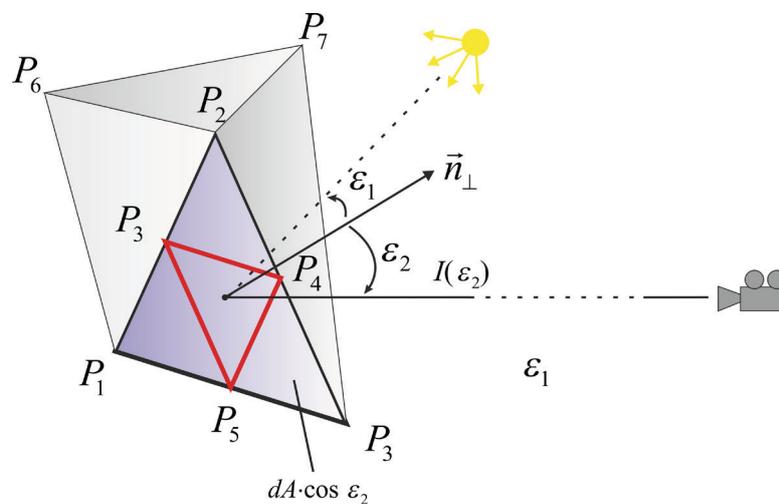


Abbildung 3.24: Das Prinzip der Objektinterpolation bei Unterschreitung der photometrischen Grenzentfernung

Wird ein Polygon als nichtausreichend fein eingestuft (die Entfernung zum Polygon ist geringer als die 10-fache größte Ausdehnung des Polygons), dann wird seine Fläche in vier Teilpolygone unterteilt. In der Abbildung ist ein Beispiel für ein Triangle-Polygon dargestellt, das Prinzip bei Quads (Vierecke) wäre das gleiche. Die neuen Eckpunkte der Teilpolygone werden in die Mitte der Kanten des ursprünglichen Polygons gelegt. Die Interpolation wird solange ausgeführt, bis die Bedingung der photometrischen Grenzentfernung erfüllt ist.

Die polygonweise Realisierung der Interpolation bietet entscheidende Vorteile im Hinblick auf die benötigte Rechenzeit. So müssen nicht alle Teile der Simulationsszene zeitaufwendig interpoliert werden, sondern nur diejenigen, wo der maximale Rechenfehler bei der Bestimmung der Strahlungsstärke die 0,25% Grenze überschreiten würde. Außerdem stellt dieser Lösungsansatz einen dynamischen Vorgang dar. Bewegt sich ein Objekt auf die Lichtquelle zu, so wird früher oder später der Interpolationsvorgang gestartet, entfernt sich das Objekt wieder von der Lichtquelle, dann wird irgendwann die photometrische Grenzentfernung überschritten und die Interpolation findet nicht statt.

Um Reflektionen in Richtung der Kamera zu bestimmen, wird der berechnete Datensatz einer zweiten Sichtbarkeitsanalyse unterzogen. Dabei wird nun bestimmt, welche Teile der beleuchteten Szene von der Sensorposition aus detektiert werden können. Für die beleuchteten und detektierbaren Bereiche der Szene werden anschließend für jeden einzelnen Punkt der Szene die Beleuchtungsintensitätswerte aus Überlagerung der Beleuchtungen aller Lichtquellen berechnet. Bei polygonalen Datensätzen werden die in den Polygonecken berechneten Werte anschließend einer *geometrischen Korrektur* unterzogen. Diese beinhaltet zwei Schritte bzw. Ideen:

- 1) Wie in Kapitel 5.1 ausführlicher behandelt, wird ein Punkt (Knoten) oft von mehreren Polygonen (Facetten) geteilt. So wird der Knoten P_2 in Abbildung 3.24 von vier Facetten eingeschlossen. Näherungsweise kann man sagen, dass sich die Beleuchtungsintensität in dem Knoten P_2 aus den Beleuchtungsintensitäten der umliegenden Bereiche zusammensetzt. Die Beleuchtungsintensitäten in den Knoten einer Facette werden entsprechend der Beleuchtungsintensität in dem mathematischen Schwerpunkt einer Facette zugewiesen. Der Schwerpunkt $P_{SP}(x_{SP}, y_{SP}, z_{SP})$ einer konvexen Facette mit den Eckpunkten $P_i(x_i, y_i, z_i)$ kann mit Hilfe der Vektoralgebra leicht bestimmt werden:

$$x_{SP} = \frac{1}{N} \sum_{i=1}^N x_i, \quad y_{SP} = \frac{1}{N} \sum_{i=1}^N y_i, \quad z_{SP} = \frac{1}{N} \sum_{i=1}^N z_i \quad (3.25)$$

N – Anzahl der Knoten einer Facette

Die Beleuchtungsintensität in einem Knoten P_N wird dementsprechend aus den umliegenden Intensitäten in den einzelnen Facetten wie folgt bestimmt:

$$I_{sim} = \frac{1}{m} \sum_m I_{P_N} \quad (3.26)$$

m – Anzahl der umliegenden Facetten

- 2) Unterscheidet sich die Position der Kamera von der Position der Lichtquelle, dann muss das in die Kamerarichtung zurückgestrahlte Licht nach dem photometrischen Grundgesetz, siehe (3.12), entsprechend gewichtet werden.

3.3.5 Testscenario Beleuchtung

Zur Begutachtung des Modellierungsansatzes und zur Verifikation der fehlerfreien programmtechnischen Umsetzung werden im Folgenden die Ergebnisse einiger Simulationen betrachtet. Es wird zunächst die reine Beleuchtung der Simulationsszene betrachtet, Simulation einer PMD-Kamera kommt vorerst nicht zum Einsatz. Als Testobjekte werden einfache Figuren gewählt. Sie erlauben einerseits eine einfachere qualitative Beurteilung der Ergebnisse und andererseits ermöglichen sie bei entsprechender Positionierung und Orientierung bezüglich der Beleuchter die Begutachtung der Korrektheit der intern ablaufenden Koordinatentransformationen – bei symmetrischen Anordnungen werden symmetrische Beleuchtungszustände erwartet.

Simulation: Zwei inkohärente Lichtquellen

Beim ersten Testscenario werden zwei inkohärente Lichtquellen auf eine Detektorplatte mit der Größe 3×8 m gerichtet. Der Abstand zwischen den beiden Lichtquellen beträgt 4,4 m, die örtlichen Koordinaten der Lichtquellen liegen bei $(-2,2 \ 0 \ 0)$ und $(2,2 \ 0 \ 0)$ m. Der Abstand jeder Lichtquelle zur Platte hin beträgt 3 m. Alle Koordinaten, sowie die Lichtintensitätsbilder werden im Weltkoordinatensystem betrachtet. In den Szenarios mit Lichtstrahlern stehen die Hauptrichtungsvektoren der Strahlungskeule senkrecht auf die Detektorplatte.

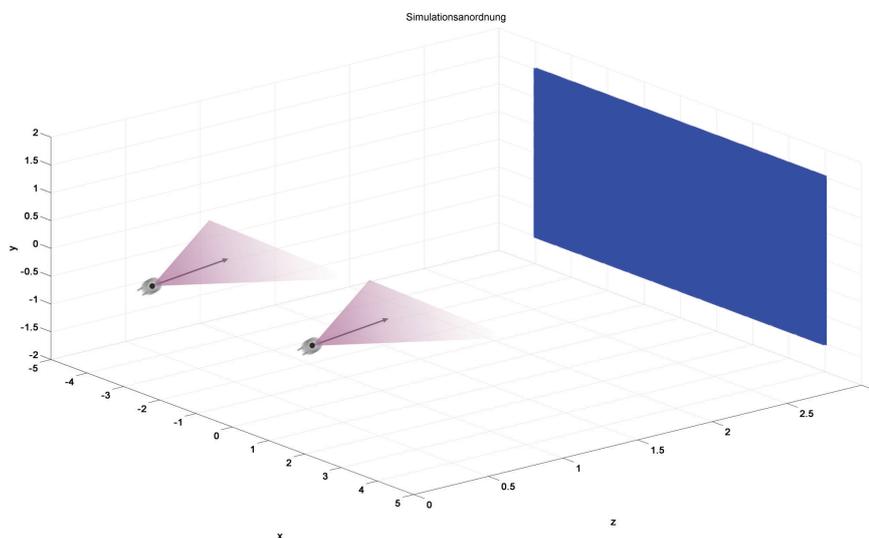


Abbildung 3.25: Simulationsszenario: zwei inkohärente Lichtquellen vor einer Detektorplatte

Abbildung 3.26 zeigt die relative Lichtintensitätsverteilung auf der Detektorplatte. Als Lichtquellen wurden zwei Punktstrahler gewählt, die ausgestrahlte Lichtleistung ist also in alle Richtungen gleich. Die zum Rand hin abfallende Intensität entsteht aufgrund des nach außen hin größer werdenden Abstandes.

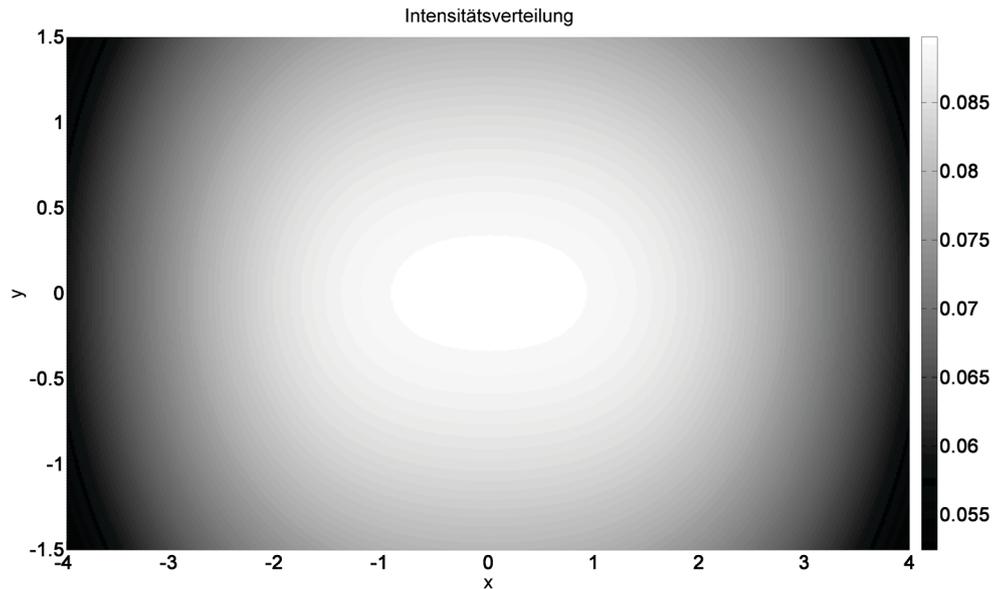


Abbildung 3.26: Simulationsszenario: zwei inkohärente Punktstrahler (\cos^0) vor einer Detektorplatte

In Abbildung 3.27 ist der Lichtintensitätsverlauf auf der Detektorplatte beim Bestrahlen mit zwei Lambert-Strahlern dargestellt. Die von jeder Lichtquelle hervorgerufenen Helligkeitsbereiche sind deutlicher zu erkennen.

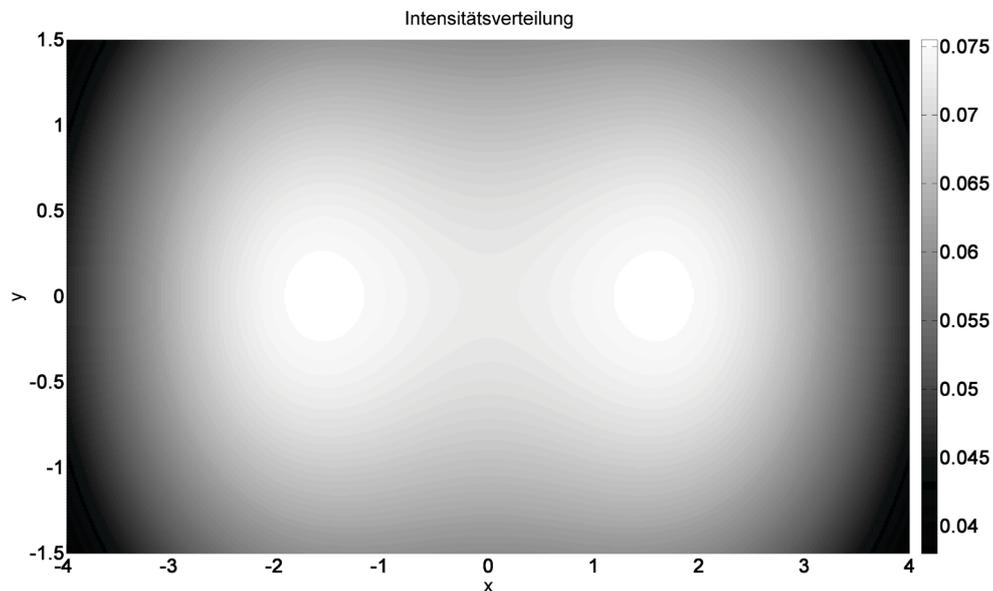


Abbildung 3.27: Simulationsszenario: zwei Lambert-Strahler (\cos^1) vor einer Detektorplatte

In Abbildung 3.28 sind die Ergebnisse der Überlagerung der Lichtkeulen zweier \cos^{12} -Strahler zu sehen. Die Abstrahlcharakteristik dieser Strahler entspricht in guter Näherung der Abstrahlcharakteristik einer PMD19k Kamera (wird in diesem Kapitel später behandelt).

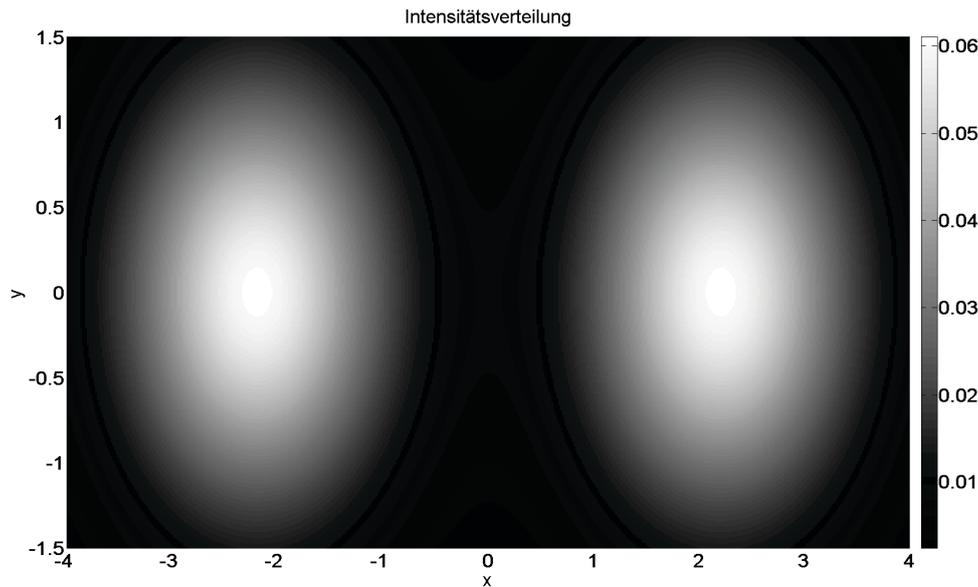


Abbildung 3.28: Simulationsszenario: zwei \cos^{12} -Richtstrahler vor einer Detektorplatte

Man erkennt sehr gut, dass an den Positionen der beiden LED's, also bei -2.2 m und $+2.2\text{ m}$ in x -Richtung, sich aufgrund der höheren Bündelung der beiden Lichtstrahlen zwei deutlich ausgeprägte helle Bereiche ergeben. Das Verhältnis der Lichthelligkeit in den hellsten Bereichen zu der Lichthelligkeit in den Ecken der Detektorplatte beträgt $\eta_{\cos^{12}} = I_{\max}/I_{\min} = 25,4$. Würde man eine solche Platte vom Ursprung aus mit einer PMD-Kamera vermessen wollen, und würde der Öffnungswinkel der Kamera die Platte exakt einschließen, dann würde der Unterschied des zu detektierenden einfallenden Lichtes noch wesentlich größer ausfallen. Der Grund dafür ist, dass in diesen Simulationen die Ausbreitung des Lichtes lediglich in eine Richtung betrachtet wurde. Angegeben sind also die Werte die vor Ort gemessen werden können. Die zurückgestrahlte Energie wird auf dem Weg zur Kamera hin streng genommen noch drei weiteren Einflüssen unterliegen, die das Verhältnis noch deutlich verschlechtern – der Abschwächung der Intensität aufgrund des Abstandes, der Abschwächung der Intensität aufgrund des Winkels zur Kamera hin (photometrisches Grundgesetz) und durch Energieverluste beim Durchqueren des Ausbreitungsmediums, wobei die letzteren in der Luft eher zu vernachlässigen sind.

Betrachtet man dagegen zwei Punktstrahler, dann beträgt das Verhältnis der Lichthelligkeit in den hellsten Bereichen zu der Lichthelligkeit in den Ecken der Detektorplatte $\eta_{\cos^0} = I_{\max}/I_{\min} = 1,71$. In Abbildung 3.29 sind zur besseren Vergleichbarkeit die relativen Lichtintensitäten normiert dargestellt.

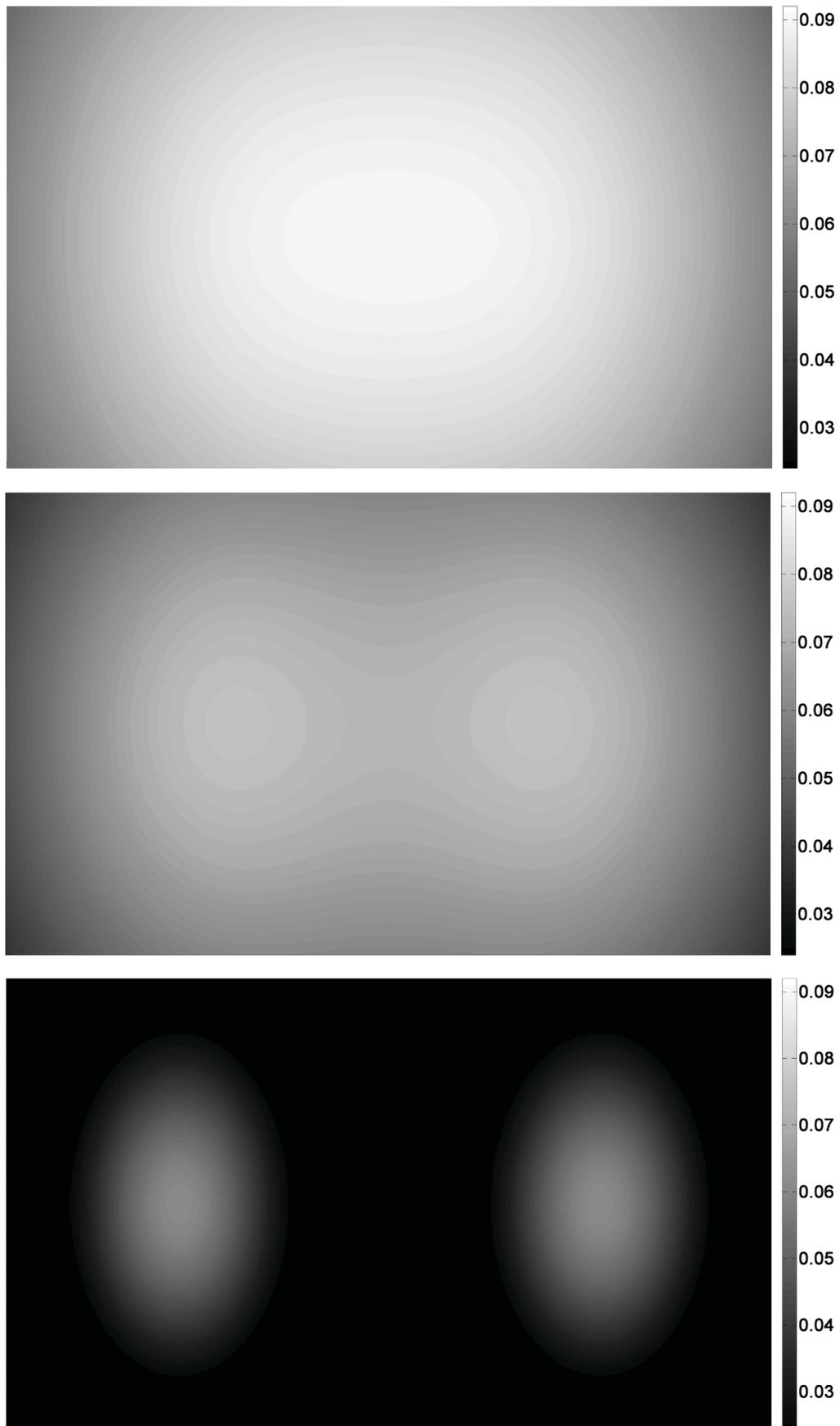


Abbildung 3.29: Normierte Lichtintensitätsverläufe für (von oben nach unten) Punktstrahler, Lambert-Strahler und \cos^{12} -Richtstrahler

Aus diesem Beispiel wird klar, wie groß die Beleuchtungsunterschiede selbst an einer flachen Oberfläche sein können. Das gezeigte Simulationsszenario mit dem für eine bessere Darstellung gewählten sehr großen Abstand zwischen den beiden Beleuchtern ist sicherlich nicht repräsentativ für ein übliches PMD-System. In der Regel liegt der Abstand im Bereich 10-100 cm. Betrachtet man jedoch die Intensitätsunterschiede im Nahbereich der Kamera, dann entsprechen die hier gezeigten Unterschiede durchaus den üblichen Werten. Bei komplizierten Szenen mit Wölbungen und/oder Kanten sind die zu detektierenden Unterschiede infolge der Abschwächung der Intensität aufgrund des Winkels zur Kamera hin wesentlich höher. Das folgende Beispiel soll dies verdeutlichen.

Simulation: Eine inkohärente Lichtquelle

In diesem Beispiel werden die zu detektierenden Lichtintensitätsunterschiede auf der Oberfläche einer Kugel betrachtet. Es wird ein monostatisches Simulationsszenario angenommen, d.h. der Beleuchter – ein Lambert-Strahler – ist am selben Ort positioniert wie die PMD-Kamera. Die Kamera und Beleuchter sind 5 m von der Kugel entfernt, die Kugel hat den Radius $R=1\text{m}$ und wird als Lambert-Reflektor angenommen. Betrachtet wird nur das Helligkeitsbild im Weltkoordinatensystem.

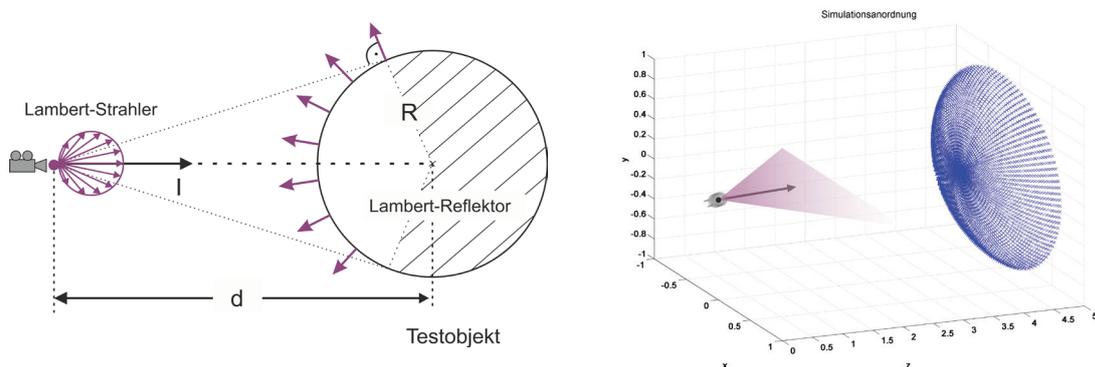


Abbildung 3.30: Monostatisches Simulationsszenario: ein Lambert-Strahler vor einer Kugel mit Reflektionseigenschaften eines Lambert-Reflektors

Beleuchtet und detektiert wird aufgrund des monostatischen Aufbaus ein und dasselbe Kugelsegment (siehe Abbildung 3.30 links), der gestrichelte Bereich der Kugel liegt im Schatten und wird daher nicht beleuchtet.

In Abbildung 3.31 ist der detektierte Intensitätsverlauf dargestellt. Während im mittleren Bereich die größte Lichtintensität zu verzeichnen ist, nimmt sie zum Rand hin schnell ab. Dies geschieht einerseits aus den im vorangegangenen Test behandelten Gründen und andererseits aus dem Grund der zusätzlich kontinuierlich abnehmenden Reflektivität, da der Winkel zwischen der Normalen auf die Oberfläche und der Reflektionsrichtung zunimmt. Bei einem Winkel von 90° wird nichts mehr zurückreflektiert, die Oberfläche verläuft dann parallel zum Wellenausbreitungsvektor.

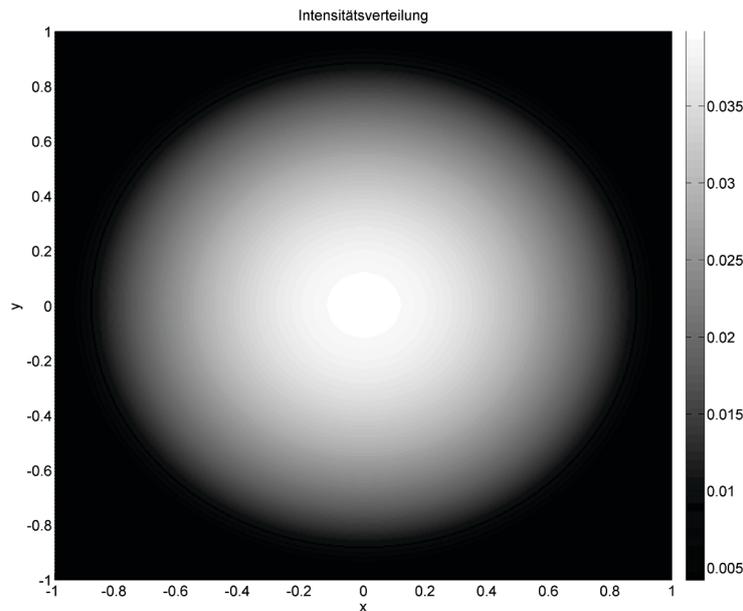


Abbildung 3.31: Detektierter Lichtintensitätsverlauf bei Annahme eines Lambert-Strahlers vor einer Kugel

Vergleich mit realen Daten

Zur Beurteilung der Simulationsergebnisse des *Light Propagation Module* wird ein Vergleich zwischen realen Messwerten und simulierten Daten angestellt. Als Testobjekt diente eine Infrarot-LED, deren Lichtintensität auf der Oberfläche einer Detektorplatte punktuell in einem festen Raster aufgenommen wurde. Die Abbildung 3.32 zeigt die gemessene Lichtintensitätsverteilung.

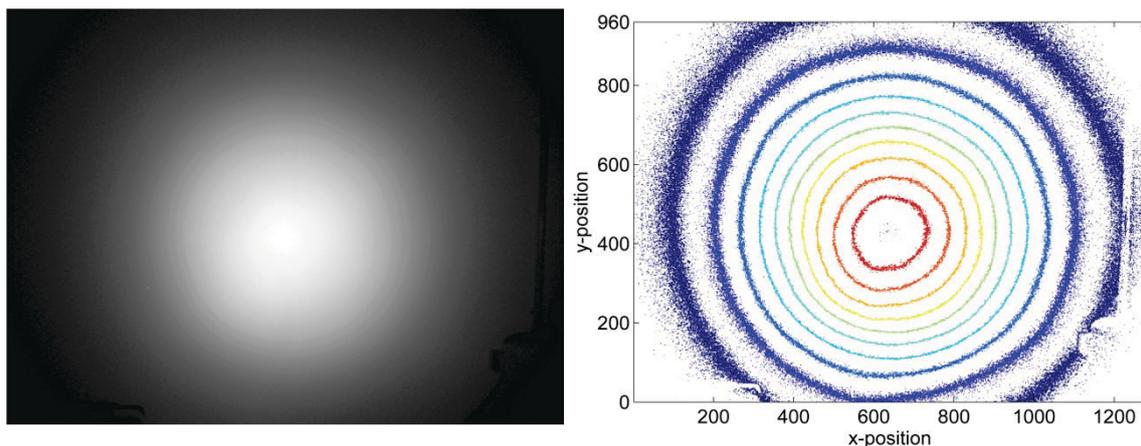


Abbildung 3.32: Gemessene Lichtintensitätsverteilung an einer Detektorplatte (links) und Farbdarstellung der Lichtintensitätsverteilung (jede Farbe repräsentiert eine Intensitätsstufe)

Wie man in der Farbdarstellung erkennen kann, sind die einzelnen Schnitte durch die Verteilungsfunktion nur näherungsweise kreisförmig. Es fällt auf, dass das Rauschen mittig, also in der Hauptrichtung der Strahlungskeule, etwas stärker ist. Danach liegen

die Werte fast ideal kreisförmig vor und sind wenig verrauscht und zum Rand hin steigt die Streuung wieder deutlich an.

Anhand der gemessenen Werte lässt sich im Hinblick auf die Minimierung des Simulationsfehlers eine Abstrahlcharakteristik bestimmen und im Simulator definieren. In Abbildung 3.33 ist ein Vergleich der simulierten Daten mit gemessenen Werten dargestellt. Das linke Bild zeigt nochmal die gemessene Lichtintensitätsverteilung in 3D-Darstellung. Die Höhe des Graphen repräsentiert die auf den maximalen Wert normierte Lichtintensität. Die Anpassung des Simulators wurde so durchgeführt, dass eine hohe Übereinstimmung im mittleren Bereich der Lichtkeule stattfindet. Am Rand ist die Lichtintensität stark verrauscht, außerdem befindet sich dieser Bereich in gängigen PMD-Kameras meistens außerhalb des Kamerawinkels.

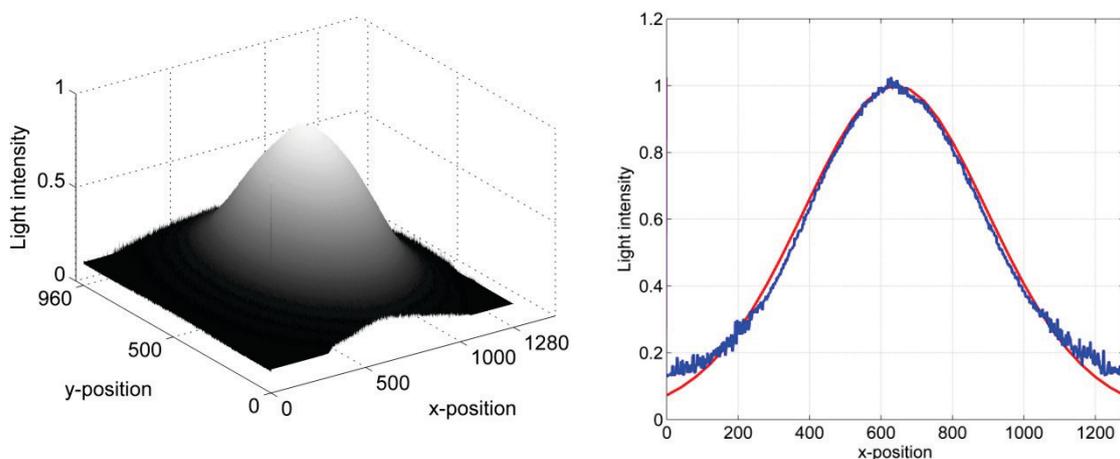


Abbildung 3.33: Links: Lichtintensitätsverteilung in 3D Darstellung;
Rechts: radialer Schnitt durch die Messwerte (blau) und das Simulationsergebnis (rot)

Bei der Simulation wurde versucht, eine analytische Funktion zu bestimmen, mit deren Hilfe die gemessene Abstrahlcharakteristik möglichst genau beschrieben werden kann. Die Grundidee dabei ist folgende: lässt sich die gemessene Intensitätsverteilung mit einer mathematisch gesehen relativ einfachen Funktion beschreiben, dann kann die Funktion der Abstrahlcharakteristik im Simulator durch eine solche Funktion beschrieben werden. Dies würde, im Gegensatz zu tabellarisch vorliegenden diskreten Werten (vgl. Tabelle 3.2), die im Simulator oft vorkommenden Abfragen der Tabellenwerte mit anschließender Interpolation für den zutreffenden Winkel überflüssig machen und somit der Rechengeschwindigkeit zugutekommen.

Wie man in der Abbildung rechts erkennen kann, bietet ein \cos^{12} -Strahler eine ziemlich gute Näherung für die vermessene LED. Die blaue Funktion stellt einen radialen Schnitt durch die gemessene Intensitätsverteilungsfunktion, die rote Funktion – einen Schnitt durch die mit einem \cos^{12} -Strahler simulierte Lichtintensitätsverteilung. Unter einem \cos^{12} -Strahler wird ein Richtstrahler gemeint, dessen Abstrahlcharakteristik im Winkelbereich von -90° bis 90° durch eine $\cos^{12}(\alpha)$ gegeben ist. In Abbildung 3.34 ist die

normierte Abstrahlcharakteristik eines solchen Strahlers und zum Vergleich die eines Lambert-Strahlers dargestellt.

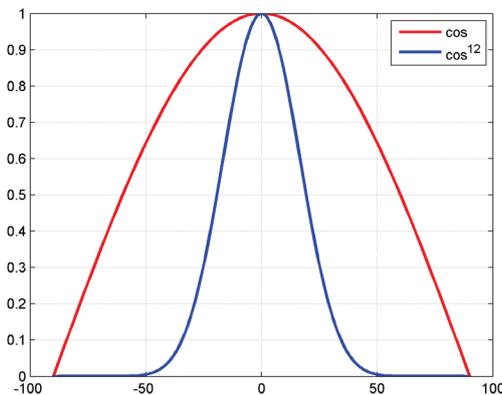


Abbildung 3.34: Abstrahlcharakteristika eines Lambert-Strahlers (\cos^1) und eines \cos^{12} -Richtstrahlers

Simulation: Zwei kohärente Lichtquellen

Nun wird ein Simulationsszenario mit zwei kohärenten Lichtquellen betrachtet. Die Simulationsanordnung ist ähnlich der in Abbildung 3.25 gezeigten. Zur besseren Darstellung der Überlagerung von zwei IR-Wellen wurden die geometrischen Ausmaße wie folgt gewählt. Der Abstand zwischen den beiden Lichtquellen beträgt 16 cm, das Interferenzmuster wird auf einer Detektorplatte mit der Größe 0,5x0,5 m betrachtet, welche in einem Abstand von 7 m zentral vor den beiden Lichtquellen platziert ist. Die Modulationsfrequenz beträgt 20 MHz, die Beleuchtungswellenlänge ist 870 nm.

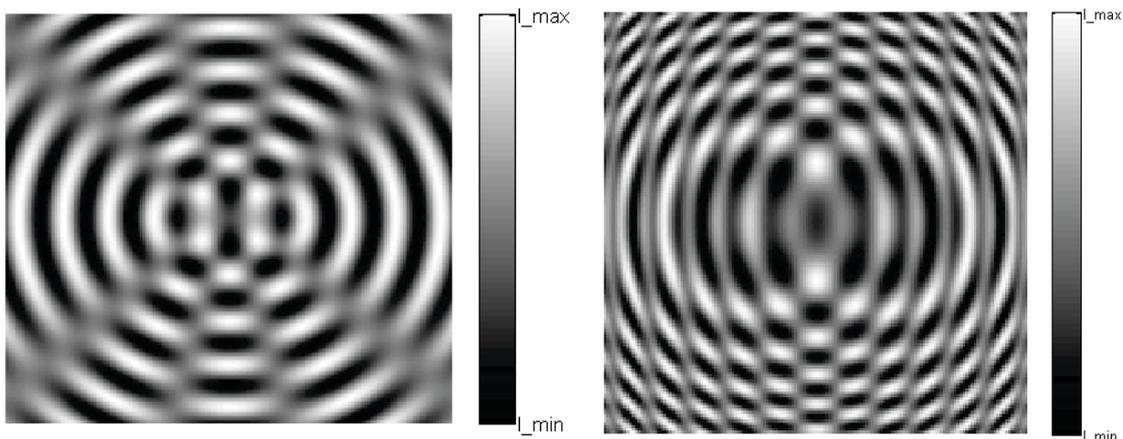


Abbildung 3.35: Interferenzmuster: links – zwei unmodulierte Lichtquellen, rechts – zwei amplitudenmodulierte Lichtquellen

Abbildung 3.35 zeigt zwei Interferenzmuster, die bei Überlagerung der Lichtwellen entstehen. In der Abbildung links handelt es sich um zwei nichtmodulierte Lichtquellen. Das Interferenzmuster entsteht nur aufgrund unterschiedlicher räumlicher Position der Punkte der Detektorplatte bezüglich der beiden Lichtquellen und somit unterschiedli-

chen den Abständen entsprechenden Signalphasen und -amplituden in diesen Punkten. In der Abbildung rechts ist eine Momentaufnahme der Überlagerung zweier mit 20 MHz amplitudenmodulierten Wellen dargestellt. Zu dem bereits besagten Effekt kommen Interferenzen aufgrund der Modulation hinzu.

4 Koordinatensysteme

Wie bereits erwähnt, bieten vektorbasierte Lösungsansätze bei PMD-Simulationen in den Bereichen wie Lichtausbreitung, Lichtüberlagerung, Signalübertragung, Sensorsimulation und Signalaufnahme viele Vorteile. Betrachtet man die gestellten Anforderungen bezüglich der gewünschten Freiheitsgrade, dann wird schnell erkennbar, dass für eine überschaubare Modellierung und systemtheoretische Beschreibung der einzelnen Schritte eine Reihe von Koordinatensystemen benötigt wird. Im Folgenden sollen die wichtigsten Koordinatensysteme vorgestellt werden, die bei der Modellierung eines PMD-Systems eingeführt wurden. Auf Darstellung aller Hilfskoordinatensysteme oder Koordinatensysteme, die im Simulator nur temporär eingesetzt werden, um bestimmte Operationen für den Entwickler überschaubarer zu machen, wird aus Gründen der Übersichtlichkeit verzichtet.

4.1 Weltkoordinatensystem

Das Weltkoordinatensystem (WKS), oder auch Globalkoordinatensystem (GKS) genannt, bildet ein fest mit der Welt (Fussboden) verbundenes kartesisches Koordinatensystem. In diesem Koordinatensystem werden hauptsächlich die Positionen der Kamera, der Beleuchter und der zu vermessenden Szene beschrieben. Prinzipiell kann die Ausrichtung der Achsen beliebig gewählt werden, es bietet sich jedoch an, zwei Achsen parallel zum Boden zu definieren, dann steht die dritte senkrecht auf den beiden. Im Simulator ist das Weltkoordinatensystem so definiert, dass die Achsen x_{WKS} und z_{WKS} parallel zum Boden verlaufen, d.h. eine Linie parallel der Ebene (x_{WKS}, z_{WKS}) würde auf dem Bildschirm waagrecht dargestellt. Die y_{WKS} -Achse vervollständigt ein Rechtssystem. Abbildung 4.1 veranschaulicht die Definition.

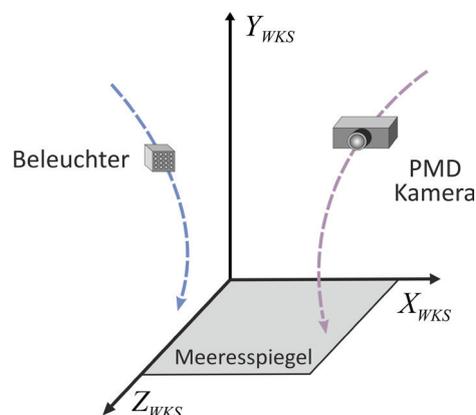


Abbildung 4.1: Definition des Weltkoordinatensystems

4.2 Kamerafestes Koordinatensystem

Das kamerafeste Koordinatensystem (KKS), oder auch Kamerakoordinatensystem genannt, ist ein rechtshändiges orthonormales Koordinatensystem, welches mit der sich bewegenden Kamera bzw. dem Kameragehäuse verknüpft ist. Es wird angenommen, dass sich der Ursprung des KKS im Zentrum der Optik befindet. Die z_{KKS} -Achse fällt mit der optischen Achse der Kamera zusammen – der Blickrichtung aus der Kamera heraus in die Szene. So haben alle Punkte, die von der Kamera aus sichtbar sein können, in diesem Koordinatensystem positive z_{KKS} -Koordinaten (Mindestkriterium).

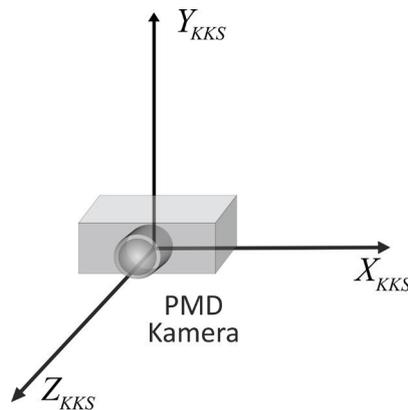


Abbildung 4.2: Definition des kamerafesten Koordinatensystems

Die y_{KKS} -Achse zeigt nach oben, die x_{KKS} -Achse vervollständigt ein Rechtssystem.

4.3 Trajektorienkoordinatensystem

Für die Beschreibung der Beleucher- und Kameraneigungen und partieller Geschwindigkeitsvektoren wird das Trajektorienkoordinatensystem (TKS) eingeführt.

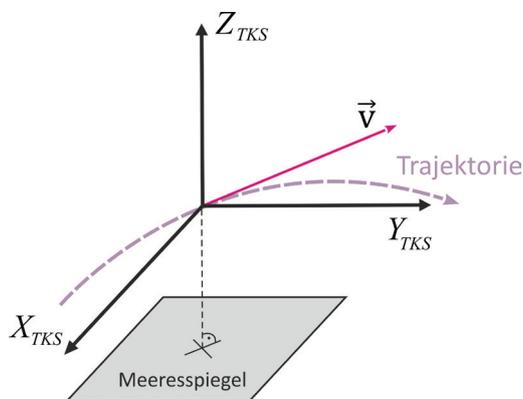


Abbildung 4.3: Definition des Trajektorienkoordinatensystems

Der Ursprung des TKS befindet sich im Zentrum der Optik bzw. des Beleuchters und fällt somit zusammen mit dem Ursprung des kamerafesten Koordinatensystems bzw. des Beleuchter-Koordinatensystems. Die z_{TKS} -Achse entspricht dem normalen Vektor auf der Ebene (x_{WKS}, z_{WKS}) , also auf dem Boden. Die Achse x_{TKS} steht senkrecht auf der Ebene, die durch den Geschwindigkeitsvektor \vec{v} und die z_{TKS} -Achse festgelegt wird, d.h.

$$\vec{e}_{x_{TKS}} = \frac{\vec{v} \times \vec{z}_{TKS}}{|\vec{v} \times \vec{z}_{TKS}|}$$

Die y_{TKS} -Achse vervollständigt ein Rechtssystem.

4.4 Sensorkoordinatensystem

Im Sensorkoordinatensystem (SKS) werden die Positionen, die Form, die Orientierung und die Größe der einzelnen Pixel definiert und beschrieben.

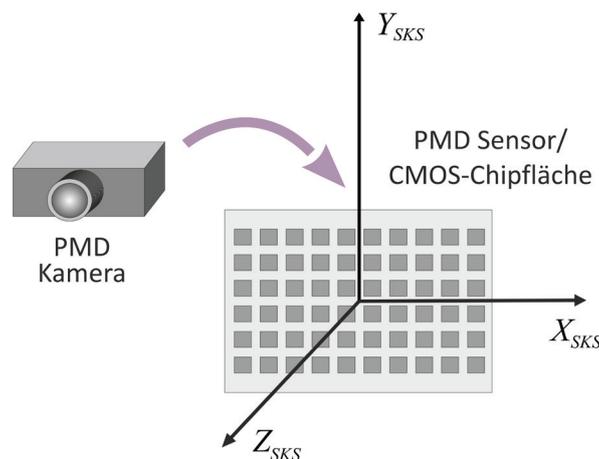


Abbildung 4.4: Definition des Sensorkoordinatensystems

Der Ursprung des SKS befindet sich in der Mitte des PMD-Chips, die Ebene (x_{SKS}, y_{SKS}) repräsentiert somit die lichtempfindliche Sensoroberfläche. Die Zeilen und die Spalten des PMD-Sensors verlaufen entsprechend parallel zu der x_{SKS} - bzw. der y_{SKS} -Achse. Die z_{SKS} -Achse steht senkrecht auf der Sensoroberfläche und zeigt in die Aufnahmerichtung der Kamera.

Genau betrachtet würde für die geforderten Definitionen einer flachen Sensoroberfläche auch ein 2D-Koordinatensystem genügen. Es wurde jedoch im Hinblick auf besser überschaubare Koordinatentransformationen an dieser Stelle ein 3D-Koordinatensystem eingeführt. Die z-Koordinaten der Pixel haben somit im Sensorkoordinatensystem den Wert Null.

4.5 Optikkoordinatensystem

Im Optikkoordinatensystem (OKS) werden Abbildungsberechnungen zwischen den im Weltkoordinatensystem definierten Simulationsszenen und dem im Sensorkoordinatensystem definierten PMD-Sensor berechnet. Die z_{OKS} -Achse fällt mit der optischen Achse der Linse bzw. der Linsen zusammen. Die Richtungen der x_{OKS} - und der y_{OKS} -Achse können wegen der Rotationssymmetrie der optischen Abbildung beliebig definiert werden. Bei Annahme einer Bikonvexlinse wird die Definition der Achsen entsprechend an das Kamerakoordinatensystem angelehnt.

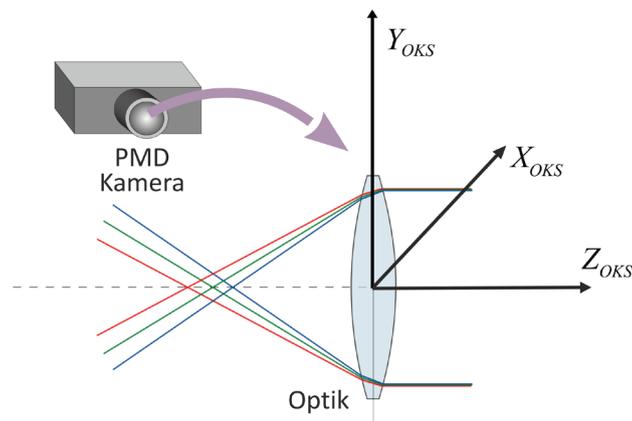


Abbildung 4.5: Definition des Optikkoordinatensystems

4.6 LED-Koordinatensystem

Das LED-Koordinatensystem (LKS) ist ein kartesisches Koordinatensystem, welches hauptsächlich zur Berechnung der Strahlungsintensität in Richtung eines bestimmten Punktes im Raum anhand der Definition der Abstrahlcharakteristik dient.

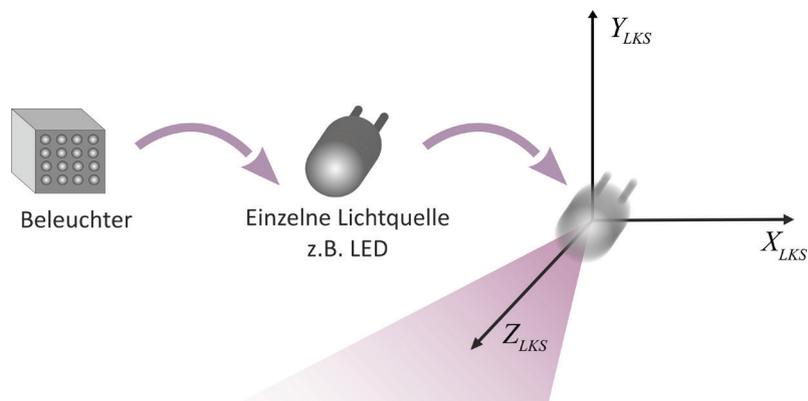


Abbildung 4.6: Definition des LED-Koordinatensystems

Der Koordinatenursprung liegt im Massenschwerpunkt der Lichtdiode. Die z_{LKS} -Achse zeigt in die Hauptrichtung der Strahlungskeule der LED. Die Richtungen der x_{SKS} - und der y_{SKS} -Achse können, ähnlich wie beim OKS, wegen der angenommenen Rotations-symmetrie der Strahlungscharakteristik beliebig definiert werden.

4.7 Beleuchter-Koordinatensystem

Im Beleuchter-Koordinatensystem (BKS) werden die Positionen und die Orientierung der LED's definiert und beschrieben. Die Definition des Koordinatensystems ist der Definition des Sensorkoordinatensystems sehr ähnlich.

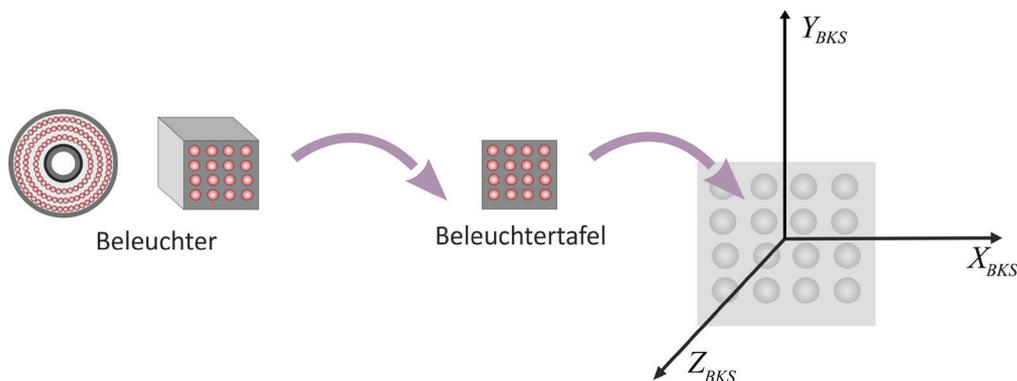


Abbildung 4.7: Definition des Beleuchter-Koordinatensystems

Der Ursprung des BKS befindet sich in der Mitte des Beleuchters. Die z_{SKS} -Achse steht senkrecht auf der Beleuchtertafel und zeigt in die Strahlrichtung des Beleuchters. Die Richtungen der beiden anderen Achsen sind nicht an bestimmte Ausrichtungen im Beleuchter gebunden und sind in der Ebene frei definierbar. Es empfiehlt sich jedoch, wie im Kapitel 3.3.2 erwähnt, die Koordinaten der einzelnen LED's achsensymmetrisch zum Ursprung des Beleuchterkoordinatensystems festzulegen.

4.8 Koordinatentransformationen

Da verschiedene Objekte und Operationen in verschiedenen Koordinatensystemen definiert und ausgeführt werden, wird im Simulator eine Reihe an Koordinatentransformationen benötigt. In diesem Kapitel werden die wichtigsten davon behandelt.

Grundlegende Modellierung

Simulationsszenen werden im Weltkoordinatensystem (WKS) definiert und beschrieben. Um die Abbildung dieser Szene auf dem Sensor berechnen zu können, müssen die

Koordinaten in ein kamerabezogenes Koordinatensystem transformiert werden. Abbildung 4.8 veranschaulicht die Grundidee der Modellierung. Ein PMD-Sensor wird zunächst als eine Matrix der Größe $M \times N$ definiert. Dabei ist M – die Anzahl der Zeilen eines PMD Sensors, und N – die Anzahl der Spalten.

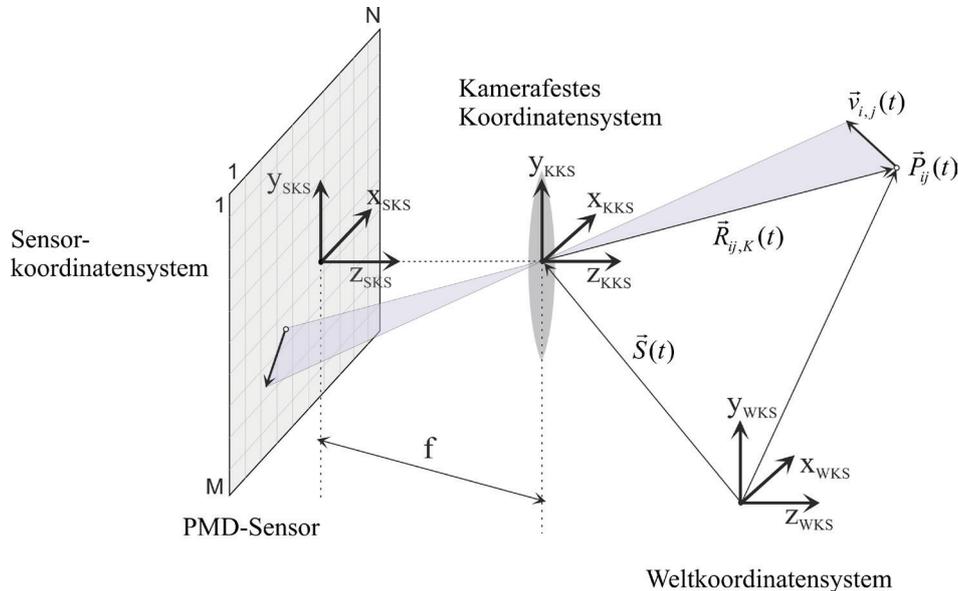


Abbildung 4.8: Abbildungsgeometrie zwischen dem kamerafestem Koordinatensystem und dem Weltkoordinatensystem

Es werden zunächst die zwei kamerabezogenen Koordinatensysteme betrachtet – das Sensorkoordinatensystem (SKS) und das kamerafeste Koordinatensystem (KKS). Als einziges starr mit der Kamera verbundenes Koordinatensystem bildet das KKS ein Kettenglied zwischen dem WKS und SKS, und im Falle einer optional definierbaren optischen Verzerrung auch mit dem Optikkordinatensystem (OKS). Aus Gründen der besseren Überschaubarkeit soll zunächst angenommen werden, dass das kamerafeste Koordinatensystem und das optische nicht gegeneinander verdreht sind. Als Kameramodell wird zunächst ein Lochkameramodell mit der entsprechenden optischen Abbildung (siehe Abschnitt 5.4) betrachtet.

Es wird angenommen, dass jedes Pixel (i, j) mit exakt einem Punkt $\vec{P}_{ij}(t)$ im Weltkoordinatensystem korrespondiert, dessen Richtung durch die entsprechende Pixelposition in der Pixelmatrix und die vorgegebene optische Abbildung und dessen Betrag dem gemessenen Entfernungswert des jeweiligen Pixels entsprechen. Der Ursprung des kamerafesten Koordinatensystems ist bezüglich des Ursprungs des Weltkoordinatensystems verschoben, die zeitabhängige Verschiebung (aufgrund möglicher Kamerabewegungen) sei $\vec{S}(t)$. Die Position des Punktzieles \vec{P}_{ij} mit $i \in [1, M]$, $j \in [1, N]$ ist gegeben durch:

$$\vec{P}_{ij}(t) = \vec{S}(t) + \vec{R}_{ij,K}(t) \quad (4.1)$$

mit $\vec{S}(t)$ – Position des Ursprungs des kamerafesten Koordinatensystems im Weltkoordinatensystem und $\vec{R}_{ij,K}(t)$ – Vektor zum Punktziel P_{ij} im kamerafesten Koordinatensystem.

Für den Abstand des Punktziels im KKS folgt somit:

$$|\vec{R}_{ij,K}(t)| = \sqrt{|\vec{S}(t)|^2 + |\vec{P}_{ij}(t)|^2 - 2 \cdot \vec{S}(t) \cdot \vec{P}_{ij}(t)} \quad (4.2)$$

Transformation zwischen Weltkoordinatensystem und kamerafestem Koordinatensystem ($WKS \leftrightarrow KKS$)

Für die Herleitung der Transformationsvorschrift zwischen WKS und KKS sollen zunächst die Abstandsvektoren in den beiden Koordinatensystemen betrachtet werden. Angenommen wird, dass das KKS gegenüber WKS verdreht ist. Auf die Indizes wird hierbei verzichtet, da es sich um denselben Punkt im Raum handelt. Die Position des Punktes sei im WKS mit

$$\vec{R} = \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix} = R_x \cdot \vec{e}_x + R_y \cdot \vec{e}_y + R_z \cdot \vec{e}_z \quad (4.3)$$

definiert, mit den Einheitsvektoren

$$\begin{aligned} \vec{e}_x &\perp \vec{e}_y \\ \vec{e}_x \times \vec{e}_y &= \vec{e}_z \end{aligned} \quad (4.4)$$

R_x, R_y, R_z sind die entsprechenden Projektionen auf die Einheitsvektoren.

$$\begin{aligned} \vec{R} \cdot \vec{e}_x &= R_x \cdot \vec{e}_x \cdot \vec{e}_x + R_y \cdot \vec{e}_y \cdot \vec{e}_x + R_z \cdot \vec{e}_z \cdot \vec{e}_x = R_x \\ \vec{R} \cdot \vec{e}_y &= R_x \cdot \vec{e}_x \cdot \vec{e}_y + R_y \cdot \vec{e}_y \cdot \vec{e}_y + R_z \cdot \vec{e}_z \cdot \vec{e}_y = R_y \\ \vec{R} \cdot \vec{e}_z &= R_x \cdot \vec{e}_x \cdot \vec{e}_z + R_y \cdot \vec{e}_y \cdot \vec{e}_z + R_z \cdot \vec{e}_z \cdot \vec{e}_z = R_z \end{aligned} \quad (4.5)$$

Die gleiche Position des Punktzieles wird im gedrehten Kamerakoordinatensystem durch

$$\vec{R}_K^R = \begin{bmatrix} R_{Kx} \\ R_{Ky} \\ R_{Kz} \end{bmatrix} = R_{Kx}^R \cdot \vec{e}_{Kx} + R_{Ky}^R \cdot \vec{e}_{Ky} + R_{Kz}^R \cdot \vec{e}_{Kz} \quad (4.6)$$

beschrieben, mit den entsprechenden Einheitsvektoren

$$\begin{aligned}\vec{e}_{Kx} &\perp \vec{e}_{Ky} \\ \vec{e}_{Kx} \times \vec{e}_{Ky} &= \vec{e}_{Kz}\end{aligned}\quad (4.7)$$

Nun werden die Gleichungen (4.3) und (4.6) kombiniert

$$R_{Kx}^R \cdot \vec{e}_{Kx} + R_{Ky}^R \cdot \vec{e}_{Ky} + R_{Kz}^R \cdot \vec{e}_{Kz} = R_x \cdot \vec{e}_x + R_y \cdot \vec{e}_y + R_z \cdot \vec{e}_z \quad (4.8)$$

und mit den einzelnen Einheitsvektoren des kamerafesten Koordinatensystems ausmultipliziert. Man erhält:

$$\begin{aligned}R_{Kx}^R \cdot \vec{e}_{Kx} \cdot \vec{e}_{Kx} + R_{Ky}^R \cdot \vec{e}_{Ky} \cdot \vec{e}_{Kx} + R_{Kz}^R \cdot \vec{e}_{Kz} \cdot \vec{e}_{Kx} &= R_x \cdot \vec{e}_x \cdot \vec{e}_{Kx} + R_y \cdot \vec{e}_y \cdot \vec{e}_{Kx} + R_z \cdot \vec{e}_z \cdot \vec{e}_{Kx} \\ R_{Kx}^R \cdot \vec{e}_{Kx} \cdot \vec{e}_{Ky} + R_{Ky}^R \cdot \vec{e}_{Ky} \cdot \vec{e}_{Ky} + R_{Kz}^R \cdot \vec{e}_{Kz} \cdot \vec{e}_{Ky} &= R_x \cdot \vec{e}_x \cdot \vec{e}_{Ky} + R_y \cdot \vec{e}_y \cdot \vec{e}_{Ky} + R_z \cdot \vec{e}_z \cdot \vec{e}_{Ky} \\ R_{Kx}^R \cdot \vec{e}_{Kx} \cdot \vec{e}_{Kz} + R_{Ky}^R \cdot \vec{e}_{Ky} \cdot \vec{e}_{Kz} + R_{Kz}^R \cdot \vec{e}_{Kz} \cdot \vec{e}_{Kz} &= R_x \cdot \vec{e}_x \cdot \vec{e}_{Kz} + R_y \cdot \vec{e}_y \cdot \vec{e}_{Kz} + R_z \cdot \vec{e}_z \cdot \vec{e}_{Kz}\end{aligned}\quad (4.9)$$

Mit Ausnutzung der Orthogonalität von (4.7) erhält man die einzelnen Komponenten des Vektors \vec{R} im Kamerakoordinatensystem

$$\begin{aligned}R_{Kx}^R &= R_x \cdot \vec{e}_x \cdot \vec{e}_{Kx} + R_y \cdot \vec{e}_y \cdot \vec{e}_{Kx} + R_z \cdot \vec{e}_z \cdot \vec{e}_{Kx} \\ R_{Ky}^R &= R_x \cdot \vec{e}_x \cdot \vec{e}_{Ky} + R_y \cdot \vec{e}_y \cdot \vec{e}_{Ky} + R_z \cdot \vec{e}_z \cdot \vec{e}_{Ky} \\ R_{Kz}^R &= R_x \cdot \vec{e}_x \cdot \vec{e}_{Kz} + R_y \cdot \vec{e}_y \cdot \vec{e}_{Kz} + R_z \cdot \vec{e}_z \cdot \vec{e}_{Kz}\end{aligned}\quad (4.10)$$

und daraus die Abbildungsvorschrift für die Transformation

$$\begin{bmatrix} R_{Kx}^R \\ R_{Ky}^R \\ R_{Kz}^R \end{bmatrix} = \begin{bmatrix} \vec{e}_x \cdot \vec{e}_{Kx} & \vec{e}_y \cdot \vec{e}_{Kx} & \vec{e}_z \cdot \vec{e}_{Kx} \\ \vec{e}_x \cdot \vec{e}_{Ky} & \vec{e}_y \cdot \vec{e}_{Ky} & \vec{e}_z \cdot \vec{e}_{Ky} \\ \vec{e}_x \cdot \vec{e}_{Kz} & \vec{e}_y \cdot \vec{e}_{Kz} & \vec{e}_z \cdot \vec{e}_{Kz} \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}\quad (4.11)$$

Formal stellt die Rotationstransformation aus dem Weltkoordinatensystem in das kamerafeste Koordinatensystem somit ein Produkt des Vektors \vec{R} mit einer Rotationstransformationsmatrix M_{WK} .

$$\vec{R}_K^R = \begin{bmatrix} \vec{e}_x \cdot \vec{e}_{Kx} & \vec{e}_y \cdot \vec{e}_{Kx} & \vec{e}_z \cdot \vec{e}_{Kx} \\ \vec{e}_x \cdot \vec{e}_{Ky} & \vec{e}_y \cdot \vec{e}_{Ky} & \vec{e}_z \cdot \vec{e}_{Ky} \\ \vec{e}_x \cdot \vec{e}_{Kz} & \vec{e}_y \cdot \vec{e}_{Kz} & \vec{e}_z \cdot \vec{e}_{Kz} \end{bmatrix} \cdot \vec{R} = M_{WK} \cdot \vec{R}$$

mit:

$$M_{WK} = \begin{bmatrix} \vec{e}_x \cdot \vec{e}_{Kx} & \vec{e}_y \cdot \vec{e}_{Kx} & \vec{e}_z \cdot \vec{e}_{Kx} \\ \vec{e}_x \cdot \vec{e}_{Ky} & \vec{e}_y \cdot \vec{e}_{Ky} & \vec{e}_z \cdot \vec{e}_{Ky} \\ \vec{e}_x \cdot \vec{e}_{Kz} & \vec{e}_y \cdot \vec{e}_{Kz} & \vec{e}_z \cdot \vec{e}_{Kz} \end{bmatrix} = \left[\vec{t}_1 \mid \vec{t}_2 \mid \vec{t}_3 \right]$$

\vec{t}_1 , \vec{t}_2 und \vec{t}_3 repräsentieren die entsprechenden neuen Einheitsvektoren, die nichts anderes sind, als die Einheitsvektoren \vec{e}_{Kx} , \vec{e}_{Ky} und \vec{e}_{Kz} des kamerafesten Koordinatensystems ausgedrückt im Weltkoordinatensystem.

$$\vec{t}_1 = \begin{bmatrix} \vec{e}_x \cdot \vec{e}_{Kx} \\ \vec{e}_x \cdot \vec{e}_{Ky} \\ \vec{e}_x \cdot \vec{e}_{Kz} \end{bmatrix}, \quad \vec{t}_2 = \begin{bmatrix} \vec{e}_y \cdot \vec{e}_{Kx} \\ \vec{e}_y \cdot \vec{e}_{Ky} \\ \vec{e}_y \cdot \vec{e}_{Kz} \end{bmatrix}, \quad \vec{t}_3 = \begin{bmatrix} \vec{e}_z \cdot \vec{e}_{Kx} \\ \vec{e}_z \cdot \vec{e}_{Ky} \\ \vec{e}_z \cdot \vec{e}_{Kz} \end{bmatrix} \quad (4.13)$$

Die Rotationstransformation kann folglich mit

$$\vec{R}_K^R = \left[\vec{t}_1 \mid \vec{t}_2 \mid \vec{t}_3 \right] \cdot \vec{R} = R_x \cdot \vec{t}_1 + R_y \cdot \vec{t}_2 + R_z \cdot \vec{t}_3 \quad (4.14)$$

beschrieben werden. Diese drückt den Vektor R_K^R im KKS mit den Termen R_x , R_y und R_z des WKS mit den neuen Einheitsvektoren aus.

Die Rotationstransformation eines beliebigen Vektors vom kamerafesten Koordinatensystem ins Weltkoordinatensystem kann mit Hilfe der inversen Transformationsmatrix von M_{WK} bestimmt werden.

$$\begin{aligned} \vec{R} &= M_{KW} \cdot \vec{R}_K^R = M_{WK}^{-1} \cdot \vec{R}_K^R \\ \text{mit} \\ M_{KW} = M_{WK}^{-1} &= \left[\vec{u}_1 \mid \vec{u}_2 \mid \vec{u}_3 \right] = \begin{bmatrix} \vec{e}_{Kx} \cdot \vec{e}_x & \vec{e}_{Ky} \cdot \vec{e}_x & \vec{e}_{Kz} \cdot \vec{e}_x \\ \vec{e}_{Kx} \cdot \vec{e}_y & \vec{e}_{Ky} \cdot \vec{e}_y & \vec{e}_{Kz} \cdot \vec{e}_y \\ \vec{e}_{Kx} \cdot \vec{e}_z & \vec{e}_{Ky} \cdot \vec{e}_z & \vec{e}_{Kz} \cdot \vec{e}_z \end{bmatrix} \end{aligned} \quad (4.15)$$

Die Spaltenvektoren \vec{u}_1 , \vec{u}_2 und \vec{u}_3 repräsentieren analog die Einheitsvektoren des WKS ausgedrückt im KKS.

Die oben hergeleiteten Transformationsvorschriften stellen im Grunde genommen eine Richtungskosinusmatrix dar. Sie sind mathematisch äquivalent zu den in der Literatur häufiger zu findenden Rotationstransformationen mit Euler-Winkeln. Je nachdem wie die Objektbewegung und somit die Bewegung jedes Punktziels im Weltkoordinatensystem definiert sind, kann die oben vorgestellten Transformationen weniger Operationen nach sich ziehen und somit schneller durchgeführt werden. Bei winkelbasierten Trajektoriendefinitionen bietet die Eulersche Form Vorteile.

Mit Beachtung der Rotationsvorschriften (4.12) und (4.15), können nun die allgemeinen Transformationsvorschriften zwischen dem WKS und KKS abgeleitet werden:

$$\begin{aligned} \vec{R}_K &= \vec{R}_K^R + \vec{T}_{WK} = M_{WK} \cdot \vec{R} + \vec{T}_{WK} \\ \vec{T}_{WK} &= \vec{O}_W - \vec{O}_K \end{aligned} \quad (4.16)$$

\vec{T}_{WK} – Verschiebungsvektor zwischen dem Ursprung \vec{O}_W des WKS und dem Ursprung \vec{O}_K des KKS.

Für die Rücktransformation gilt entsprechend:

$$\begin{aligned}\vec{R} &= \vec{R}_K^R + \vec{T}_{KW} = M_{KW} \cdot \vec{R}_K + \vec{T}_{KW} \\ \vec{T}_{KW} &= \vec{O}_K - \vec{O}_W\end{aligned}\quad (4.17)$$

Transformation zwischen kamerafestem Koordinatensystem und Trajektorienkoordinatensystem ($KKS \leftrightarrow TKS$)

Da die Ursprünge der beiden Koordinatensysteme zusammenfallen, sind die beiden Koordinatensysteme gegeneinander nur verdreht. Die Transformation KKS ins TKS lässt sich somit mit Hilfe einer Rotationsmatrix beschreiben:

$$\vec{R}_T = M_{KT} \cdot \vec{R}_K = [\underline{R}_x(\alpha) \cdot \underline{R}_y(\beta) \cdot \underline{R}_z(\gamma)] \cdot \vec{R}_K \quad (4.18)$$

mit den Rotationsmatrizen $\underline{R}_x(\alpha)$, $\underline{R}_y(\beta)$ und $\underline{R}_z(\gamma)$ um die Achsen x_{KKS} , y_{KKS} und z_{KKS} .

$$\begin{aligned}M_{KT} &= \underline{R}_x(\alpha) \cdot \underline{R}_y(\beta) \cdot \underline{R}_z(\gamma) = \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix} \cdot \begin{pmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} \cos \beta \cos \gamma & \cos \beta \sin \gamma & -\sin \beta \\ \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \cos \beta \\ \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{pmatrix}\end{aligned}\quad (4.19)$$

Für die Rücktransformation aus dem TKS ins KKS gilt:

$$\vec{R}_K = M_{KT}^{-1} \cdot \vec{R}_T = [\underline{R}_x(-\alpha) \cdot \underline{R}_y(-\beta) \cdot \underline{R}_z(-\gamma)] \cdot \vec{R}_T \quad (4.20)$$

Transformation zwischen kamerafestem Koordinatensystem und Optikkoordinatensystem ($KKS \leftrightarrow OKS$)

Analog zu dem Trajektorienkoordinatensystem fallen die Ursprünge des KKS und des OKS zusammen. Die Transformationsvorschriften entsprechen hier den Transformationsvorschriften von (4.18) und (4.20), jedoch mit den entsprechenden Rotationswinkeln zwischen den Achsen der beiden Koordinatensysteme.

Transformation zwischen kamerafestem Koordinatensystem und Sensorkoordinatensystem ($KKS \leftrightarrow SKS$)

Bei dieser Transformation sind zwei Fälle zu unterscheiden. Wird ein ideal ausgerichtetes Sensor angenommen (Sensorebene senkrecht zur optischen Achse, Sensormitte auf der optischen Achse), dann ist der Ursprung des Sensorkoordinatensystems gegenüber dem kamerafesten Koordinatensystem lediglich um die Brennweite f entlang der z-Achse verschoben. Die Transformationsvorschriften lauten in diesem Fall:

$$\vec{R}_{S,ideal} = \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} + \vec{R}_K \quad \text{bzw.} \quad \vec{R}_K = \vec{R}_{S,ideal} - \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} \quad (4.21)$$

Hat der Sensor eine Neigung zur optischen Achse, dann lässt sich die Transformation analog zu (4.12) wie folgt durchführen:

$$\vec{R}_S = \begin{bmatrix} \vec{e}_{Kx} \cdot \vec{e}_{Sx} & \vec{e}_{Ky} \cdot \vec{e}_{Sx} & \vec{e}_{Kz} \cdot \vec{e}_{Sx} \\ \vec{e}_{Kx} \cdot \vec{e}_{Sy} & \vec{e}_{Ky} \cdot \vec{e}_{Sy} & \vec{e}_{Kz} \cdot \vec{e}_{Sy} \\ \vec{e}_{Kx} \cdot \vec{e}_{Sz} & \vec{e}_{Ky} \cdot \vec{e}_{Sz} & \vec{e}_{Kz} \cdot \vec{e}_{Sz} \end{bmatrix} \cdot \vec{R}_K + \begin{bmatrix} 0 \\ 0 \\ f \end{bmatrix} + \begin{bmatrix} \Delta x_{mHP} \\ \Delta y_{mHP} \\ 0 \end{bmatrix} = M_{KS} \cdot \vec{R}_K + \begin{bmatrix} \Delta x_{mHP} \\ \Delta y_{mHP} \\ f \end{bmatrix} \quad (4.22)$$

mit

$$M_{KS} = \begin{bmatrix} \vec{e}_{Kx} \cdot \vec{e}_{Sx} & \vec{e}_{Ky} \cdot \vec{e}_{Sx} & \vec{e}_{Kz} \cdot \vec{e}_{Sx} \\ \vec{e}_{Kx} \cdot \vec{e}_{Sy} & \vec{e}_{Ky} \cdot \vec{e}_{Sy} & \vec{e}_{Kz} \cdot \vec{e}_{Sy} \\ \vec{e}_{Kx} \cdot \vec{e}_{Sz} & \vec{e}_{Ky} \cdot \vec{e}_{Sz} & \vec{e}_{Kz} \cdot \vec{e}_{Sz} \end{bmatrix} = [\vec{v}_1 | \vec{v}_2 | \vec{v}_3]$$

Δx_{mHP} , Δy_{mHP} sind die x- und y- Anteile der Verschiebung des Sensormittelpunktes bezüglich des Hauptpunktes (siehe auch Abschnitt 5.4). Die Rücktransformation ist gegeben durch:

$$\vec{R}_K = M_{SK} \cdot \left(\vec{R}_S - \begin{bmatrix} \Delta x_{mHP} \\ \Delta y_{mHP} \\ f \end{bmatrix} \right) \quad (4.23)$$

mit

$$M_{SK} = M_{KS}^{-1} = [\vec{w}_1 | \vec{w}_2 | \vec{w}_3] = \begin{bmatrix} \vec{e}_{Sx} \cdot \vec{e}_{Kx} & \vec{e}_{Sy} \cdot \vec{e}_{Kx} & \vec{e}_{Sz} \cdot \vec{e}_{Kx} \\ \vec{e}_{Sx} \cdot \vec{e}_{Ky} & \vec{e}_{Sy} \cdot \vec{e}_{Ky} & \vec{e}_{Sz} \cdot \vec{e}_{Ky} \\ \vec{e}_{Sx} \cdot \vec{e}_{Kz} & \vec{e}_{Sy} \cdot \vec{e}_{Kz} & \vec{e}_{Sz} \cdot \vec{e}_{Kz} \end{bmatrix}$$

Transformation zwischen dem LED-Koordinatensystem und dem Beleuchter-Koordinatensystem ($LKS \leftrightarrow BKS$)

Wie im Abschnitt 3.3.2 beschrieben, besitzen die einzelnen Lichtquellen im Beleuchter eine feste Position und Richtung, die während der Simulation nicht veränderbar ist. Die

Richtung ist durch die Hauptrichtung des Beleuchters – die Achse z_{BKS} – vorgegeben. Die Position der Lichtquelle im Beleuchter wird für jede Lichtquelle als bekannt vorausgesetzt. Die Transformationsvorschrift für die i -te Lichtquelle im Beleuchter stellt somit eine einfache Verschiebung dar:

$$\vec{R}_{L,i} = \vec{R}_B - \begin{bmatrix} x_{L,i} \\ y_{L,i} \\ 0 \end{bmatrix} \quad \text{bzw.} \quad \vec{R}_B = \vec{R}_{L,i} + \begin{bmatrix} x_{L,i} \\ y_{L,i} \\ 0 \end{bmatrix} \quad (4.24)$$

$\begin{bmatrix} x_{L,i} & y_{L,i} & 0 \end{bmatrix}^T$ – Position der Lichtquelle i im BKS

Transformation zwischen dem Weltkoordinatensystem und dem Beleuchter-Koordinatensystem ($WKS \leftrightarrow BKS$)

Die Beleuchter im Simulator besitzen die gleiche „Freiheit“ wie die Kamera – ihre relative Position und die Richtung bezüglich des WKS kann sich zeitlich ändern. Die Transformationsvorschrift entspricht somit formal der Transformationsvorschrift ($WKS \leftrightarrow KKS$) mit den entsprechend angepassten Einheitsvektoren.

5 Modellierung und Simulation einer PMD-Kamera

Das nachfolgende Kapitel beschäftigt sich mit den Fragestellungen der Modellierung und Simulation einer PMD-Kamera. Es werden die im Simulator umgesetzten Kriterien, Algorithmen und die gestellten Anforderungen an die Simulatorarchitektur sowie an die zu simulierenden Szenarien betrachtet. Zu Beginn wird die Definition der Objektumgebung vorgestellt, gefolgt von der Modellierung eines PMD-Sensors. Anschließend wird der geometrische Modellierungsansatz eines jeden Pixels vorgestellt. Darauf folgend wird das Konzept der optischen Abbildung betrachtet. Erst nach der systematischen Betrachtung der einzelnen Grundkomponenten wird das übergreifende Simulationskonzept einer kompletten PMD-Kamera mit ihren physikalischen Kenngrößen und Effekten, sowie der Modellierungsansatz der theoretischen Antwort des Sensors als Gesamtkonzept vorgestellt.

Der Simulationsvorgang kann grob in vier Schritte eingeteilt werden:

- Sichtbarkeitsanalyse – umfasst die Berechnung der Teile der Szene, die von den Komponenten der Kamera aus sichtbar sind.
- Abbildung der Szene auf den Sensor und Zuordnung der Szenenbereiche den korrespondierenden Pixeln
- Berechnung der physikalischen Effekte (Bewegungsunschärfe, optische Verzerrung usw.)
- Berechnung der theoretischen Antwort des Sensors

Obwohl die Sichtbarkeitsanalyse programmtechnisch vor allen anderen Schritten durchgeführt wird, erfordert das Simulationskonzept und die Modellierung des darin gewählten Ansatzes Kenntnisse über den Sensor- und den Kameraaufbau. Daher wird dieser Schritt erst nach der Modellierung des PMD-Sensors und der Kamera in Abschnitt 6 behandelt.

5.1 Objektumgebung und ihre Definition

Alle grafischen Objekte in der Objektumgebung können auf zwei Weisen definiert vorliegen, die auch in der Computergrafik üblich sind – die Punkteform und die Polygonform. Bei der Punkteform, auch Punktwolke genannt, wird die Oberfläche eines Objektes mit Hilfe der Punkte bzw. ihrer Koordinate auf dieser Oberfläche repräsentiert. Solche Umgebungsmodelle erhält man z.B. beim Abtasten der Umgebung mit einem Laserscanner. Die Koordinaten der Punktwolken aller Objekte in der Szene werden dabei objektweise sortiert in eine Matrix eingetragen:

$P_{M,i} \rightarrow [x_i \ y_i \ z_i \ D]$, mit x_i, y_i, z_i - die Koordinaten des Punktes i des Objektes D im Weltkoordinatensystem.

Der Vorteil einer solchen Darstellung ist die Einfachheit des Datenmanagements. Ein Punkt ist in sich eine geschlossene Einheit und es muss im Falle der Überlappung zweier Objekte nicht daraufhin untersucht werden, ob der Punkt noch weiteren Strukturen zugehören kann oder nicht. Doch genau diese Eigenschaft der einzelnen Punkte stellt die größte Herausforderung bei der Sichtbarkeitsberechnung dar. Jeder Punkt ist per Definition lediglich durch seine Koordinaten im Raum repräsentiert und besitzt keinerlei Nachbarschaftsbeziehungen zu anderen Punkten. Eine Punktwolke ist somit komplett durchsichtig, es können ohne weiteres keine Oberflächen gebildet werden. Abbildung 5.1 veranschaulicht das Sichtbarkeitsproblem der Punkteform-Definition. In einem Polygon-Modell dagegen bilden 3D-Punkte die Eckpunkte von geschlossenen Flächen. Diese zusätzliche Information bietet enorme Vorteile hinsichtlich der Berechnung von Sichtbarkeit einer Szene.

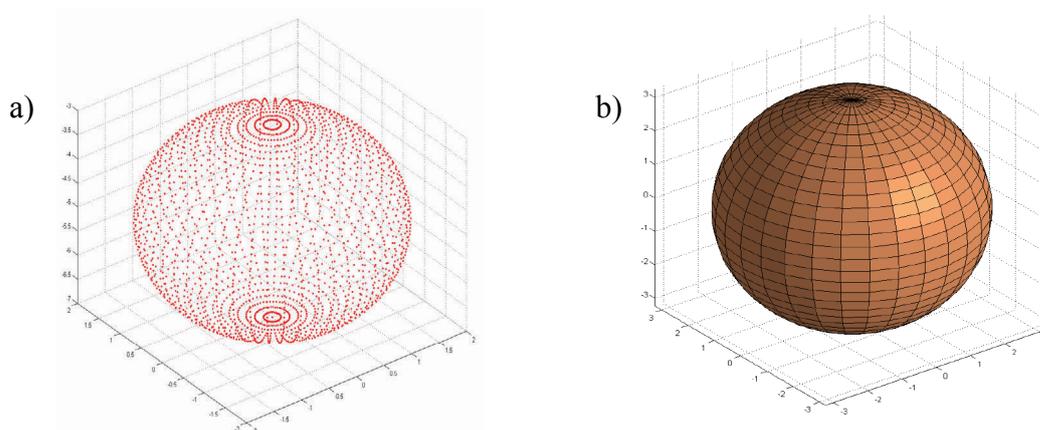


Abbildung 5.1: Darstellung einer Szene in Punktwolke-Form (links) und in Polygonform (rechts)

Das entwickelte Simulationsmodul erkennt automatisch anhand des vorliegenden Formates wie die Objekte definiert sind und wählt dementsprechend die Strategie der Sichtbarkeitsanalyse aus. Jedoch sind hinsichtlich der Definition der Polygone (Facetten) folgende Einschränkungen zu beachten:

- Methodenbedingt werden nur Modelle mit drei (Triangles) oder vier (Quads) Polygoneckpunkten, auch Knoten oder Vertices genannt, betrachtet. Objekte mit größerer Knotenzahl müssen zuerst auf die entsprechende Form umgerechnet werden.
- Unterstützt wird *polygonorientierte Datenstruktur*, also der Form

$$\text{Polygon} = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_n, y_n, z_n)\} \quad (5.1)$$

Dabei werden Kanten zwischen aufeinanderfolgenden Knoten erzeugt.

- Quads dürfen nur konvex sein, d.h. nach außen gewölbt. Ein Polygon ist dann konvex, wenn alle Linien, die zwei beliebige Knoten verbinden, vollständig innerhalb des Polygons liegen.
- Objekte dürfen nur aus planaren Quads bestehen (Triangles sind per Definition planar), d.h. alle Eckpunkte eines Polygons sollen auf einer Ebene im Raum liegen.
- Alle Polygone besitzen Umlaufsinn, welcher gleich orientiert ist, oder mit anderen Worten, die Position der Koordinaten der einzelnen Objekte (siehe Tabelle 5.1) in dem Objektdatensatz ist entscheidend für die Berechnung der Sichtbarkeit und somit für die theoretische Antwort des Sensors. Die nachfolgenden Berechnungen gehen von folgender Definition des Umlaufsinnes aus:

Das Kreuzprodukt der Vektoren zwischen den Knoten $P_1 \rightarrow P_2$ und $P_2 \rightarrow P_3$ zeigt in die gleiche Richtung wie der Normalvektor auf die damit aufgespannte Facettenebene: $\frac{\overrightarrow{P_1P_2} \times \overrightarrow{P_2P_3}}{|\overrightarrow{P_1P_2} \times \overrightarrow{P_2P_3}|} = \vec{n}_0 = \frac{\vec{n}}{|\vec{n}|}$. Der Normalvektor zeigt Richtung Außenseite einer Facette.

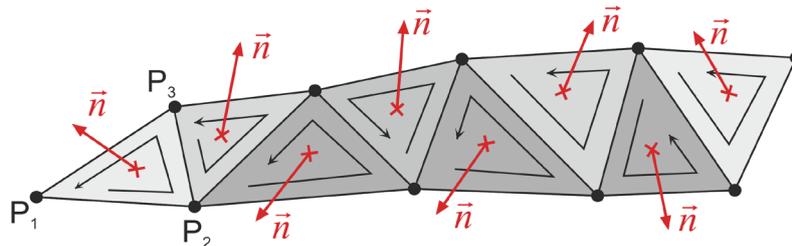


Abbildung 5.2: Veranschaulichung des Umlaufsinnes der Facetten

In der Tabelle 5.1 ist ein Ausschnitt aus einem Objektdatensatz dargestellt. Das Datenformat umfasst am Anfang die Koordinaten der einzelnen Knoten, die Position der Knoten in der Tabelle ist entscheidend für die Definition der Außenseite der Facette.

Laufende Nr.	Knoten 1	Knoten 2	Knoten 3	Knoten 4
1	$\begin{pmatrix} 80,3950 \\ 50,2117 \\ 127,1235 \end{pmatrix}$	$\begin{pmatrix} 80,3894 \\ 50,2097 \\ 127,1212 \end{pmatrix}$	$\begin{pmatrix} 80,3895 \\ 50,2093 \\ 127,1208 \end{pmatrix}$	$\begin{pmatrix} 80,4032 \\ 50,2099 \\ 127,1197 \end{pmatrix}$
2	$\begin{pmatrix} 80,3909 \\ 50,2183 \\ 127,1302 \end{pmatrix}$	$\begin{pmatrix} 80,3949 \\ 50,2154 \\ 127,1273 \end{pmatrix}$	$\begin{pmatrix} 80,4033 \\ 50,2097 \\ 127,1243 \end{pmatrix}$	$\begin{pmatrix} 80,4095 \\ 50,2173 \\ 127,1301 \end{pmatrix}$
3	$\begin{pmatrix} 80,0120 \\ 50,2257 \\ 127,1412 \end{pmatrix}$	$\begin{pmatrix} 80,0118 \\ 50,2418 \\ 127,1407 \end{pmatrix}$	$\begin{pmatrix} 80,0113 \\ 50,2333 \\ 127,1517 \end{pmatrix}$	NaN

Tabelle 5.1: Beispiel: Ausschnitt aus einem gemischten Objektdatensatz mit Triangles und Quads

Jedes Objekt der Simulationsszene kann außerdem über ein zusätzliches *object motion set* (Bewegungsdaten) verfügen. In diesen wird eine mögliche Änderung der Szene durch Bewegung der einzelnen Objekte angegeben. Somit können nicht nur die Kamera und die Beleuchter gesteuert bewegt werden, sondern auch die einzelnen Elemente der Simulationsszene. Ein *object motion set* umfasst immer das gesamte Objekt, bzw. wird die Änderung auf das gesamte Objekt angewendet. Da die Objekte ursprünglich immer im Weltkoordinatensystem definiert und beschrieben werden, beziehen sich die Positionsänderungen auf das WKS. Dabei ist zu beachten, dass die Aktualisierung der Positionswerte zu den diskreten Zeitpunkten während der Inkrementierung der Zwischenframe-Zeitachse durchgeführt wird und nicht zwischen den Frames (siehe Kapitel 5.8). Dadurch wird sichergestellt, dass das etwaige „Verwischen“ der aufgenommenen Szene (Bewegungsunschärfe) bei längeren Integrationszeiten nicht nur durch Kamerabewegung stattfinden kann, sondern auch zusätzlich durch schnelle Objektbewegungen. Der Datensatz des *object motion set* wird nach dem „Delta“-Prinzip aufgebaut, d.h. es beinhaltet Änderungen der Position bezüglich der letzten Position.

$$[Position_{neu}]_{t_{Sim,ZW}} = \left[[Position_{alt}]_{t_{Sim,ZW-1}} + \Delta l \right]$$

Der Vorteil dieses Ansatzes gegenüber einer direkten Speicherung aller neuen Koordinaten liegt darin, dass die umfassenden Objektdatensätze nicht bei jeder Zeitinkrementierung aufs Neue geladen werden müssen, sondern die neuen Koordinaten durch eine einfache Matrizenoperation quasi direkt zur Verfügung stehen und somit die Rechenzeit deutlich verkürzt werden kann. Der Nachteil ist, dass sich mit dem Ansatz nur Verschiebungen starrer Objekte beschreiben lassen. Für den eher seltenen Simulationsfall, dass ein Objekt während der Simulation seine Form (und somit automatisch die Position der einzelnen Punkte) verändern kann, kann das *object motion set* als „NaN“ (Not-a-Number) deklariert werden, dann wird der Simulator automatisch beim Inkrementieren der Zeit die einzelnen Objektdatensätze laden.

Im Rahmen der Optimierung des Ablaufprozesses und Minimierung von Datenträgerzugriffen, die wegen der enormen Datenmengen eine Simulation erheblich ausbremsen können, wurde ein Objektdateiformat entwickelt, welches alle für die aktuelle und die nachfolgende Berechnung benötigten Informationen in sich zusammenfasst und trägt. Die übrigen Variablen in der Objektumgebung sind meistens lokaler Natur und beanspruchen somit nicht kontinuierlich viel Speicher. Abbildung 5.3 veranschaulicht den Aufbau des Objektdateiformates. Eine Objektdatei wird in Objektkategorien unterteilt. Jede Objektkategorie repräsentiert ein Objekt in der Objektumgebung.

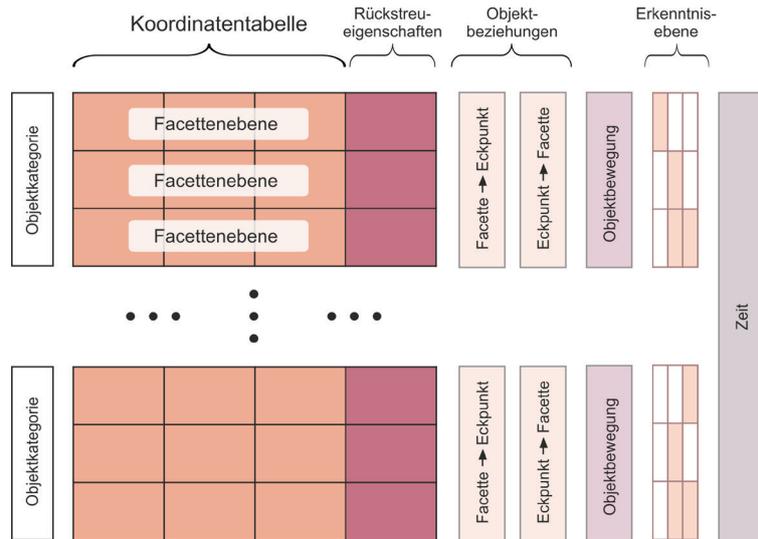


Abbildung 5.3: Aufbau des programminternen Objektdateiformats während der Initialisierung

In jeder Objektkategorie existieren bis zu fünf weitere sogenannte Objektebenen, die nachfolgend kurz erläutert werden:

- 1) *Die Koordinatenebene* beinhaltet die Koordinatentabellen der einzelnen Objekte.
- 2) *Die Rückstreuungseigenschaften-Ebene* beinhaltet die vorgegebenen und die berechneten Informationen über die Rückstreuungseigenschaften aller Objektpunkte.
- 3) *Die Objektbewegungsebene* beinhaltet Änderungen der Objektpositionen.
- 4) *Die Objektbeziehungsebene* erhält objektbezogene Informationen der Zugehörigkeitsbeziehungen Eckpunkte → Facette und Facette → Eckpunkte.

Aufgrund der, wie bereits oben erwähnt, genutzten polygonorientierten Datenstruktur werden einzelne Polygone aus Kanten zwischen den benachbarten Punkten in der Datenstruktur zusammengesetzt (vgl. (5.1)). Diese Datenstruktur sieht keine explizite Kennzeichnung der von benachbarten Polygonen geteilter Ecken und Kanten vor. Bei der Berechnung der Sichtbarkeit (vgl. Kapitel 6) kann dank solcher im Voraus berechneten Beziehungen die Effizienz der sehr vielen Abfragen signifikant gesteigert werden. Dabei wird die Dateistruktur um zusätzliche Informationen, ähnlich wie bei der Winged-Edge-Darstellung nach Baumgart [Baum 75], erweitert.

Zusätzlich zu den Informationen der Knoten in einer Facette werden weitere Zeiger auf die Facetten abgelegt, die diese Knoten ebenfalls einschließen. Mit Hilfe dieser Erweiterung der Datenstruktur ist es möglich, in konstanter Zeit abzufragen welcher Knoten zu welcher Facette gehört und umgekehrt. Ohne diese Informationen müsste der Datensatz immer wieder aufs Neue nach diesen Beziehungen abgesucht werden.

- 5) Die *Erkenntnis-Ebene* erhält die zur weiteren Berechnung erforderlichen Daten wie Markierung der sichtbaren Bereiche bezüglich Kamera- und Beleuchterposition, die zu jedem Pixel korrespondierende Szenenbereiche usw.

5.2 Modellierung des PMD-Sensors

Der PMD-Sensor wird idealisiert als eine rechteckförmige Pixelmatrix der Größe $N \times M$ angenommen. Definiert wird er durch seine Breite b_{PMD} , seine Höhe h_{PMD} , die Anzahl der Zeilen M und die Anzahl der Spalten N . Die Größe eines Pixels ergibt sich daraus automatisch. Wie schon in Abschnitt 4.8 erwähnt, wird der PMD-Sensor im Sensorkoordinatensystem definiert und beschrieben. Die von dem Sensor aufgespannte Ebene liegt in der (x_{SKS}, y_{SKS}) -Ebene, wobei die Achsen parallel zu den Zeilen bzw. Spalten des Sensors verlaufen. Der Ursprung liegt in der Mitte des Sensors. Entsprechend den oben genannten Parametern wird für den PMD-Sensor ein Knotengitter erstellt. Dieses beinhaltet die Eckpunkte aller Pixel und bildet das Grundgerüst des PMD-Sensors. D.h. jedes Pixel ist vorerst durch seine Eckpunkte innerhalb des Sensors definiert.

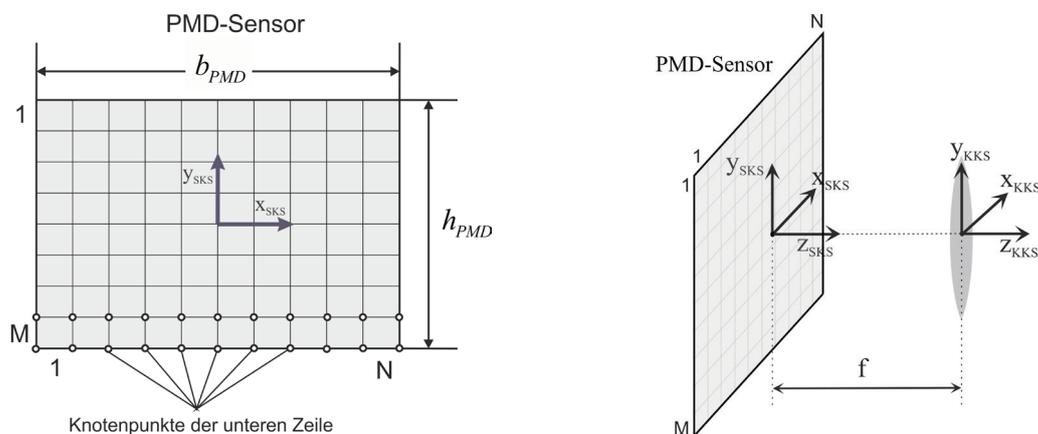


Abbildung 5.4: Modellierung des PMD-Sensors (links); Ausgangsdefinition des Sensors im Weltkoordinatensystem (rechts)

Der Sensor wird wie folgt modelliert:

- Der Ursprung des Sensorkoordinatensystems ist bezüglich des Ursprungs des kamerafesten Koordinatensystems um die Brennweite f versetzt (siehe auch Abschnitt 5.4). Die x- und y-Achsen der beiden Koordinatensysteme zeigen bei einem nicht geneigten Sensor in die gleiche Richtung, die z-Achsen stimmen überein.
- Das Knotengitter wird bezüglich der x_{SKS} und der y_{SKS} -Achse symmetrisch angeordnet.

- Die Sensorparameter wie Auflösung, Sensorgröße, Brennweite, Sensorneigung und Sensor-Shift (die letzten beiden werden im Abschnitt 5.5 näher erklärt) sind während der gesamten Simulation fest und können nicht verändert werden.
- Obwohl alle Knoten der PMD-Matrix in einer Ebene liegen und sie theoretisch auch in einem zweidimensionalen Raum \mathbb{R}^2 dargestellt werden könnten, werden, wie bereits in Abschnitt 4.4 beschrieben, die Koordinaten um die dritte z-Koordinate erweitert und im \mathbb{R}^3 betrachtet. Dieser Schritt erleichtert später die Transformationen zwischen dem SKS und KKS oder WKS.

5.3 Modellierung eines Pixels

Bei einem CMOS- oder CCD-Chip sind die lichtempfindlichen Sensoren meist in einer rechteckförmigen Anordnung auf dem Chip verteilt. Durch die Beschaltung (Leitungsbahnen) der einzelnen Pixel, sowie durch die Auslese- und Auswerteschaltung werden die eigentlichen lichtempfindlichen Bereiche eines Pixels, und nur über diese findet Signalaufnahme statt, reduziert. Diese Reduktion der Pixelfläche wird im Simulator auf folgende Weise modelliert.

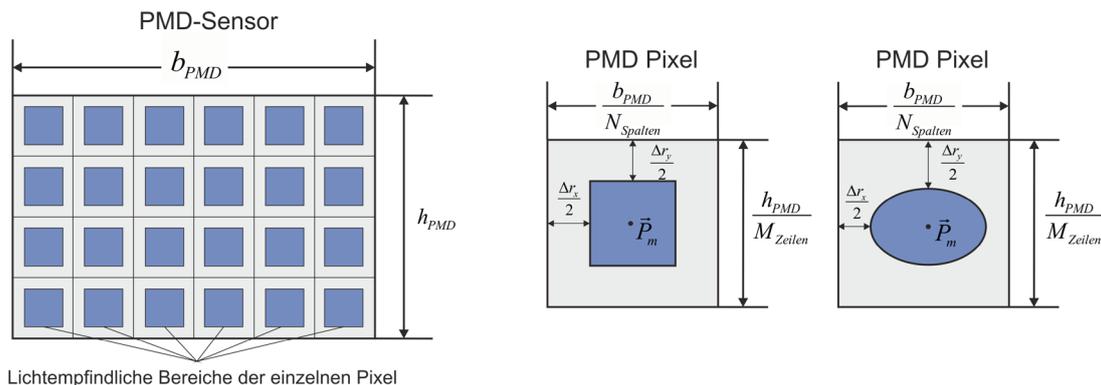


Abbildung 5.5: Modellierung eines PMD-Pixels und der lichtempfindlichen Pixelbereiche

Die gesamte lichtempfindliche Fläche des PMD-Chips wird entsprechend der Chip-Auflösung in gleich große Bereiche aufgeteilt (siehe Abbildung 5.5 links). Diese Bereiche stellen die eigentlichen Pixel dar. Die lichtempfindlichen Bereiche der Pixel, auch aktive Bereiche genannt, können wahlweise komplett das ganze Pixel einnehmen oder nur teilweise, wobei zwei miteinander kombinierbare oder einzeln anwendbare Einschränkungsverfahren zur Verfügung stehen:

- 1) *Die Einschränkung mittels eines Rechtecks:* Die rechteckig angenommene Pixelfläche wird seitlich um Δr_x in der horizontalen Richtung und um Δr_y in der vertikalen Richtung eingeschränkt. Dabei wird überprüft, ob die Abbildung des

Punktzieles auf dem PMD-Sensor $\vec{P}_{SKS} = (P_{x,SKS}, P_{y,SKS})$ folgenden Einschränkungen genügt:

$$\begin{aligned} P_{m,x} - \frac{b_{PMD}}{2N_{Spalten}} + \frac{\Delta r_x}{2} &\leq P_{x,SKS} \leq P_{m,x} + \frac{b_{PMD}}{2N_{Spalten}} - \frac{\Delta r_x}{2} \\ P_{m,y} - \frac{h_{PMD}}{2M_{Zeilen}} + \frac{\Delta r_y}{2} &\leq P_{y,SKS} \leq P_{m,y} + \frac{h_{PMD}}{2M_{Zeilen}} - \frac{\Delta r_y}{2} \end{aligned} \quad (5.2)$$

mit $\vec{P}_m = (P_{m,x}, P_{m,y})$ – der Mittelpunkt des Pixels, für welches die Überprüfung durchgeführt wird. Bei Erfüllung dieser beiden Abfragen liegt die Abbildung des Punktes im lichtempfindlichen Bereich des Pixels und kann somit potenziell detektiert werden.

- 2) *Die Einschränkung mittels einer Ellipse:* Hier wird analog vorgegangen, wobei die Fläche nach der Einschränkung ellipsenförmig ist. Die Abfrage lautet somit

$$\frac{(P_{x,SKS} - P_{m,x})^2}{a^2} + \frac{(P_{y,SKS} - P_{m,y})^2}{b^2} \leq 1$$

mit $a = \frac{1}{2} \left(\frac{h_{PMD}}{M_{Zeilen}} - \Delta r_y \right)$ und $b = \frac{1}{2} \left(\frac{b_{PMD}}{N_{Spalten}} - \Delta r_x \right)$ (5.3)

– die beiden Halbachsen der eingrenzenden Ellipse

Der Punkt liegt innerhalb der Ellipse, wenn die Bedingung erfüllt ist.

Bei Kombination dieser beiden Methoden können rechteckige Pixel mit abgerundeten Ecken oder Seiten simuliert werden. Abbildung 5.6 zeigt das Prinzip einer solchen mehrschichtigen Auswahlprozedur.

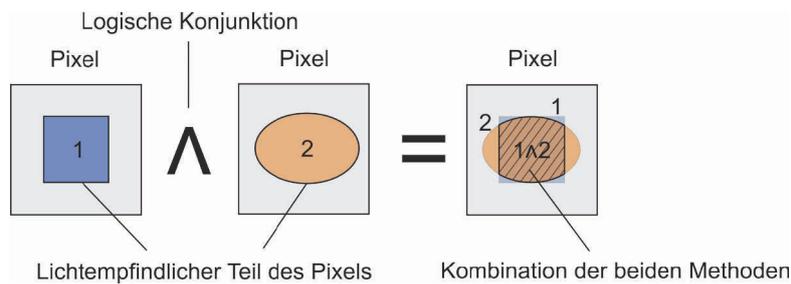


Abbildung 5.6: Veranschaulichung der Kombination der beiden Einschränkungsmöglichkeiten

5.4 Lochkameramodell

Das einfachste und deswegen auch in den Bereichen Simulation und Bildverarbeitung weit verbreitete Modell echter Kameras ist das in Abbildung 5.7 dargestellte *Lochkame-*

ramodell. In der Modellierung des PMD-Simulators wird dieses Kameramodell als Grundlage verwendet. Die Simulation optischer Verzerrungen baut auf dieser Grundlage auf und wird in Kapitel 7 ausführlicher behandelt.

Die Optik einer Lochkamera wird durch ein unendlich kleines Loch auf der optischen Achse nachgebildet. Mathematisch gesehen stellt dieses Loch einen Punkt dar, genannt *Brennpunkt* oder auch optisches Zentrum. Der Schnittpunkt der optischen Achse mit der Bildebene heißt *Hauptpunkt*. Der Abstand zwischen dem Hauptpunkt und dem Brennpunkt wird als *Brennweite* f der Kamera bezeichnet. Sie legt fest, wie groß das Objekt in der Kamera erscheint, bzw. bestimmt das Abbildungsverhältnis zwischen dem realen Objekt und seiner Abbildung. Kürzere Brennweiten bewirken einen weiteren Öffnungswinkel, längere Brennweiten einen engeren.

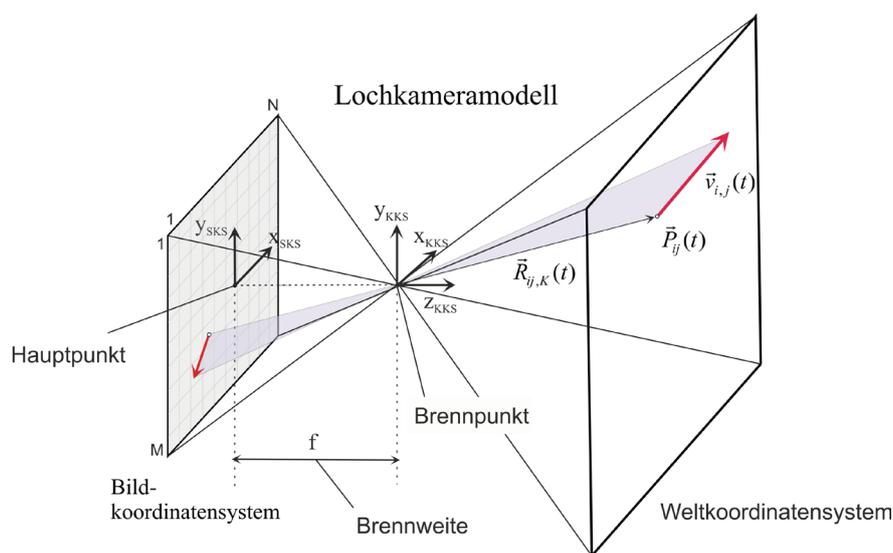


Abbildung 5.7: Das Lochkameramodell bestehend aus einer Bildebene und einem Brennpunkt

Wie erwähnt, liegen die Koordinaten der Objektpunkte üblicherweise im Weltkoordinatensystem vor. Durch eine affine Transformation können die Koordinaten in das kamerafeste Koordinatensystem überführt werden, wessen Ursprung im optischen Zentrum – dem Brennpunkt – liegt (siehe Abschnitt 4.2 sowie die Transformationsvorschrift (4.12)). Durch eine perspektivische Projektion durch den Brennpunkt werden die Objektpunkte auf die Bildpunkte transformiert, welche dann im Sensorkoordinatensystem vorliegen. Steht die optische Achse senkrecht auf der Sensorebene (ideale Sensorausrichtung), so lässt sich dank der einfachen Geometrie die Projektion eines Punktes gegeben im Kamerakoordinatensystem in den Sensorkoordinaten wie folgt beschreiben:

$$x_{SKS} = f \cdot \frac{x_{KKS}}{z_{KKS}} \quad y_{SKS} = f \cdot \frac{y_{KKS}}{z_{KKS}} \quad (5.4)$$

Die optische Achse sollte idealerweise senkrecht und mittig (telezentrisch) auf den Sensor auftreffen. Trotz aller Bemühungen kann es bei der Fertigung allerdings trotzdem passieren, dass der Sensor gegenüber seiner idealen Position etwas versetzt und/oder

verdreht ist, was eine Verschiebung der optischen Abbildung hervorrufen kann. Für den allgemeinen Fall, dass die optische Achse nicht senkrecht auf der Sensorebene steht, muss der Scherungseffekt bei der Abbildung berücksichtigt werden. Die Abbildung der Punktziele auf dem gekippten Sensor kann jedoch nicht durch eine einfache Drehtransformationmatrix aus der Abbildung der Punktziele auf dem idealen Sensor nachgebildet werden. Die Abbildung 5.8 macht es deutlich.

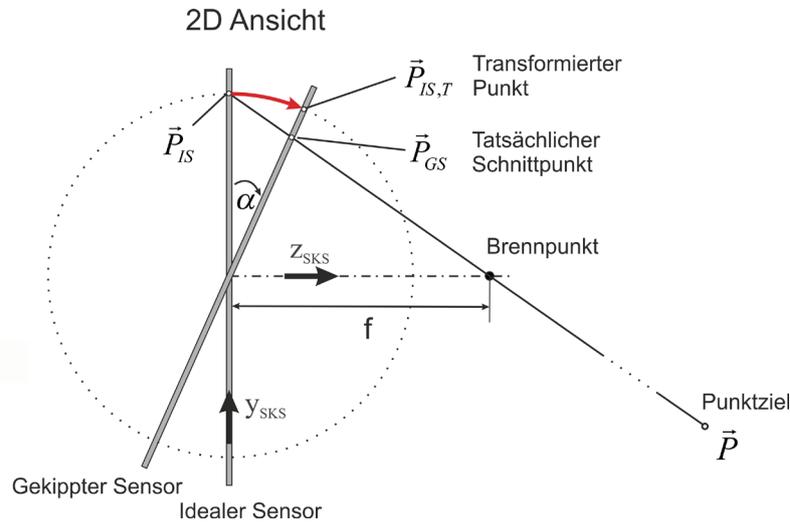


Abbildung 5.8: Das Lochkameramodell im Falle eines gekippten Sensors

Betrachtet wird ein Punktziel \vec{P} in der yz -Ebene im Sensorkoordinatensystem. Die perspektivische Projektion des Punktzies auf die Sensorebene sei \vec{P}_{IS} . Wird jetzt eine Drehung im Uhrzeigersinn um die x -Achse durchgeführt, dann wandert der Punkt bei einer solchen Transformation nach $\vec{P}_{IS,T}$. Dieser entspricht jedoch nicht dem tatsächlichen Schnittpunkt des Projektionsvektors mit der gekippten Sensorebene. Mit der Kenntnis des Drehwinkels α lässt sich der neue Schnittpunkt geometrisch bestimmen. Das Problem dabei ist, dass man es theoretisch mit bis zu drei Drehungen um alle drei Achsen zu tun hat. Dazu kommt ein eventueller Versatz der Mitte des Sensors wodurch der Punkt, um welchen die Drehungen gemacht werden, nicht mehr auf der optischen Achse liegt. Der Schnittpunkt auf dem gekippten Sensor lässt sich bei bekannten Drehwinkeln mathematisch herleiten. Auf die Herleitung wird an dieser Stelle verzichtet. Die dabei entstehenden Ausdrücke sind relativ kompliziert, so dass aufgrund der darin enthaltenen trigonometrischen Funktionen genauest auf die Ausrichtung der Anordnung geachtet werden muss. Einen wesentlich einfacheren Ansatz, welcher auch im Simulator implementiert wurde und im Abschnitt 5.6 ausführlicher behandelt wird, bietet die Betrachtung der perspektivischen Projektion \vec{P}_{GS} des Punktzies auf die Sensorebene im Sensorkoordinatensystem. Der Vorteil dieses Ansatzes liegt in seiner Einfachheit und Schnelligkeit. Er ist vektorbasiert und somit in jeder Situation eindeutig (keine Definitionsbereichsunterscheidung, wie z.B. bei Tangensfunktionen, erforderlich).

5.5 Geometrische Modellierung der PMD-Kamera

Bei der Modellierung einer PMD-Kamera werden die einzelnen Bestandteile der Kamera – der Sensor, die Optik, und der/die Beleuchter – zusammengeführt. In der Simulation besteht die Kamera also aus mindestens drei Komponenten. Die Anzahl der Beleuchter ist theoretisch nicht begrenzt, die Mindestanzahl ist jedoch einer. Das Kameramodell umfasst ein geometrisches Modell und ein Signalsimulationsmodell. Das Signalsimulationsmodell wird etwas später vorgestellt, hier soll zunächst das rein geometrische Modellierungskonzept betrachtet werden. Da die einzelnen Kamerakomponenten bereits ausführlich behandelt wurden, wird hier lediglich eine Zusammenstellung der wichtigsten Kriterien und Parameter gegeben. Abbildung 5.9 zeigt das bereits bekannte und nun erweiterte vereinfachte geometrische Modell.

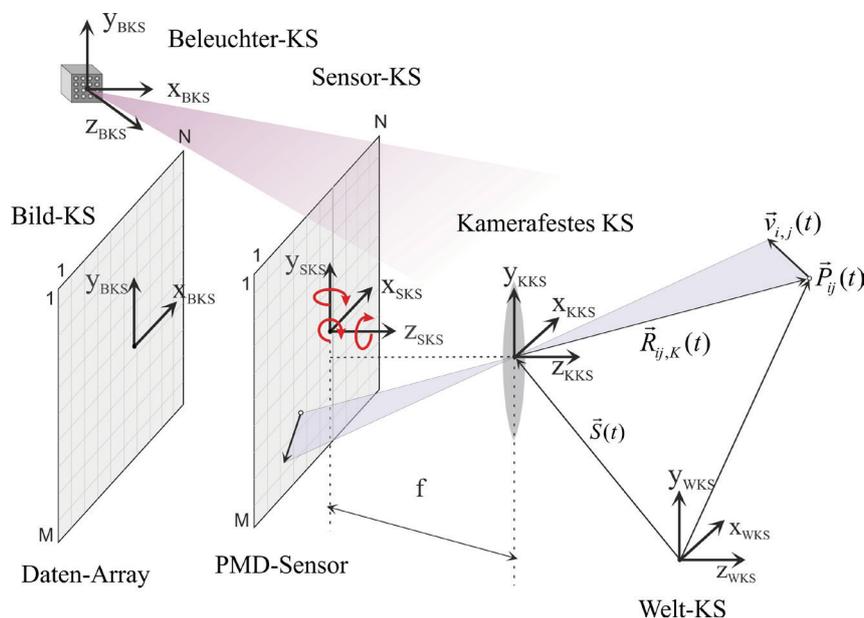


Abbildung 5.9: Das vereinfachte (ohne Zwischenkoordinatensysteme) geometrische Modell einer PMD-Kamera

Die geometrische Modellierung der PMD-Kamera wird im Simulator wie folgt umgesetzt. Der Sensor wird, wie bereits erwähnt, durch eine Pixelmatrix mit rechteckigen Pixeln simuliert. Alle Pixel sind per Definition gleich groß, der lichtempfindliche Bereich des Pixels kann allerdings entsprechend (5.2) und/oder (5.3) eingeschränkt werden. Der Sensor kann bezüglich der optischen Achse der Kamera verdreht und/oder verschoben werden. Die Neigung des Sensors wird mit dem normalen Vektor zur Sensorebene vorgegeben, welcher definitionsgemäß dem Vektor \vec{z}_{SKS} entspricht. Als Verschiebung (Sensor-Shift) kann eine beliebige vektorielle Verschiebung in der (x_{SKS}, y_{SKS}) -Ebene sein. Der Öffnungswinkel der Kamera wird durch die Größe des PMD-Sensors und durch die Brennweite vorgegeben. Diese bilden zusammen eine Pyramide, welche in den Raum abgebildet wieder eine Pyramide bildet. Der Sichtbereich der Kamera ist somit immer pyramidenförmig.

Der/die Beleuchter werden im Simulator entsprechend den Vorgaben der einzelnen Lichtquellen zusammengesetzt. Sie können in der Simulation entweder fest an die Kamera gebunden werden (dann bewegen sie sich synchron mit der Kamera), oder sie können unabhängig von der Kamera aufgestellt und bewegt werden. Die Bewegung der einzelnen Beleuchter und der Kamera wird in den Trajektorien-dateien vorgegeben.

Wird bei der Simulation kein Beleuchter vorgegeben, dann wird automatisch eine Punktlichtquelle im Ursprung des Sensorkoordinatensystems angenommen. Die Simulation entspricht dann einer *monostatischen Sensor-Beleuchter Konstellation*. An dieser Stelle muss allerdings beachtet werden, dass durch die Sensorausmaße von bis zu einigen Millimetern dennoch ein bistatischer Effekt zu verzeichnen sein wird. Dieser fällt jedoch bei solch kleinen bistatischen Abständen kaum ins Gewicht und ist in der Regel deutlich kleiner als der Quantisierungsfehler infolge der digitalen Verarbeitung (siehe Abschnitt 5.9).

Zusätzlich zu der geometrischen Realisierung des Sensors – dem Knotengerüst – wird ein mehrschichtiges Datenarray der gleichen Auflösung wie der Sensor angelegt. In diesem werden die berechneten Bildinformationen, die Entfernungs- und die Helligkeitswerte, abgelegt.

Jede Simulation startet mit einem Initialisierungsvorgang. Während dieser kurzen Prozedur werden die wichtigsten Daten und Parameter vorberechnet. Auf diese Weise ist es möglich, die eigentliche Simulation von der Aufbereitung der Daten zu trennen. Es ist vergleichbar mit einem Schachspiel, bei dem die Figuren am Anfang des Spiels auf dem Schachbrett aufgestellt werden. Wenn man aber eine bestimmte Schachpartie durchspielen möchte, dann sollen die Figuren zuerst an die entsprechenden Plätze versetzt werden. So wird auch am Anfang der Initialisierung zunächst eine Standardaufstellung vorgeladen, die dann entsprechend den Vorgaben für die Simulation des ersten Frames vorbereitet wird – die Koordinatensysteme werden neu aufgestellt, die Szene an ihren Platz verschoben, der Sensor bezüglich der Kamera neu ausgerichtet und anschließend zusammen mit der Kamera neupositioniert und ausgerichtet. Das gleiche gilt für die Beleuchter. Am Ende des Initialisierungsvorgangs „stehen“ alle Komponenten an ihren Plätzen, die Datenformate und Daten-Arrays sind vorbereitet, die Simulation kann gestartet werden.

5.6 Berechnung des zum Punktziel korrespondierenden Pixels

Wie in Kapitel 5.4 behandelt, werden die Punkte der Zielobjekte infolge einer optischen Abbildung auf die Sensorfläche projiziert. Für die Berechnung der theoretischen Antwort jedes Pixels müssen die Teile der Szene, die von diesem Pixel erfasst werden, berechnet werden. Die Tiefeninformationen der Punktziele sind später maßgeblich für die berechneten Tiefeninformationen des Sensors. Die Aufgabenstellung trägt also eigent-

lich einen umgekehrten Charakter – für jedes Pixel soll die entsprechende Szene berechnet werden. Die Realisierung dieser Berechnung wird aber durch die Zuordnung der Projektionen der Punktziele zu den entsprechenden Pixeln durchgeführt. Dies zieht einen deutlich geringeren rechnerischen Aufwand nach sich, als wenn die komplette Szene pixelweise auf die Detektierbarkeit überprüft würde.

Die Umkehrung der Problemstellung ist aus optischer Sicht absolut legitim, weil optische Abbildung bei einem Linsensystem in beide Richtungen umkehrbar ist. Ein einfallender Strahl auf einer Seite wird vom Linsensystem gebrochen und tritt auf der anderen Seite aus. Schickt man einen Lichtstrahl in umgekehrter Richtung, so tritt dieser an der exakt gleichen Stelle in die exakt gleiche Richtung aus wie der einfallende Strahl.

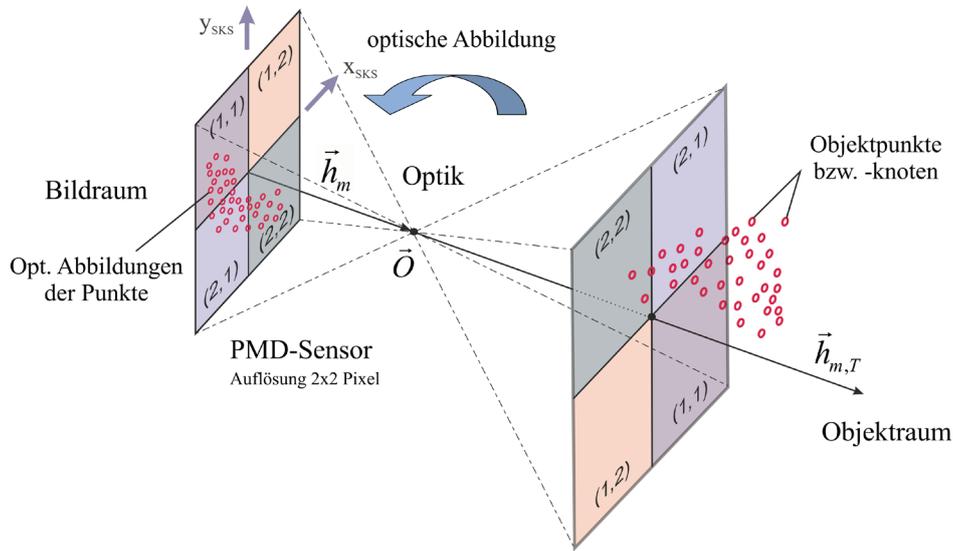


Abbildung 5.10: Illustration der pixelweisen Zuordnung der Abbildungen der Punktziele zu den entsprechenden Pixel

Die Realisierung der pixelweisen Szenensortierung wird im Simulator wie folgt umgesetzt: Gegeben ist eine Punktwolke im Objektraum mit Koordinaten aller Punkte, die von mindestens einer Lichtquelle beleuchtet und von der Kamera potenziell detektiert werden können (siehe Abbildung 5.10):

$$P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3), \dots, P_N(x_N, y_N, z_N) \quad (5.5)$$

Jeder Punkt der Punktwolke wird entsprechend der optischen Abbildung auf den Sensor abgebildet

$$P_1^{proj}(x_{1,SKS}, y_{1,SKS}), P_2^{proj}(x_{2,SKS}, y_{2,SKS}), \dots, P_N^{proj}(x_{N,SKS}, y_{N,SKS}) \quad (5.6)$$

Die Projektionen der Punkte werden einer bis zu dreistufigen Analyse unterzogen. Dabei wird zuerst bestimmt, von welchem Pixel das Punktziel detektiert werden kann und anschließend nach (5.2) und (5.3) überprüft, ob die Projektion des Punktes in dem lichtempfindlichen Bereich des Pixels liegt, falls welche definiert wurden.

Die einfachste Methode ist die Betrachtung der Koordinaten der Abbildungen der Punkte im Sensorkoordinatensystem. Das ist im Allgemeinen das einzige Koordinatensystem, in dem die Pixel bezüglich des Betrachters (z.B. eines anderen Koordinatensystems) nicht verzerrt sind und stellt auch den Hauptgrund der Einführung dieses Koordinatensystems dar. Im Abschnitt 12.1 wird ein alternativer Lösungsansatz bei Verzicht auf das zusätzliche Koordinatensystem vorgestellt.

Werden die Projektionen im SKS betrachtet, so kann das entsprechende Pixel aufgrund der gegebenen Sensorsymmetrie direkt bestimmt werden. Mit den Sensor- und Pixelgrößen aus 5.3 folgt für die Projektion des k -ten Punktes $P_k^{proj}(x_{k,SKS}, y_{k,SKS})$

$$\begin{aligned} i(Pixel) &= \text{round} \nearrow \left(\frac{x_{k,SKS} + b_{PMD}/2}{b_{PMD}/N_{\text{Spalten}}} \right) \in \mathbb{N} = \{1, 2, 3, \dots\} \\ j(Pixel) &= \text{round} \nearrow \left(\frac{y_{k,SKS} + h_{PMD}/2}{h_{PMD}/M_{\text{Zeilen}}} \right) \in \mathbb{N} = \{1, 2, 3, \dots\} \end{aligned} \quad (5.7)$$

mit $i(Pixel)$ und $j(Pixel)$ – die Indices des Pixels, von dem der Punkt $P_k(x_k, y_k, z_k)$ detektiert wird. $\text{round} \nearrow$ entspricht dem Operator *Aufrunden*.

Nachdem das zu dem Punkt $P_k(x_k, y_k, z_k)$ korrespondierende Pixel bestimmt wurde, werden seine Koordinaten in einen Pufferspeicher kopiert. Führt man die Berechnung für alle Punktziele fort, dann erhält man die zu jedem Pixel korrespondierenden Punkte im Objektraum, z.B.:

$$Pixel (1,1) \rightarrow \begin{cases} P_1(x_1, y_1, z_1) \\ P_7(x_7, y_7, z_7) \\ P_{10}(x_{10}, y_{10}, z_{10}) \\ \dots \end{cases}, \quad Pixel (1,2) \rightarrow \begin{cases} P_2(x_2, y_2, z_2) \\ P_4(x_4, y_4, z_4) \\ P_{15}(x_{15}, y_{15}, z_{15}) \\ \dots \end{cases} \quad usw. \quad (5.8)$$

5.7 Berechnung der theoretischen Antwort des Sensors

Mit den schrittweise und pixelweise gewonnenen Informationen darüber, welche Teile der Szene von einem Pixel detektiert werden können, kann die theoretische Antwort eines Pixels berechnet werden. Zunächst soll jedoch der einfache Fall mit einer Lichtquelle und einem Punktziel betrachtet werden.

Sendet eine Lichtquelle eine sinus- $E = A \cdot \sin(\omega t + \varphi_0)$ oder kosinusförmige Schwingung $E = A \cdot \cos(\omega t + \varphi_0)$ aus, dann kann im Punktziel P in einem Abstand l von der Lichtquelle die Schwingung

$$E_{hin} = A_0 \cdot \sin\left(2\pi \cdot \left(\frac{t}{T} - \frac{l}{\lambda}\right) + \varphi_0\right) \quad \text{oder} \quad E_{hin} = A_0 \cdot \cos\left(2\pi \cdot \left(\frac{t}{T} - \frac{l}{\lambda}\right) + \varphi_0\right) \quad (5.9)$$

gemessen werden, mit φ_0 – die Anfangsphase – die Schwingungsphase der Welle zum Zeitpunkt $t = 0$ am Ort $l = 0$; A_0 – die durch die Ausbreitung gedämpfte Amplitude der Schwingung; λ – die Wellenlänge.

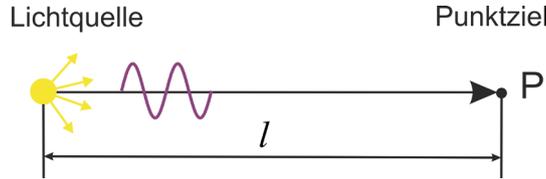


Abbildung 5.11: Darstellung der Wellenausbreitung

Zur rechnerischen Vereinfachung von Überlagerungen solcher Wellen, bietet die komplexe Form wesentliche Vorteile. Die Gleichung (5.9) wird zu:

$$E_{hin} = A_0 \cdot \text{Im}\left(e^{j\left[2\pi \cdot \left(\frac{t}{T} - \frac{l}{\lambda}\right) + \varphi_0\right]}\right) \quad \text{oder} \quad E_{hin} = A_0 \cdot \text{Re}\left(e^{j\left[2\pi \cdot \left(\frac{t}{T} - \frac{l}{\lambda}\right) + \varphi_0\right]}\right) \quad (5.10)$$

Beide lassen sich zusammenführen, wobei dann bei der Rechnung, je nach gewähltem Ausgangssignal, entsprechend nur der Realteil oder der Imaginärteil physikalische Bedeutung hat.

$$E_{hin} = A_0 \cdot e^{j\left[\frac{2\pi}{\lambda}(\lambda t - l) + \varphi_0\right]} = A_0 \cdot e^{j\varphi_0} \cdot e^{j\frac{2\pi}{\lambda}[ct - l]} \quad (5.11)$$

$$c = \frac{f}{\lambda} \quad - \quad \text{die Ausbreitungsgeschwindigkeit der Schwingungsphasen}$$

(5.11) beschreibt den zeitlichen Verlauf der Schwingung im Punktziel P. Wird der Hin- und Rückweg der Welle betrachtet, dann muss der Phasenterm entsprechend der einzelnen Entfernungen angepasst werden.

Nimmt man die Anfangsphase gleich Null an, so lässt sich im Falle mehrerer Punktziele die Überlagerung der zurückgestrahlten Wellen im Positionspunkt des PMD-Sensors (siehe Abbildung 5.12) analog zu (5.11) durch

$$\bar{r}_l = \sum_{n=1}^N B_n^* \cdot e^{j\varphi_n} = \sum_{n=1}^N B_n^* \cdot e^{j\frac{2\pi}{\lambda}(r_{l,P_n} + r_{P_n})} \quad (5.12)$$

angeben. B_n^* stellt dabei die Amplitude der aufgenommenen Schwingung dar, die infolge der Signalausbreitung und -dämpfung, wie in Kapitel 3 behandelt, eine andere ist, als die Amplitude des gesendeten Signals. r_{l,P_n} und r_{P_n} sind die Abstände zwischen der Lichtquelle und dem Punktziel P_n (Hinweg der Welle bis zur Reflektion) und dem

Punktziel und dem PMD-Pixel (Rückweg der Welle nach der Reflexion). $\lambda = \frac{c}{f_{\text{mod}}}$ ist die Wellenlänge der modulierten Schwingung, f_{mod} – Modulationsfrequenz. N – Anzahl der Punktziele.

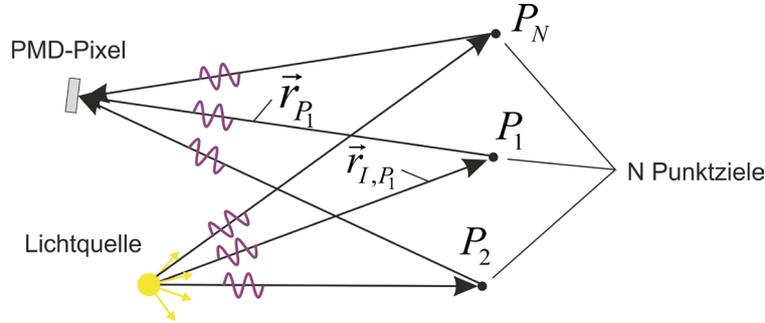


Abbildung 5.12: Überlagerung von Wellen auf dem PMD-Pixel

Bei mehr als nur einer Lichtquelle findet nicht nur Überlagerung reflektierter Wellen von verschiedenen Punktzielen, sondern auch zusätzlich Überlagerung unterschiedlicher Reflektionen von unterschiedlichen Lichtquellen statt. Mit den Erkenntnissen des Abschnitts 3.2.3 wird der Ausdruck (5.12) erweitert zu:

$$\bar{r} = \sum_{m=1}^L \sum_{n=1}^N \chi_{\text{Sim}} \cdot \rho_R(P_n) \cdot \frac{I_{I_m P_n} \cdot I_{\text{refl,korr}}}{(r_{I_m P_n} + r_{P_n})^2} \cdot e^{-\alpha(r_{I_m P_n} + r_{P_n})} e^{j \cdot 2\pi \frac{r_{I_m P_n} + r_{P_n}}{\lambda}} \quad (5.13)$$

L – Anzahl der Lichtquellen

$I_{I_m P_n}$ – die Abstrahldiagramm basierte Lichtintensität in Richtung des Punktzieles

$I_{\text{refl,korr}}$ – der winkelabhängige Korrekturfaktor des zurückgestrahlten Lichtes, welcher sich aus der relativen Lage des Pixels bezüglich des Punktzieles ergibt (photometrisches Grundgesetz). Bei Objekten in Punkteform ist dieser gleich eins.

α – Absorptionskoeffizient des durchquerten Mediums

$\rho_R(P_n) \in [0,1]$ – der relative Reflektionsgrad des Punktzieles bezogen auf die maximale Reflektion von 100%. Bei vollständiger Reflektion ist der relative Reflektionsgrad gleich eins, bei vollständiger Absorption – gleich Null. Im Simulator wird der Reflektionsgrad üblicherweise objektweise vorgegeben.

χ_{Sim} – Simulationskoeffizient (Normierungskoeffizient). Der Prozess der Ladungsträgergenerierung in den Photogates, das Auslesen der Ladungsträger mittels Ausleseschaltungen und die anschließende Verarbeitung dieser Messgrößen in Helligkeitswerte ist ein sehr komplizierter Prozess, welcher, wenn überhaupt, nur für eine bestimmte Kombination aus Sensor, seiner Beschaltung, optischen Eigenschaften des Objektivs usw. realisiert werden kann. Bei der Modellierung eines weitgehend universellen PMD-

Sensors stand dieses Ziel nicht im Vordergrund. Aufgrund der bekannten Lichtintensitätsverteilung an den Objekten und bei Voraussetzung bekannter Rückstreuungseigenschaften dieser Objekte ist es möglich, die reflektierte Lichtmenge, die auf einen PMD-Sensor auftrifft, zu bestimmen. Die von einem PMD-Pixel empfangene Lichtmenge entspricht zwar qualitativ dem Helligkeitswert dieses PMD-Pixels, sie spiegelt aber nicht exakt den tatsächlichen Wert wieder, der infolge der internen Verarbeitung des detektierten Lichtes von der Kamera als Messwert ausgegeben wird. Für eine bestimmte Optik (Lichtstärke, Brennweite und weitere wichtige Kennwerte eines Objektivs) und Kamera lässt sich eine Normierung der simulierten Helligkeitswerte durchführen. Erst im normierten Zustand lassen sich die Werte miteinander vergleichen.

Aus der berechneten Phasenverschiebung $\varphi_d = \arg(\bar{r})$ kann unter Annahme einer kontinuierlichen sinusförmigen Modulation (cw-Modulation) anschließend mit (2.3) der gesuchte Abstand berechnet werden.

$$R_{Sim} = \frac{c \cdot \varphi_d}{4\pi f_{mod}} \quad (5.14)$$

Der Betrag des Vektors \bar{r} repräsentiert den simulierten Helligkeitswert (die Amplitude) des PMD-Pixels.

$$A_{Sim} = |\bar{r}| \quad (5.15)$$

Die Antwort eines Pixels ist demnach ein Vektor, aus dessen Betrag die scheinbare Helligkeit und aus dessen Phase über die Signallaufzeit die Entfernung R berechnet wird. Jedes Pixel des PMD-Sensors repräsentiert somit nur einen Punkt im Raum mit der Helligkeit A_{Sim} in der Entfernung R_{Sim} .

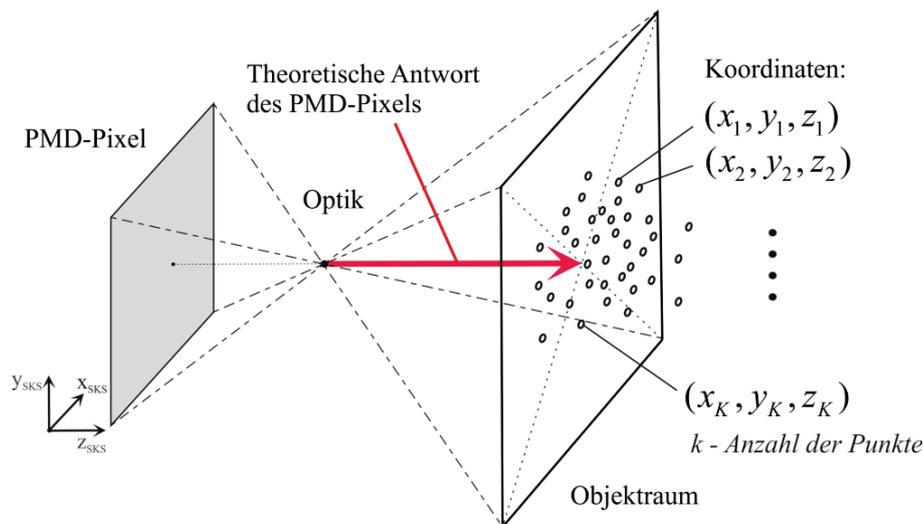


Abbildung 5.13: Illustration der theoretischen Antwort eines PMD-Pixels

Die Position dieses „Objekt“punktes im 3D-Raum ergibt sich dann aus dem Abstand R_{Sim} und der Richtung des Vektors, welche sich aus der optischen Abbildung der Pixelmitte in den Raum ergibt.

Dieser Richtungsvektor entspricht dem Vektor vom Ursprung des Optikkoordinatensystems zur Mitte des Öffnungswinkels der Pixelfacetten. Die gemessenen sowie die simulierten Entfernungswerte liegen somit grundsätzlich im kamerafesten Koordinatensystem vor.

Die gesamte Antwort des Sensors setzt sich aus den Antworten der einzelnen Pixel der Pixelmatrix zusammen. Man erhält auf diese Weise zwei Matrizen mit Amplituden- und Entfernungswerten.

5.7.1 Berücksichtigung des 4-Phasenalgorithmus

Wie in Abschnitt 2.2 behandelt, wird bei der PMD-Kamera die Korrelationsfunktion zur Bestimmung der Phasenverschiebung an zwei Messpunkten mit einem Intervall von $\Delta\tau = T_0/4$ abgetastet. Aus beiden Kanälen entstehen dann vier Messungen der Korrelationsfunktion. Nach (2.11) folgt für ein sinusförmiges Sendesignal für die gemessene Korrelationsfunktion:

$$k(\tau) = A_0^2 + \frac{A^2}{2} \cos(\omega_0\tau - \varphi_d) \quad (5.16)$$

Wird die störungsfreie Korrelationsfunktion an vier Stellen abgetastet, z.B. bei $\tau = 0, T_0/4, T_0/2, 3T_0/4$, dann entspricht die aus den vier Abtastwerten bestimmte Phasenverschiebung und die daraus gebildete Entfernung exakt der Entfernung (welche im Simulator stets bekannt ist) zum Punktziel, für welches die Verhältnisse gebildet werden. Anders sieht es aus, wenn die einzelnen Korrelationsfunktionen verrauscht sind. In der Realität unterliegen die einzelnen Abtastwerte der Korrelationsfunktion einer Zufallsstreuung. Bei Simulation von rauschbehafteten Entfernungswerten macht es daher Sinn, nicht abschließend die bereits ermittelten Entfernungsmesswerte zu verrauschen, sondern die Abtastwerte der störungsfreien Korrelationsfunktion. Somit wird letztendlich eine rauschbehaftete Korrelationsfunktion nachgebildet (siehe Abbildung 2.4, realer Verlauf). Aus den verrauschten Korrelationswerten kann nach (2.19) die verrauschte Phasenverschiebung

$$\varphi_{d,Rausch} = \arctan \left(\frac{k_{2,Rausch} - k_{4,Rausch}}{k_{1,Rausch} - k_{3,Rausch}} \right) \quad (5.17)$$

bestimmt werden. Für den verrauschten Abstand folgt somit mit (2.23)

$$R_{Rausch} = \frac{c_0}{2 \cdot f_{\text{mod}}} \cdot \left(\frac{\varphi_{d,Rausch}}{2\pi} + N \cdot 2\pi \right), \text{ mit } N = 0, 1, 2, 3, \dots \quad (5.18)$$

Beispiel: Einfluss der Messunsicherheit der detektierten Korrelationsfunktion auf die Messunsicherheit des Abstandes.

In Abbildung 5.14 ist der Einfluss des Winkelrauschens der Korrelationsfunktion auf die daraus berechnete Entfernung dargestellt.

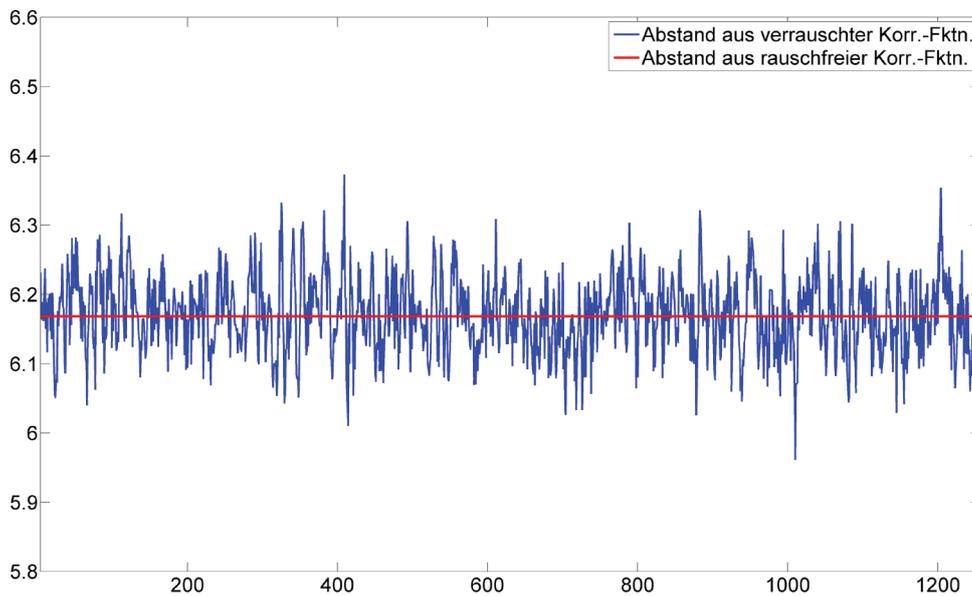


Abbildung 5.14: Simulierte Abstandswerte einer ideal angenommenen Korrelationsfunktion (rot) und einer verrauschten Korrelationsfunktion (blau)

Der rote Verlauf stellt den simulierten Entfernungswert eines Pixels bei Annahme einer störungsfreien Korrelationsfunktion dar. Der blaue Verlauf repräsentiert den berechneten Entfernungswert infolge der verrauschten Abtastwerte der Korrelationsfunktion. Die Modulationsfrequenz $f_{\text{mod}} = 20 \text{ Mhz}$, der Abstand zum Punktziel $R_{\text{ideal}} = 6,17 \text{ m}$, es wurden insgesamt 1250 Messungen mit der Standardabweichung der Phase der Korrelationsfunktion $\sigma(\text{phase}(k_{n,Rausch}))$ von 0,5 Grad simuliert. Wie man aus der Abbildung erkennen kann, ruft ein solcher Fehler der Phase der Korrelationsfunktion alleine eine Messunsicherheit im Bereich einiger Zentimeter hervor: $\sigma(R_{Rausch}) = 5,52 \text{ cm}$.

Die Abbildung 5.15 zeigt das Verteilungshistogramm der in Abbildung 5.14 dargestellten aus der verrauschten Korrelationsfunktion berechneten Entfernungswerte.

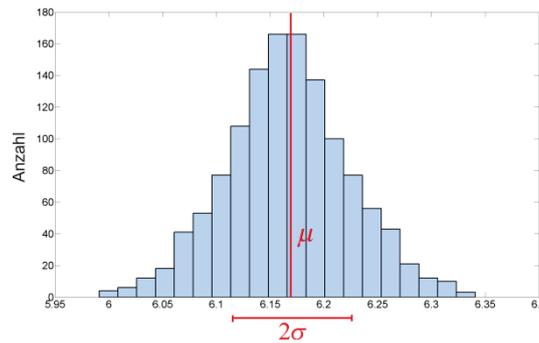


Abbildung 5.15: Abweichung der verrauschten Abstandswerte von dem tatsächlichen Abstand in Metern

Die zeitdiskrete Berechnung der simulierten Phasenverschiebungen aus den vier einzelnen Korrelationsfunktionen (nach 5.17) erfordert die Kenntnis der Objektpositionen bzw. der Koordinaten der einzelnen Knoten der Objekte zu den Zeitpunkten, zu denen die Korrelationsfunktionen gebildet werden. Diese Zeitpunkte stehen in einem festen Verhältnis zu der Grundfrequenz des Sendesignals (z.B. $\Delta\tau = T_0/4$), welche im Simulator eine veränderbare Größe ist bzw. sein kann.

3D Objektszenen liegen im Allgemeinen in einer Form vor, bei der die Koordinaten der einzelnen Knoten bezüglich der Zeitachse zeitäquidistant vorgegeben sind, d.h. zu jedem Zeitpunkt $n \cdot \Delta t$ ($n \in [0, N_{time} - 1]$ mit $N_{time} = \text{Anzahl der Zeitschritte}$) existiert ein Datensatz mit den entsprechenden Koordinaten. Werden die Objektsequenzen selbst erzeugt, dann lassen sich die Sequenzzeitpunkte – die Punkte, an denen ein Datensatz der Simulationsszene generiert wird – oft frei vorgeben. Bei fremd-generierten Datensätzen entsprechen die Sequenzzeitpunkte aber in der Regel nicht den Zeitpunkten, an denen die Korrelationsfunktionen gebildet werden sollen, d.h. im Allgemeinen gilt: $\Delta\tau \neq \Delta t$. Somit wird das „Einfrieren“ und Abbilden einer Szene zu den Abtastzeitpunkten der Korrelationsfunktionen unmöglich – es liegen zu diesem Zeitpunkt keine Daten vor.

Um dem Problem entgegenzuwirken, wurde ein alternativer Lösungsansatz realisiert, der besonders hinsichtlich der Simulation des Effekts der Bewegungsunschärfe, welcher bei allen bildgebenden Sensoren vor allem bei längeren Integrationszeiten physikalisch bedingt entsteht, dem tatsächlichen Integrationsprozess des Sensors nahe kommt. Dieser wird nachfolgend vorgestellt.

5.8 Betrachtung der Bewegungsunschärfe

In der Simulationstechnik haben sich je nach Einsatzgebiet und gestellten Anforderungen unterschiedliche Simulationsmethoden etabliert. Bei dynamischen Simulationen unterscheidet man zwischen kontinuierlicher, diskreter und hybrider Simulation. Bei

diskreten Simulationsmodellen werden die Änderungen des Simulationszustandes zu bestimmten diskreten Zeitpunkten betrachtet. In jedem zeitdiskreten Simulationsmodell existiert somit zwischen dem Simulationszustand und der Simulationszeit eine feste Beziehung. Bei zeitkontinuierlichen Simulationen ändern sich die Systemgrößen kontinuierlich, d.h. in beliebig kleinen Zeitabschnitten.

Betrachtet man Kamera- und Beleuchterbewegungen sowie die dynamischen Abläufe in der Simulationsszene, so stellen diese einen kontinuierlichen Prozess dar und lassen sich theoretisch beliebig fein auflösen und darstellen. Dagegen findet die Bildaufnahme einer Kamera, wie bereits erwähnt, zu festdefinierten Zeiten statt, die an die Integrationszeit der Kamera gekoppelt sind. So können theoretisch bei einer Integrationszeit von 20 ms maximal 50 Bilder/Sek. aufgenommen werden, bei einer Integrationszeit von 25 ms – 40 Bilder/Sek. usw. Praktisch sind die Bildraten bei solchen Integrationszeiten deutlich niedriger, weil die Bilder noch ausgelesen, verarbeitet und übertragen werden. Während der Integrationszeit wird die Szene kontinuierlich aufgenommen. Abhängig von der Integrationszeit der Kamera und dem Ausmaß der Szenenänderungen während der Aufnahmezeit eines einzelnen Bildes (Frame), kommt es bei jeder dynamischen Anordnung unfreiwillig zu natürlicher Bewegungsunschärfe. Die Kamera kann nicht beliebig schnell ein Motiv „einfrieren“. Dieser Effekt nimmt mit steigender Integrationszeit zu. Würde man im Simulator zu festdefinierten Zeitpunkten die Position aller beteiligten Komponente bestimmen und diese von der virtuellen Kamera per „Schnappschuss“ aufnehmen, so würde jede Aufnahme ein scharfes Bild ergeben. Um die Bewegungsunschärfe zu simulieren, wird der kontinuierliche Ansatz für die Objektbewegungen mit dem diskreten Ansatz der Bildgenerierung über Definition der Zwischenframes zusammengeführt. Das Prinzip ist in Abbildung 5.16 dargestellt.

Alle Abläufe innerhalb des Simulators sind an eine einheitliche Zeitachse gebunden, die äquidistant diskretisiert wird. Die Dauer des minimalen Zeitabschnittes ist fest und beträgt Δt . Die Bewegungsabläufe sind direkt an die Zeitachse gebunden, die Aktualisierung der Werte findet jeweils nach der Zeit Δt statt. Parallel dazu werden zu den diskreten Zeitwerten Zwischenframes berechnet. Diese stellen im Prinzip eine Momentaufnahme der aktuellen Szene dar. Betrachtet man den Aufnahmeprozess einer realen Kamera, so wird das Bild dagegen kontinuierlich über die Dauer der Integrationszeit aufgenommen. Um diesen Prozess zu simulieren, werden nun einige Zwischenframes zu einem Frame zusammengefasst. Die Anzahl der Zwischenframes in einem Frame ergibt sich aus dem Verhältnis der Integrationszeit zur Dauer der Zeitdiskretisierung $F = T_{\text{int}} / \Delta t$. Beide sind frei definierbar. Genau genommen wird der Simulationsprozess durch das Verhältnis $F \in \mathbb{N}$ der beiden Größen gesteuert.

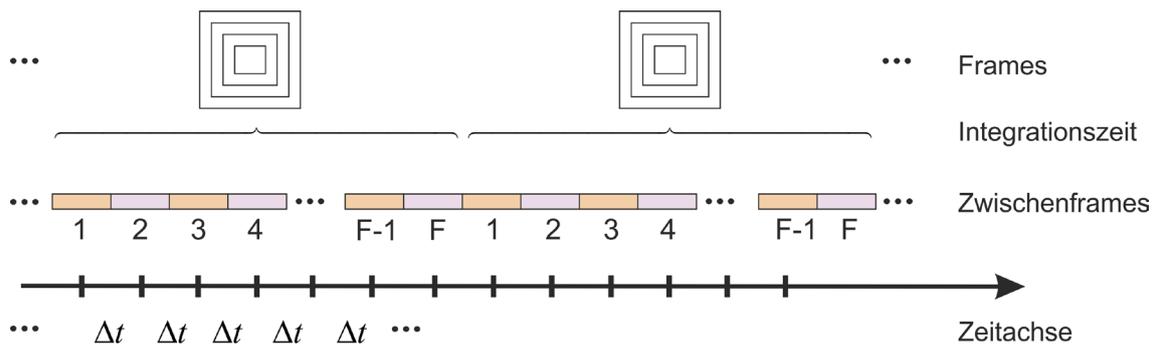


Abbildung 5.16: Das Prinzip der Berechnung eines Frames aus Zwischenframes

Mathematisch gesehen wird bei der Bestimmung eines Frames die Historie eines Datensatzes aus F Zwischenframes betrachtet. Die Summe für die theoretische Antwort des Sensors (5.13) wird zu:

$$\bar{r}_{Integration} = \sum_{f=1}^F \bar{r}(\vec{r}_{I_m P_n}, \vec{r}_{P_n}, I_{I_m P_n}, I_{refl, korrr}, \lambda, \alpha, \rho_R, \chi_{Sim}) \quad (5.19)$$

Im Kapitel 9.2 wird ein Vergleich der Simulationsergebnisse bei unterschiedlichen Ansätzen für die Integrationszeit dargestellt.

5.9 Quantisierung der Messwerte

Die gemessenen und ausgewerteten Messwerte einer PMD-Kamera werden im Rahmen der digitalen Verarbeitung, der Möglichkeiten der eingesetzten Hardware sowie der zu erfüllenden Spezifikationen quantisiert. Unter Quantisierung versteht man die Bewertung der Helligkeit oder der Entfernung eines Pixels mittels einer festgelegten Grauwertmenge. Es findet also eine Diskretisierung des Messbereiches sowie der Helligkeitswerte statt.

$$f(x) \in [0, R_{\max}] \rightarrow f[m] \in [0, Q-1] \quad (5.20)$$

mit Q – Anzahl der Quantisierungsstufen

Unterschiedliche Kameraausführung bieten dabei unterschiedliche Auflösungen des Messbereiches. So wird der Wertebereich einer PMD19k mit 12Bit aufgelöst, der Sensor der 41k-S2 – mit 16Bit.

Modulationsfrequenz	Eindeutigkeitsbereich	Max. Quantisierungsfehler
10 MHz	15 m	3,7 mm
20 MHz	7,5 m	1.8 mm
30 MHz	5 m	1,2 mm

Tabelle 5.2: Maximaler Quantisierungsfehler in Abhängigkeit des Eindeutigkeitsbereichs

Der Speicherbedarf für ein Bild der Auflösung $N \times M$ errechnet sich zu:

$$b = N \cdot M \cdot \log_2(Q) \quad (5.21)$$

Es ist möglich, im PMD-Simulator die Auflösung in bits anzugeben. Der berechnete Wert wird dann entsprechend der Quantisierungsvorgaben auf nächsthöheren Quantisierungswert angehoben.

6 Sichtbarkeitsanalyse

Eine der wichtigsten Fragestellungen bei der Entwicklung des PMD-Simulators stellt die Frage der Sichtbarkeit der Szenenteile dar. Die Berechnung der Beleuchtungsverhältnisse von einer Lichtquelle in einem Punkt erfordert die Vorkenntnisse, ob dieser Punkt von der besagten Lichtquelle physikalisch erreicht werden kann. Bei der Berechnung der Beleuchtungsverhältnisse eines ganzen Beleuchters oder einer Anordnung aus mehreren Beleuchtern muss die Analyse für jede einzelne Lichtquelle dieser Anordnung durchgeführt werden. Das gleiche gilt auch für die Detektierbarkeit eines Objektpunktes seitens Sensorpixels bzw. eines ganzen Kamerasensors. Das *Visibility Check Module (VCM)*, welches für die Berechnung der Sichtbarkeit einer Szene entwickelt wurde, stellt somit das mit am meisten im Simulator aufgerufene Modul, bestehend aus mehreren Routinen, dar. Besteht die Szene aus mehr als 1000 Polygonen (was in der Regel üblich ist), dann werden mehr als 90% der Berechnungen im Bereich der Sichtbarkeitsanalyse gemacht.

Wie in Kapitel 5.1 beschrieben, werden zwei Formen der Objektdefinition unterstützt – die Punkteform und die Polygonform. Im Gegensatz zur polygonalen Form gibt es bei der Punkteform keine zusätzlichen Informationen über die Nachbarschaftsbeziehungen. Es existieren somit bei den Objekten keine Oberflächen im wirklichen Sinne. Diese Oberflächen müssen errechnet werden.

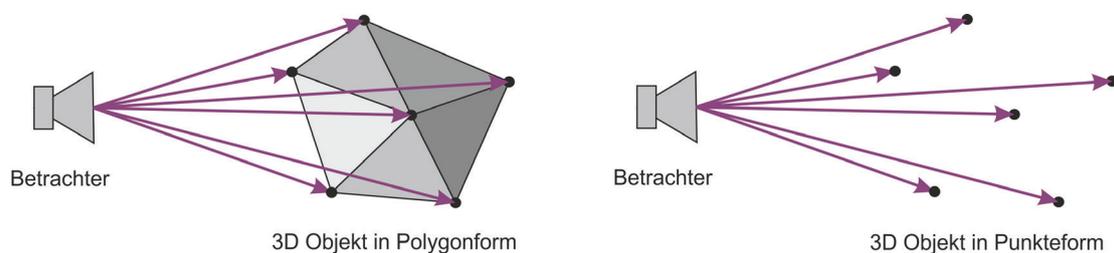


Abbildung 6.1: Zwei Definitionsformen von Objekten – die polygonale Oberflächenstruktur und oberflächenlose Punktedarstellung der gleichen Szene

Die Aufgabe des *Visibility Check Module* lässt sich relativ leicht definieren – es sollen alle Bereiche der Szene im Datensatz markiert werden, die von einer beliebigen Position aus nicht gesehen werden können. Unter dem Begriff „sehen“ wird dabei das virtuelle Sehen gemeint.

Die Sichtbarkeitsanalyse wird in der Computergrafik allgemein in zwei Problemstellungen unterteilt – die Berechnung der Kernschatten und das Problem der Verdeckung von Objekten. Durch gezielte Selektion dieser Bereiche wird die Komplexität einer Szene in der Regel um mindestens 50% reduziert und dadurch die Darstellungsgeschwindigkeit deutlich erhöht, da die verdeckten Szenen erst gar nicht gezeichnet und gemappt werden müssen. Das Hauptziel des VCM ist dagegen das Extrahieren der verdeckten Bereiche,

weil sie keinen Beitrag bei der Berechnung der theoretischen Antwort des Sensors leisten dürfen. Die Problemstellungen bleiben jedoch die gleichen.

Wie bereits erwähnt, muss die Sichtbarkeitsanalyse sowohl bei der Berechnung der Lichtausbreitung, als auch bei der Berechnung der zu einem Pixel korrespondierenden Szenenbereiche durchgeführt werden. Die Ansätze in den einzelnen Bereichen verfolgen zwar das gleiche Ziel, unterscheiden sich jedoch in ihrer Ausgangssituation. So wird von einer Punktlichtquelle, die das Licht in alle Richtungen gleichermaßen ausstrahlt, mehr von einer Szene beleuchtet, als von einem Richtstrahler, der nur in eine bestimmte Richtung Licht abgibt. Die Abbildung 6.2 verdeutlicht die Problematik.

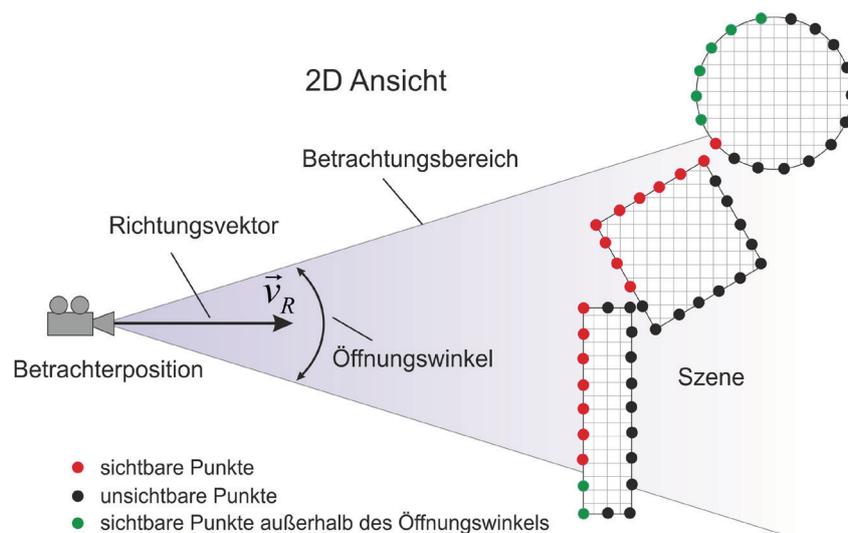


Abbildung 6.2: Problemstellung der Sichtbarkeitsanalyse

Als Ausgangssituation sind die Position des Betrachters, der Richtungsvektor (mit Ausnahme einer Punktlichtquelle), die Position und Orientierung der Szene und der Öffnungswinkel, bzw. die Geometrie des Blickfeldes bekannt. So können die Punkte einer Szene sichtbar, unsichtbar oder aber auch sichtbar und sich außerhalb des Öffnungswinkels befindend sein.

Ein möglicher Ansatz wäre die Szene schrittweiser zuerst auf Sichtbarkeit hin zu überprüfen und anschließend auf die Zugehörigkeit zu dem Betrachtungsbereich. Unter dem Betrachtungsbereich wird der Bereich der Szene verstanden, der bei einer Kamera mit dem Sensor aufgenommen werden kann oder bei einer Lichtquelle in dem Öffnungswinkel der Lichtquelle liegt. Bei einem Sensor wird der Bereich von einer Pyramide eingeschlossen, eine Lichtquelle beleuchtet definitionsgemäß in einem Kugelkegel. Die sichtbaren Punkte würden anschließend im Rahmen der Abbildung der Szene auf den Sensor später automatisch selektiert und den entsprechenden Pixeln zugewiesen. Die Punkte, die aufgrund ihrer Position außerhalb des Öffnungswinkels nicht von der Kamera detektiert werden können, werden dabei ebenfalls herausgesiebt. Diese Methode funktioniert im Simulator aufgrund der Reihenfolge der Algorithmen und der Abfragen sozusagen automatisch. Sie hat aber einen entscheidenden Nachteil. Aufgrund der zu-

nächst generellen Analyse der Sichtbarkeit einer Szene, also eines Datensatzes, aus einem Punkt heraus, ergibt sich meistens ein deutlich größerer Datensatz, welcher in späteren Schritten sequentiell abgearbeitet wird. Es ist vergleichbar mit einem Menschen in einem Raum, welchen er verlassen sollte. Wenn er sich erst mal komplett umschaute und erst dann durch eine Tür hinausgeht, deren Richtung er eigentlich von Anfang an kennt, dann dauert es wesentlich länger, als wenn man direkt zur Tür läuft. Genauso kann ein Datensatz bereits vor der Sichtbarkeitsanalyse daraufhin ausgewertet werden, welche Teile davon im Betrachtungsbereich liegen. Denn sollte ein Punkt außerhalb dieses Bereiches liegen, sind weitere Berechnungen zur Sichtbarkeit und evtl. Berechnungen der Abbildungen auf den Sensor überflüssig.

Die Modellierung des *Visibility Check Module* muss dementsprechend anfragenorientiert sein. Bevor die eigentliche Sichtbarkeitsanalyse durchgeführt werden kann, gibt es, je nachdem von welchem Modul des Simulators das VCM angesteuert wird, drei unterschiedliche Szenarios – die Berechnung der Sichtbarkeit ausgehend von einem Punkt, ausgehend von einer Kamera und ausgehend von einer Lichtquelle. Im ersten Fall wird die Objektszene nicht durch einen vorgegebenen Betrachtungsbereich eingeschränkt, die Sichtbarkeitsanalyse kann also direkt durchgeführt werden. Die beiden anderen Fälle erfordern eine weitere Voranalyse.

6.1 Sichtbarkeit: Voranalyse für einen Sensor

Wie bereits erwähnt, ergibt sich bei einer Kamera durch die rechteckige Definition des Sensors (nicht mit der Definition eines Pixels zu verwechseln) ein pyramidenförmiger Sichtbereich. Bei der Voranalyse geht es darum, zu bestimmen, welche Teile einer gegebenen Szene außerhalb des Sichtbereiches der Kamera liegen. In Abbildung 6.3 ist die Geometrie des Sichtbereiches dargestellt. Der Bereich ergibt sich durch die Abbildungen der Eckpunkte \vec{P}_1 bis \vec{P}_4 des Sensors durch die Optik in den Raum. Die Abbildungen der Vektoren $\vec{P}_1\vec{O}$ bis $\vec{P}_4\vec{O}$ sind erstens durch die Position der Optik, bzw. des Mittelpunktes der Optik, und zweitens durch die Abbildungsvorschrift der Optik selbst vorgegeben (s. Kapitel 7). Der Vektor \vec{h}_m ist der Vektor zwischen der Mitte des Sensors und zum Mittelpunkt der Optik und gibt Hauptrichtung des Sichtbereiches vor. $\vec{h}_{m,r}$ stellt seine Abbildung in den Raum dar, welche im Vorzeichen gedreht und entsprechend der Linsenverzeichnung transformiert ist.

Es wird vorerst angenommen, dass es keine Linsenverzeichnung gibt. Demnach entspricht der Vektor $\vec{h}_{m,r}$ dem Vektor \vec{h}_m . Falls der Sensor gegenüber der optischen Achse verschoben und verdreht ist, dann steht der Vektor \vec{h}_m nicht senkrecht auf der Sensorebene. Der Sichtbereich des Sensors ergibt somit allgemein eine schiefe Pyramide. Bei

der Voranalyse geht es darum, zu überprüfen, welche Punkte der Szene innerhalb der vom Sensor aufgespannten Pyramide mit der theoretisch unendlichen Höhe liegen.

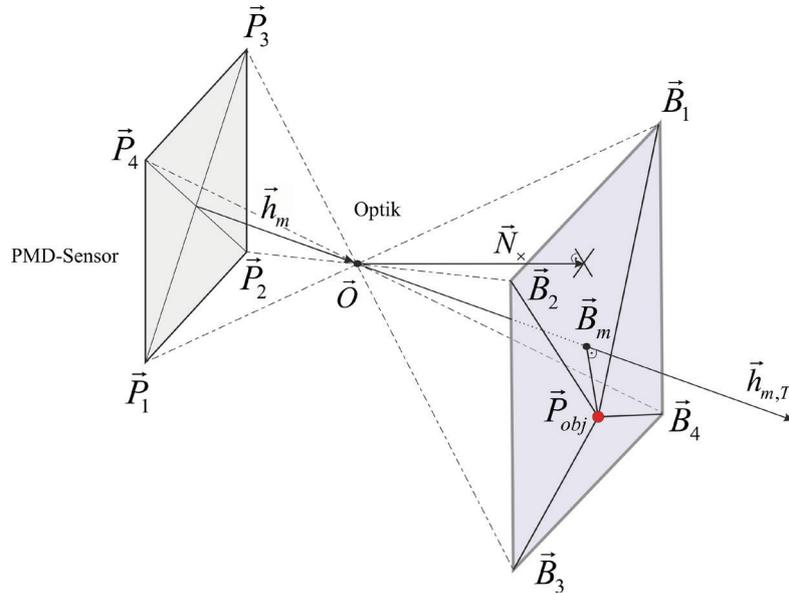


Abbildung 6.3: Das Prinzip der Voranalyse für eine Kamera

Dabei hat sich der folgende vektorielle Ansatz als sehr robust und relativ schnell erwiesen. Für einen beliebigen Punkt der Szene \vec{P}_{obj} wird senkrecht zum Vektor $\vec{h}_{m,T}$ eine Ebene definiert. Diese Ebene bildet als Grund mit den vier Abbildungen der Vektoren $\vec{P}_1\vec{O}$ bis $\vec{P}_4\vec{O}$ eine Pyramide – $\vec{B}_1\vec{B}_2\vec{B}_3\vec{B}_4\vec{O}$ (siehe Grafik). Innerhalb dieser Ebene wird überprüft, ob der Punkt \vec{P}_{obj} von dem Viereck $\vec{B}_1\vec{B}_2\vec{B}_3\vec{B}_4$ eingeschlossen ist oder nicht. Wir halten an dieser Stelle fest, dass die Ebene $\vec{B}_1\vec{B}_2\vec{B}_3\vec{B}_4$ nicht parallel zur Sensorebene verläuft.

In Parameterform wird eine Gerade durch zwei Punkte mit den Ortsvektoren \vec{p} und \vec{q} beschrieben durch:

$$\vec{x} = \vec{p} + \lambda(\vec{q} - \vec{p}) \quad (6.1)$$

Im Falle des Vektors $\vec{h}_{m,T}$ ergibt sich (6.1) mit $\vec{p} = \vec{O}$ und $\vec{q} - \vec{p} = \vec{h}_{m,T}$ zu

$$\vec{x} = \vec{O} + \lambda \cdot \vec{h}_{m,T} \quad (6.2)$$

Für die Ebene durch den Punkt \vec{P}_{obj} senkrecht auf die Gerade \vec{x} gilt:

$$(\vec{q} - \vec{p}) \cdot [\vec{x} - \vec{P}_{obj}] = \vec{h}_{m,T} \cdot [\vec{x} - \vec{P}_{obj}] = \vec{h}_{m,T} \cdot \vec{x} - \vec{h}_{m,T} \cdot \vec{P}_{obj} = 0 \quad (6.3)$$

Die Bedingung für den Schnittpunkt der Geraden mit der Ebene ist:

$$\vec{h}_{m,T} \cdot (\vec{O} + \lambda \cdot \vec{h}_{m,T}) - \vec{h}_{m,T} \cdot \vec{P}_{obj} = 0 \quad (6.4)$$

Daraus folgt für den Parameter λ im Schnittpunkt:

$$\lambda = \frac{\vec{h}_{m,T} \cdot \vec{P}_{obj} - \vec{h}_{m,T} \cdot \vec{O}}{h_{m,T}^2} = \frac{\vec{h}_{m,T} \cdot (\vec{P}_{obj} - \vec{O})}{h_{m,T}^2} \quad (6.5)$$

Jetzt wird λ in die Geradengleichung eingesetzt und man erhält die Koordinaten des Schnittpunktes:

$$\vec{B}_m = \vec{O} + \frac{\vec{h}_{m,T} \cdot (\vec{P}_{obj} - \vec{O})}{h_{m,T}^2} \cdot \vec{h}_{m,T} \quad (6.6)$$

Der Ausdruck (6.6) gilt allgemein für alle definierten orthogonalen Koordinatensysteme und beschreibt die Projektion des Vektors $\overrightarrow{OP_{obj}}$ auf den Vektor $\vec{h}_{m,T}$.

Nun können die Schnittpunkte der Abbildungen $\overline{P_1O}$ bis $\overline{P_4O}$, also \vec{B}_1 bis \vec{B}_4 , bestimmt werden. Für die Normalenform der Ebene durch den Punkt \vec{B}_m mit dem Normalenvektor $\vec{h}_{m,T}$ auf die Ebene gilt:

$$\vec{h}_{m,T} \cdot (\vec{B} - \vec{B}_m) = 0 \quad \text{bzw.} \quad \vec{h}_{m,T} \cdot \vec{B} - \vec{h}_{m,T} \cdot \vec{B}_m = 0 \quad (6.7)$$

mit \vec{B} – Koordinaten eines beliebigen Punktes in der Ebene

Für den Schnittpunkt der Geraden $\overline{P_iO}$ mit der Ebene gilt mit (6.1):

$$\begin{aligned} \vec{h}_{m,T} \cdot (\vec{P}_i + \lambda_i \cdot (\vec{O} - \vec{P}_i)) - \vec{h}_{m,T} \cdot \vec{B}_m &= 0 \\ \Rightarrow \lambda_i &= \frac{\vec{h}_{m,T} \cdot (\vec{B}_m - \vec{P}_i)}{\vec{h}_{m,T} \cdot (\vec{O} - \vec{P}_i)} \end{aligned} \quad (6.8)$$

Den Parameter in die Geradengleichung eingesetzt ergibt den gesuchten Schnittpunkt:

$$\begin{aligned} \vec{h}_{m,T} \cdot (\vec{P}_i + \lambda \cdot (\vec{O} - \vec{P}_i)) - \vec{h}_{m,T} \cdot \vec{B}_m &= 0 \\ \Rightarrow \vec{B}_i &= \vec{P}_i + \frac{\vec{h}_{m,T} \cdot (\vec{B}_m - \vec{P}_i)}{\vec{h}_{m,T} \cdot (\vec{O} - \vec{P}_i)} \cdot (\vec{O} - \vec{P}_i) \end{aligned} \quad (6.9)$$

Nun kann der Sichtbereich des Sensors durch die Koordinaten der vier Punkte \vec{B}_1 bis \vec{B}_4 eingegrenzt werden. Dies geschieht nach dem in Kapitel 12.1 (siehe Fall 2) beschriebenen Verfahren. Dabei wird mit Hilfe der Flächeninhalte der vier Dreiecke überprüft, ob der Punkt \vec{P}_{obj} innerhalb des Sichtbereiches liegt oder außerhalb:

$$A(\vec{B}_1, \vec{B}_2, \vec{P}_{obj}) + A(\vec{B}_2, \vec{B}_3, \vec{P}_{obj}) + A(\vec{B}_3, \vec{B}_4, \vec{P}_{obj}) + A(\vec{B}_1, \vec{B}_4, \vec{P}_{obj}) \stackrel{?}{=} A(\vec{B}_1, \vec{B}_2, \vec{B}_3, \vec{B}_4) \quad (6.10)$$

mit $A(\vec{B}_1, \vec{B}_2, \vec{P}_{obj})$ – Flächeninhalt des Dreiecks, das durch \vec{B}_1 , \vec{B}_2 und \vec{P}_{obj} aufgespannt wird usw. Sind die beiden Seiten des Ausdruckes gleich, so gilt der Szenenpunkt als erst einmal potentiell detektierbar und wird in weiteren Schritten auf absolute Detektierbarkeit hin überprüft.

Für den Fall einer eingeschalteten Linsenverzeichnung (siehe Kapitel 7) kann die hier vorgestellte Vorgehensweise nicht eingesetzt werden, da aufgrund von optischen Verzerrungen die Regeln der linearen Abbildung der Vektoren nicht mehr gelten (Geraden werden nicht auf Geraden abgebildet). In einem solchen Fall wird das in Abschnitt 12.1 vorgestellte Verfahren eingesetzt, welches zwar langsamer ist, jedoch unabhängig von der modellierten Abbildungsvorschrift funktioniert.

6.2 Sichtbarkeit: Voranalyse für eine Lichtquelle

Bei einer Lichtquelle ist der Beleuchtungsbereich radial um die Strahlhaupttrichtung \vec{h}_m durch den Öffnungswinkel $\varphi = 2\varphi_h$ vorgegeben (siehe Abbildung 6.4).

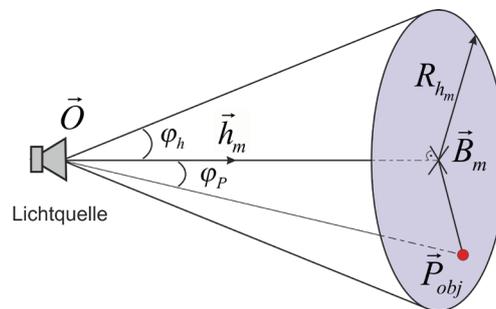


Abbildung 6.4: Das Prinzip der Voranalyse für eine Lichtquelle

Der Öffnungswinkel wird durch den Definitionsbereich der Lichtquelle bestimmt. Üblicherweise liegt er bei gängigen PMD Beleuchtern im Bereich $0 < \varphi \leq \pi$. Sollte der Öffnungswinkel π oder $\pi - \varepsilon$, mit $\varepsilon \ll 1$ (wie z.B. bei einem Lambert-Strahler) nicht übersteigen, so können die Punkte, die hinter einer Lichtquelle liegen, und somit nicht beleuchtet werden können, mit einer einfachen Abfrage bestimmt werden: $z_{LKS}(P_i) < 0$. Alle Punkte hinter einer Lichtquelle werden im Koordinatensystem der Lichtquelle per Definition negative z-Koordinaten besitzen. Für den sonstigen Bereich von $0 < \varphi < 2\pi$ kann die Zugehörigkeit zu dem Belichtungsbereich sehr einfach anhand des Winkelvergleichs bestimmt werden:

Eine Facette ist potenziell sichtbar, wenn der Winkel zwischen der Normalen auf die Facette und dem Blickwinkel zur Facette hin kleiner als 90° ist, anderenfalls ist sie unsichtbar. Das Verfahren heißt also:

$$\text{Prüfe } \varphi < 90^\circ? \quad \text{oder } \varphi \geq 90^\circ \quad (6.12)$$

$$\text{mit } \cos(\varphi) = \frac{\vec{n} \cdot \vec{v}_R}{|\vec{n}| \cdot |\vec{v}_R|}$$

Die Normale auf die Facette ist durch das Kreuzprodukt der beiden Kantenvektoren $\overrightarrow{P_2P_1}$ und $\overrightarrow{P_2P_3}$ gegeben:

$$\vec{a} = \vec{P}_3 - \vec{P}_2 \quad \vec{b} = \vec{P}_1 - \vec{P}_2 \quad (6.13)$$

$$\vec{n} = \vec{a} \times \vec{b} = \begin{bmatrix} i & j & k \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{bmatrix} \quad (6.14)$$

Der Vektor \vec{v}_R ist die Blickrichtung, also der Vektor vom Betrachter aus zu einem beliebigen Punkt innerhalb der Facette. Es bietet sich an, dafür einen bereits vorhandenen Punkt zu nehmen, z.B. einen Facettenknoten

$$\vec{v}_R = \vec{P}_1 \quad \text{oder auch möglich mit} \quad \vec{v}_R = \frac{1}{4}(\vec{P}_1 + \vec{P}_2 + \vec{P}_3 + \vec{P}_4) \quad (6.15)$$

Auf diese Weise muss für jedes Polygon der Szene überprüft werden, ob es sich um die Schattenseite handelt oder nicht. Das Problem beim *Backface Culling* ist, dass es kein vollständiges Sichtbarkeitsverfahren ist. Die Verdeckung von Objekten, das sogenannte *Hidden Surface Removal* (HSR) Problem, kann damit nicht ausgewertet und gelöst werden. Daher wird der Datensatz einer weiteren, auf dem Prinzip des *Ray Casting* basierten Sichtbarkeitsanalyse unterworfen.

Angepasstes Ray Casting

Beim *Ray Casting* wird für jedes Pixel getrennt berechnet, was von dort aus sichtbar ist. Dabei werden von jedem Pixel aus in Blickrichtung ein Blickstrahl definiert und dessen Schnittpunkte mit dem Polygonnetz bestimmt. Anhand der Entfernung der Schnittpunkte von dem Betrachter wird dasjenige Polygon gerendert, welches zum Betrachter am nächsten liegt.

Ähnlich wie bei herkömmlichen 3D-Grafikkarten, bei denen im Z-Buffer gespeichert wird, wie weit ein Polygon von der Bildschirm-Projektionsfläche bzw. vom Betrachter entfernt ist (bei letzterem spricht man auch vom W-Buffer), wird für jedes Polygon seine Entfernung zum Betrachter bestimmt. D.h. es findet für jedes Polygon ein Z-Test statt. Liegt auf aktueller Position schon ein Polygon "näher" zum Sensor als das neue,

wird das gerenderte Dreiecks-Pixel nicht in den *Backbuffer* geschrieben, sondern verworfen. Ansonsten wird es in den *Backbuffer* "gemalt", und der für dieses Polygon gültige Z-Wert in den Z-Buffer geschrieben, womit dann das vorherige Polygon überdeckt wird. Der Unterschied zu den Rendering-Verfahren bei Grafikkarten besteht darin, dass die Grafikkarten für jedes Bildschirmpixel jeweils einen einzigen Tiefenwert (Z-Wert) im Z-Buffer halten. Dies würde bei der PMD-Simulation bedeuten, dass für jedes einzelne Pixel nur ein einziger Wert und zwar der mit dem minimalen Abstand übrig bleibt. Eine solche Reduzierung des Datensatzes würde bei der Berechnung der simulierten Antwort einen deutlichen Fehler hervorrufen. Besonders stark wäre der Fehler bei den Pixeln, in deren Öffnungswinkeln zur Kamera hin stark angewinkelte Objekt-oberflächen oder deutliche Tiefenunterschiede (Z-Werte) vorkämen. Um diesem Problem entgegenzuwirken, wird für jedes Pixel im Simulator nicht nur von einem Blickstrahl gescannt, sondern der Blickstrahl fährt dabei einen vordefinierten Pfad ab und scannt somit die Umgebung ab. Das Prinzip ist in Abbildung 6.6 gezeigt.

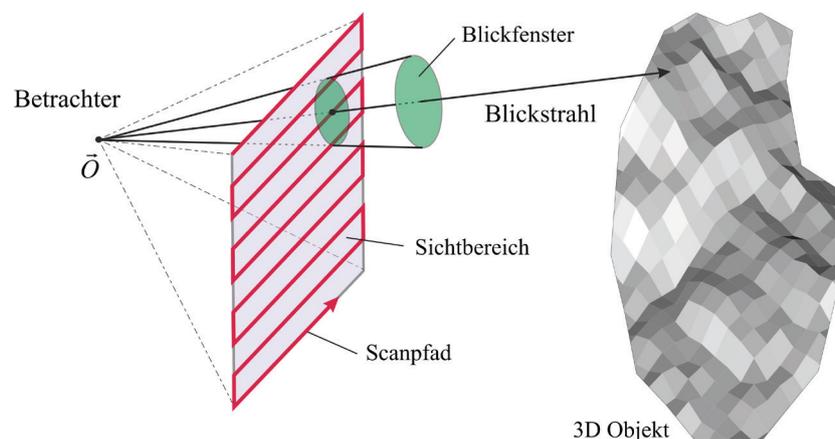


Abbildung 6.6: Das Prinzip des angepassten Ray Casting

Das Problem dabei ist, dass die Anzahl der zu bewältigenden Berechnungsschritte enorm anwächst. Möchte man eine aus 10.000 Polygonen bestehende Szene im Öffnungswinkel einer mit 160x120 Pixel aufgelösten Kamera auf Sichtbarkeit überprüfen, so ergeben sich $192 \cdot 10^6$ Blickstrahlrichtungen pro Pixel, wenn dabei nur ein einziger Strahl verfolgt wird. Beim Scannen steigt die Anzahl um Faktor 25 bis 200 an. Der Rechenaufwand kann jedoch erheblich reduziert werden, wenn beim Scannen nicht immer aufs Neue die komplette Szene untersucht werden muss, sondern die Suche nur auf die Bereiche der Szene beschränkt wird, deren Richtungsvektoren in der näheren Umgebung des Lichtstrahles liegen. In einer Voranalyse der Szene kann der Richtungsvektor zu jedem Polygon hin und die Entfernung bestimmt werden. Markiert man die entsprechenden Polygone in einer zusätzlichen Tabelle mit einem Zeiger entsprechend ihrer Richtung, so lassen sich später zu einer bestimmten Richtung die naheliegenden Polygone bestimmen und zwar ohne großen rechnerischen Aufwand. Die Sichtbarkeitsanalyse findet somit in einem kleinen Bereich um den Blickstrahl herum statt – in dem Blickfenster (siehe Abbildung 6.6). Die Adressierung kann mit Hilfe von Zeigern reali-

siert werden. Dies erspart einerseits die langsamen Dateizugriffe und minimiert andererseits den erforderlichen Datendurchsatz.

Das Prinzip der Datenreduktion ist in Abbildung 6.7 dargestellt. Die Richtungsvektoren der einzelnen Polygone werden in ihre vertikalen und horizontalen Anteile zerlegt. Der gesamte Öffnungsbereich wird durch eine Tabelle repräsentiert. Die Zeilen dieser Tabelle repräsentieren die gerasterten vertikalen Auslenkungsbereiche, die Spalten – die gerasterten horizontalen Auslenkungsbereiche. Die bei der Voranalyse berechneten Richtungsvektoren der Polygone „markieren“ im gleichen Schritt die entsprechenden Zellen der Tabelle. In diese Zelle wird ein Zeiger auf das gerade berechnete Polygon gelegt. D.h. nach der Voranalyse besitzt man das Wissen darüber, welche Polygone zu einem bestimmten Raumsegment gehören.

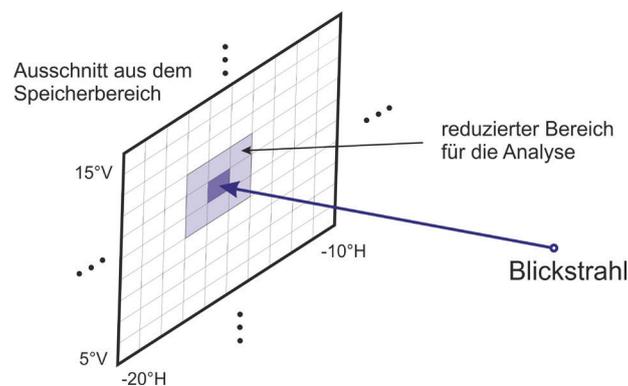


Abbildung 6.7: Das Prinzip der Modellierung des Blickfensters

Bei dem anschließenden Ray Casting geht man den umgekehrten Weg. Der Vektor der Blickstrahlrichtung wird in seinen vertikalen und horizontalen Anteil zerlegt und markiert somit eine bestimmte Zelle in der Tabelle. Um diese Zelle herum werden entsprechend dem Öffnungswinkel des Betrachters, bzw. des Sensors (und diese sind aus der Geometrie bekannt), die Teile der gesamten Szene ausgewählt, die für die Sichtbarkeitsanalyse in Betracht kommen können.

6.4 Sichtbarkeit: Nachanalyse für Objekte in Punkteform

Im Gegensatz zur polygonalen Form der Objekte, geht es bei der Punkteform um eine reine Ansammlung von Punkten im 3D Raum, die zueinander keinen Bezug haben. Die benachbarten Punkte in einem Datensatz müssen nicht zwangsweise benachbarte Punkte im Raum repräsentieren. Solche Datensätze entstehen, wie bereits erwähnt, beispielsweise durch Abtastung von Objekten mittels 3D-Scanner. Auch können solche Datensätze mit Hilfe einiger 3D Modellierungswerkzeuge, z.B. mit Blender, erstellt werden.

Das Problem der Berechnung der Sichtbarkeit von solchen Objekten liegt auf der Hand. Da diese Form keine Oberflächeninformationen bietet, können keine Polygonnetze bzw.

geschlossene Objekte oder zumindest geschlossene Oberflächenbereiche definiert werden. Es gibt zwar viele Triangulationsverfahren aus solchen Datensätzen polygonale Datensätze zu berechnen, indem man versucht, durch die räumlich benachbarten Punkte mit Hilfe der Disparitätsschätzung in der Art Polygone zu legen, dass eine geschlossene Oberfläche entsteht, die Methoden sind jedoch meistens sehr aufwendig bezüglich Laufzeit- und Speicherbedarf. Auch sind die Implementierungen selbst bei einfachen Ansätzen oft sehr komplex, da eine große Menge von Informationen verwaltet werden muss (räumliche Beziehungen der Punkte untereinander, Punkt-Kante-Beziehungen, Punkt-Facette-Beziehungen, bereits ermittelte Triangulationen, Historie der räumlichen Suchvorgänge usw.). Als klassische Triangulationsverfahren können z.B. Delaunay-Triangulation [McCu 80] oder auch MPU Implicits [Alexa 01], [Ohta 03] genannt werden.

Daher wurde bei der Modellierung der Sichtbarkeitsanalyse auf ein viel einfacher zu realisierendes Verfahren zurückgegriffen, welches dem bereits vorgestellten angepassten Ray Tracing ähnlich ist. Dabei wird die Szene nicht wie beim Ray Tracing mit einem Blickstrahl abgetastet, sondern mit einem Blickfenster. Der komplette Scanbereich, etwa der eines Pixels, wird mit Hilfe eines kleinen Fensters sequenziell abgescannt.

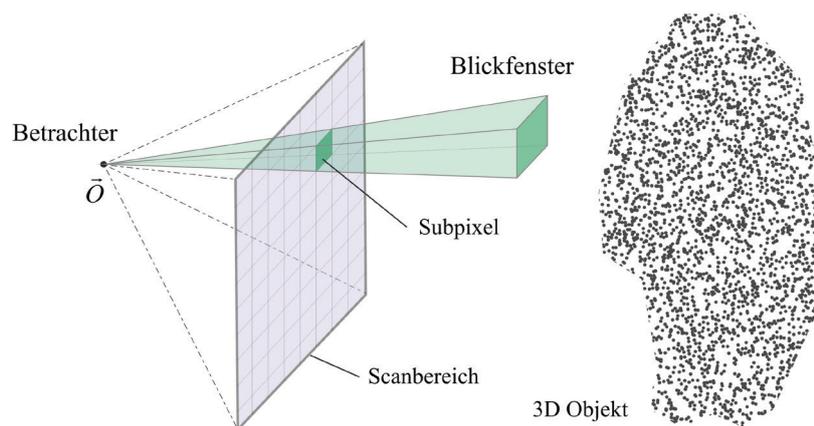


Abbildung 6.8: Das Prinzip des Renderings mit einem Blickfenster

Abbildung 6.8 zeigt das Prinzip des Scanvorgangs. Der Sichtbereich eines Pixels wird in viele kleinere Teilbereiche unterteilt, welche mit Hilfe von Subpixeln realisiert werden. Anschließend wird Punkt für Punkt daraufhin untersucht, ob er in dem durch das Subpixel vorgegebenen Blickfenster liegt. Wird ein solcher Punkt gefunden, dann wird der Abstand zu dem Punkt in den lokalen *Z-Buffer* geschrieben und die Koordinaten des Punktes in den lokalen *Pointbuffer*. Sollte ein weiterer Punkt gefunden werden, dessen Abstand zum Betrachter hin kleiner ist als der gespeicherte Wert im *Z-Buffer*, dann wird der neue Wert in den *Z-Buffer* geschrieben und die neuen Koordinaten in den *Pointbuffer*. Auf diese Weise stehen nach einem kompletten Scandurchgang in jeder Zelle des *Pointbuffers* die Koordinaten von maximal einem Punkt, und zwar dem mit

dem kürzesten Abstand zum Betrachter. Dieser Punkt muss somit zwangsläufig am nächsten zum Betrachter und ein Teil der Front eines Objektes sein.

Dieser einfache Ansatz hat allerdings auch einige Nachteile. Infolge des Z-Buffer-basierten Algorithmus findet eine Datenreduktion statt. Bei der Reduzierung aller Punkte im Blickfenster auf einen einzigen Punkt mit dem kürzesten Abstand werden nicht nur die potenziellen Bereiche der Rückseite und verdeckter Objekte eliminiert, sondern evtl. auch einige Teile der sichtbaren Daten. Besonders starke Reduktion findet dann statt, wenn der Öffnungswinkel des Blickfensters sehr viele Punkte einschließt. Wird der Winkel dagegen sehr klein gehalten, dann kann es unter Umständen dazu kommen, dass man zwischen den Punkten an der Front des Objektes hindurchschaut. In Abbildung 6.9 wird das Problem anhand einer 2D Projektion illustriert. Das größere Blickfenster (grün dargestellt) zeigt einen für diese Szene optimal gewählten Scanstrahl – die Datenreduktion im relevanten Bereich (rote Punkte) wäre nicht sehr groß, die Rückseite würde dagegen komplett herausgeschnitten. Mit dem kleineren Blickfenster kann dagegen durch die roten Punkte hindurchgeschaut werden und die Rückseite würde mit dekletiert.

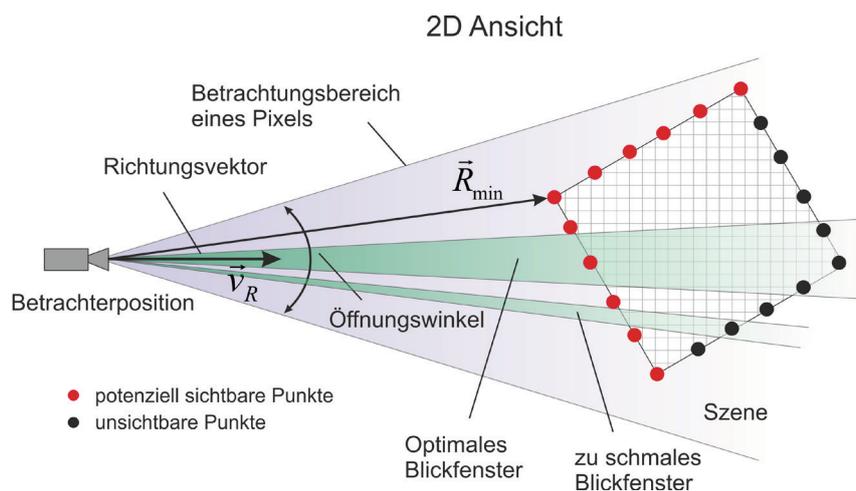


Abbildung 6.9: Das Prinzip des Renderings mit einem Blickfenster

Das Beispiel zeigt sehr deutlich, dass ein geeignet gewählter Öffnungswinkel des Blickfensters maßgebend für die Qualität des Rendering ist. Er kann nicht konstant, sondern muss frameweise adaptiv gewählt werden. Bei Objekten mit einem halbwegs festen Abstand zwischen den räumlich benachbarten Punkten, lässt sich mit der Kenntnis der Abstände zwischen benachbarten Punkten und des kleinsten Abstandes zum Punktziel hin ein optimierter Öffnungswinkel vorgeben. Um den mittleren Abstand zwischen den benachbarten Punkten abzuschätzen, werden im Datensatz einige Punkte (in der Regel reichen 10-20 Punkte) herausgegriffen und der minimale Abstand zu allen anderen Punkten bestimmt. Der gemittelte minimale Abstand wird als mittlerer Abstand zwischen benachbarten Punkten angenommen. Bei Objekten, die ihre Form im Laufe der

Simulation nicht ändern, sondern nur ihre Position, braucht dieser Schritt nur einmal gemacht zu werden.

Zur Reduktion des Scanbereichs kann auch beim Rendern einer Punktwolke auf die im vorherigen Abschnitt beschriebene Methode mit einer Richtungstabelle für alle Punkte zurückgegriffen werden. Vor dem eigentlichen Rendern werden die Punkte nach ihrer Richtung in einer Tabelle sortiert, im gleichen Schritt kann auch der minimale Abstand \bar{R}_{\min} (siehe Abbildung) bestimmt werden. So kann ohne großen zusätzlichen Rechenaufwand ein optimierter Öffnungswinkel für eine Szene quasi automatisch vorgeben werden. Dies funktioniert jedoch nur bei Daten mit relativ festen Abständen der benachbarten Punkte zueinander.

6.5 Benchmark Test

Mit dem folgenden Benchmark Test soll eine Vorstellung für die Größenordnung der für das Rendern benötigten Zeit vermittelt werden. Es werden insgesamt 5 Methoden für zwei verschiedene Objekte und drei verschiedene Objektüberdeckungen gegenübergestellt.

Simulationsszenario: 2 Polygonnetze bzw. Punktwolken mit je ca. 5000 Polygonen bzw. Punkten, beide im Sichtbereich, Sensorauflösung 160x120 Pixel

System: Intel Core i5-650- Processor (4MB Cache, 3.2GHz), Anzahl der CPU-Kerne – 2, 4GB RAM, Schreibdurchsatz der Festplatte 35-52 MB/s.

<i>Objektüberdeckung in %</i>	<i>Backface Calling + Ray Casting</i>	<i>Backface Calling + angepasstes Ray Casting</i>	<i>Subpixel-Methode ohne Neigung des Sensors</i>	<i>Subpixel-Methode mit Neigung des Sensors</i>	<i>Subpixel-Methode mit Neigung des Sensors nach Abschnitt 12.1</i>
0	5 Min 20 Sek	64 Sek	42 Sek	44 Sek	8 Min 22 Sek
20	6 Min 04 Sek	74 Sek	46 Sek	50 Sek	9 Min 37 Sek
50	6 Min 34 Sek	80 Sek	50 Sek	54 Sek	10 Min 24 Sek

Tabelle 6.1: Vergleich der benötigten Rechenzeiten bei unterschiedlichen Sichtbarkeitsverfahren

Die in der Tabelle dargestellten Rechenzeiten beziehen sich nur auf das reine Rendern, die Simulation der Amplituden- und Entfernungsbilder nimmt durchschnittlich noch ein paar Sekunden in Anspruch. Bei Simulationen mit Zwischenframes steigt die Rechenzeit näherungsweise um den Faktor F (Anzahl der Zwischenframes) an, da die

Berechnungen in den einzelnen Zwischenframes an sich den Berechnungen ohne diese ($F=1$) entsprechen. Dabei ist anzumerken, dass die Sichtbarkeitsanalyse im Simulator auf Wunsch komplett umgangen werden kann, wenn die Daten bereits voranalysiert vorliegen. Dies macht Sinn bei wiederholten Simulationen eines und des gleichen Szenarios mit unterschiedlichen Kameraparametern, z.B. bei Fehleranalysen durch unterschiedliches Rauschen, Quantisierung und/oder teilweise die Veränderung der lichtempfindlichen Pixelbereiche der Kamera.

7 Linsenverzeichnung

In beinahe allen optischen Systemen tritt der Effekt der Linsenverzeichnung auf. Je nach verwendeter Optik, ihrer Dimension, Lichtstärke, Öffnungswinkel, Abbildungsmaßstab usw. ist dieser Effekt mehr oder weniger ausgeprägt. Besonders bei Weitwinkelobjektiven kann eine Linsenverzeichnung sehr deutlich beobachtet werden. Der Einfluss der optischen Verzerrung auf die Messunsicherheit wurde u.a. in [Misc 07] behandelt. Für die Modellierung von Optikverzerrungen werden die klassischen Linsenverzerrungen betrachtet.

7.1 Klassische Linsenverzeichnungen

Eine der häufigsten und vor allem schwerwiegendsten Linsenverzeichnungen, besonders bei günstigen und einfach aufgebauten Optiksyste men, ist die radiale Verzeichnung. Dabei wird der Abstand des Bildpunktes zum Zentrum der Verzerrung radial skaliert. Sie lässt sich mathematisch durch ein Produkt mit einem Polynom beschreiben [Atk 96]:

$$\begin{aligned}x_d &= (1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \cdot x = x + \delta_{r,xd}(x, y) \\y_d &= (1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \cdot y = y + \delta_{r,yd}(x, y) \\ \text{mit } r^2 &= (x - x_c)^2 + (y - y_c)^2\end{aligned}\tag{7.1}$$

(x, y) – Koordinaten des Punktes im unverzerrten Bild

(x_d, y_d) – Koordinaten des Punktes im verzerrten Bild

(x_c, y_c) – Koordinaten des Verzerrungszentrums (liegt in der Regel auf der optischen Achse)

$\delta_{r,xd}(x, y)$, $\delta_{r,yd}(x, y)$ – die jeweiligen Verschiebungen der Koordinaten in x- und y-Richtung

k_n – der n-te radiale Verzeichnungskoeffizient

Positive k_1 -Werte bewirken die sogenannte Kissenverzerrung (die Krümmung verläuft zur Bildmitte), negative Werte bewirken dagegen eine Tonnenverzerrung (die Krümmung verläuft von der Bildmitte weg). Wie man aus der Formel erkennen kann, werden Punkte, die gleichen Abstand zum Verzeichnungszentrum haben, gleich stark verzeichnet.

Im Gegensatz zu Simulationen geht es in der Praxis oft nicht darum, ein Bild zu verzerren, sondern das aufgenommene Bild zu entzerren. Da der Weg des Lichtstrahles um-

kehrbar ist, lässt sich dementsprechend jede Verzerrung in der geometrischen Optik umkehren. Die radialen Verzeichnungskoeffizienten sind linsenspezifisch und lassen sich in der Regel durch ein geeignetes Kamerakalibrierungsverfahren bestimmen, siehe z.B. [Graf 07]. Dabei werden die unendlichen Potenzreihen meistens durch ein Polynom 2. (biquadratische Modelle) oder 3. (bikubische Modelle) Grades approximiert. Jede weitere Potenz bringt eine enorme Steigerung des Komplexitätsumfanges mit sich. Simulationstechnisch macht es also bei Verzerrungen wenig Sinn, Polynome höherer Ordnung als drei zu benutzen. Objektivverzeichnungen höherer Grade können vielleicht bei Sensoren sehr hoher Auflösung noch ausreichend genau bestimmt werden, bei typischen Auflösungen aktueller PMD-Kameras ist es durch die relativ grobe Rasterung der Pixelmatrix nicht mehr gegeben.

Weitere Optikverzeichnungen

Eine neben der radialen Verzeichnung häufig auftretende Verzerrung ist die tangentiale Verzeichnung eines Optiksystems. Idealerweise sollten alle Elemente eines Optiksystems kollinear mit der optischen Achse ausgerichtet sein. Bei der Fertigung kommt es aber aufgrund zulässiger Toleranzen zu kleinen Verschiebungen und Verdrehungen der einzelnen Linsen. Die daraus resultierende optische Verzeichnung der Bilder wird als tangentielle Verzeichnung bezeichnet. Für das tangential verzerrte Bild folgt nach [Wang 08]:

$$\begin{aligned} x_d &= x + \delta_{t,xd}(x, y) \\ y_d &= y + \delta_{t,yd}(x, y) \end{aligned} \tag{7.2}$$

mit

$$\begin{aligned} \delta_{t,xd}(x, y) &= p_1(3x^2 + y^2) + 2p_2 \cdot x \cdot y \\ \delta_{t,yd}(x, y) &= p_2(3x^2 + y^2) + 2p_1 \cdot x \cdot y \end{aligned}$$

p_1 und p_2 sind die tangentialen Verzeichnungsparameter.

Eine Erweiterung der radialen Verzeichnung stellt die elliptische Verzeichnung dar, mit der Deformationen auf der Linse und Verkippungen der Linse gegenüber dem Sensorarray modelliert werden können [Graf 07].

$$\begin{aligned} x_d &= x + \delta_{e,xd}(x, y) \\ y_d &= y + \delta_{e,yd}(x, y) \end{aligned} \tag{7.3}$$

mit

$$\delta_{e,xd}(x) = x \cdot \sum_{i=1}^{\infty} s_{i,x}(x^2 + y^2) \quad \text{und} \quad \delta_{e,yd}(y) = y \cdot \sum_{i=1}^{\infty} s_{i,y}(x^2 + y^2)$$

$s_{i,x}$ und $s_{i,y}$ sind die elliptischen Verzeichnungsparameter.

7.2 Modellierung der Linsenverzeichnungen

Berücksichtigt man die Tatsache, dass die Sensorebene im Simulator gegenüber der optischen Achse vollständig gekippt und verschoben werden kann, so können tangentiale Verzeichnungen bereits damit vollständig simuliert werden. Eine weitere Realisierung dieser ist somit unnötig. Auf die Simulation der elliptischen Verzeichnungen wird im Rahmen dieser Arbeit nicht eingegangen. Da solche Verzerrungen allgemein nicht-symmetrischer Natur sind, sind die elliptischen Verzeichnungsparameter schwer zu bestimmen und liegen für die vorhandene Optiken nicht vor. Sie können allerdings wegen der einfachen additionalen Realisierbarkeit dieser Verzerrungsart jederzeit hinzugefügt werden.

Die meisten Kalibrierungs- und Entzerrungsmethoden arbeiten in der Bildebene. Dabei wird eine vordefinierte Szene aufgenommen, z.B. ein Schachbrett mit bekanntem Ausmaß (s. [Zha 99], [Zha 00]), und nachträglich im Bild anhand der zu erwartenden Geometrie und der bekannten geometrischen Proportionen die intrinsischen Parameter der Optik bestimmt. Nach der Bestimmung der Entzerrungsvorschrift kann jedes Pixel des verzerrten Bildes entsprechend der Vorschrift auf das entzerrte Bild abgebildet werden. Dabei wird es je nach verwendetem Algorithmus entweder einfach verschoben, oder es wird zusätzlich unter Ausnutzung der Nachbarschaftsbeziehungen in seinen Eigenschaften beeinflusst. Denn die Koordinaten eines transformierten Pixels (x, y) werden in der Regel nie genau mit der Rasterung eines entzerrten Bildes übereinstimmen. Es muss immer eine „quantisierte“ Entscheidung getroffen werden.

Betrachtet man den Prozess der Simulation eines Pixels im PMD-Simulator, so macht es wenig Sinn, die Verzeichnungen der Linse nachträglich pixelweise auf das gewonnene Bild anzuwenden, wenn man über die Kenntnis der optischen Abbildung jedes einzelnen Punktes der zu simulierenden Szene auf die Pixelfläche verfügt. Viel sinnvoller ist es, die Ablenkung des Strahles durch die Optik bereits nach der Bestimmung der verzerrungsfreien Abbildung eines Punktes bzw. aller Punkte auf die Sensorebene vorzunehmen. In Abbildung 7.1 ist das Prinzip der Modellierung der Verzeichnungen des

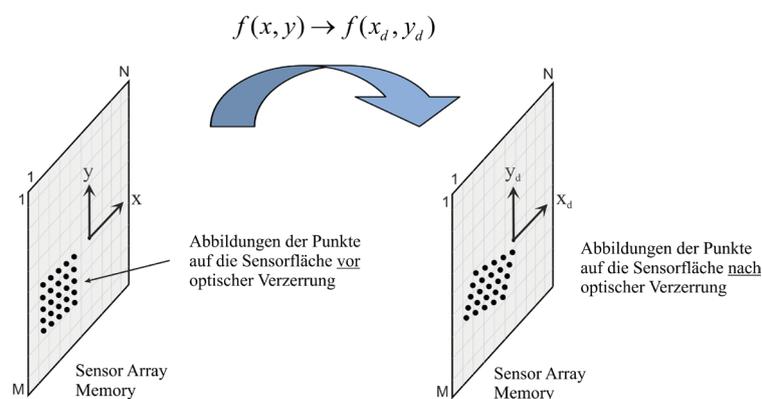


Abbildung 7.1: Das Prinzip der Modellierung der radialen Verzerrung

eingesetzten Linsensystems (gilt nicht nur für radiale) vorgestellt. Nach der Berechnung der Abbildung der sichtbaren Bereiche auf die Sensoroberfläche werden die Koordinaten der Projektionen der Punkte im Sensorkoordinatensystem entsprechend der Vorschrift (7.1) verschoben. Anschließend findet eine Zuordnung der Punkte den entsprechenden Pixel statt.

Abbildung 7.2 zeigt ein Beispiel für eine Kissenverzerrung bei $k_1 = 0,1$, $k_2, \dots = 0$. Links im Bild ist ein Punktgitter dargestellt, welches rechts im Bild verzerrt wurde. Die Differenzvektoren – die Vektoren von den richtigen Positionen zu verzerrten Positionen hin – sind im Bild blau dargestellt.

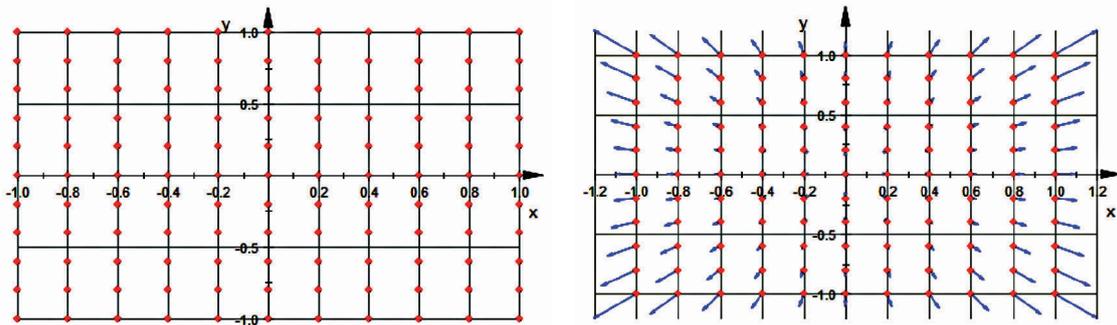


Abbildung 7.2: Veranschaulichung der Verzerrung vektoriell: links – ein unverzerrtes Punktfeld, rechts – Kissenverzerrung als Differenzvektorfeld ($k_1 = 0,1$)

Abbildung 7.3 zeigt das Ergebnis der optischen Verzerrung eines kompletten Gitters.

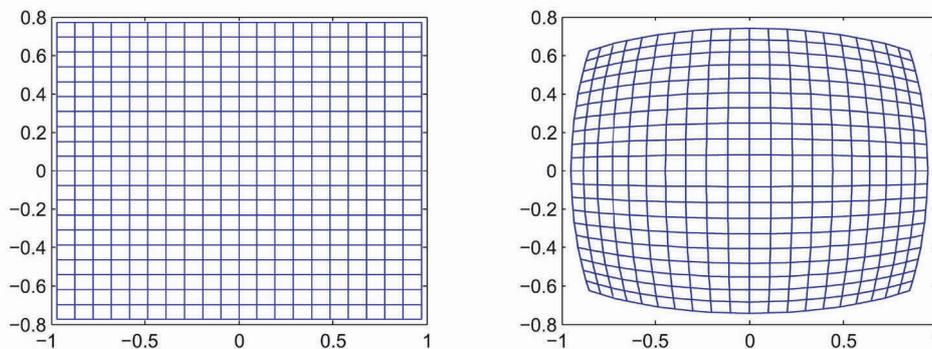


Abbildung 7.3: Das Prinzip der radialen Verzerrung in Flächendarstellung: links – das unverzerrte, rechts – Beispiel einer Tonnenverzerrung

8 Simulatorarchitektur

Im nachfolgenden Kapitel wird die im Rahmen dieser Arbeit entwickelte Simulatorarchitektur vorgestellt. Das Zusammenspiel der behandelten Modellierungsansätze, basierend auf den grundlegenden Anforderungen an den PMD-Simulator aus Kapitel 1.2, wird anhand der wichtigsten Teilkomponenten, ihrer Aufgaben, Funktionen und deren Interaktion untereinander nachfolgend ausführlicher betrachtet. Die Wahl einer bestimmten Architektur ist entscheidend für die nichtfunktionalen, qualitativen Eigenschaften des resultierenden Systems. Schon vor der eigentlichen Entwicklung der Software bestimmt sie über den internen Aufbau, Effizienz und die eventuelle Erweiterbarkeit des künftigen Konzeptes. Bei der Entwicklung des PMD-Simulators wurde großer Wert darauf gelegt, die einzelnen Teilberechnungen in autonomen Komponenten auszuführen. Die sich im Laufe der Simulation häufig wiederholenden Schritte und Abläufe lassen sich so besser kontrollieren, die Größe des benötigten Programm-Codes sinkt gleichzeitig.

8.1 Frame Control Unit (FCU)

Wie bereits aus der Aufgabenstellung des Simulators ersichtlich ist, finden aufgrund der umfangreichen Interaktionsberechnungen beim Zusammenwirken der einzelnen Simulationskomponenten, wie Lichtquellen, Punktzielen und PMD-Pixel im Zusammenspiel mit der zeitlichen Varianz, umfangreiche und aufwendige Berechnungen statt. Um die ereignis- und zeitgesteuerten Abläufe während einer Simulation kontrollieren zu können, wurde die Ablaufsteuerungs- und Kontrollkomponente implementiert – die *Frame Control Unit*. Als zentrales Bindeglied verbindet sie die einzelnen Teilkomponenten und sorgt für einen reibungslosen Ablauf der Simulation. Sie beinhaltet die methodische und zeitliche Ablaufsteuerung der mitwirkenden Komponenten, die ihrerseits jeweils eine spezifische Funktionalität bereitstellen. So kann jede dieser Funktionalitäten sequenziell und unabhängig von anderen aufgerufen und abgearbeitet werden, ohne dass die internen Verschachtelungen übermäßig kompliziert, undurchschaubar und ineffizient werden. Neben der zeitlichen Steuerung stellt und überwacht sie die Schnittstellen zwischen den einzelnen Modulen. Kein Modul interagiert mit einem anderen außerhalb der *Frame Control Unit*. Sie stellt somit stets eine zentrale „Logik“-Komponente dar. Alle weiteren Komponenten sind darauf abgestimmt, anhand der durchgegebenen Eingangsinformationen eine bestimmte Funktion auszuführen und die Ergebnisse zurückzuliefern. Sie stellen programtechnisch Klassen dar, wobei sie selbst aus weiteren Unterklassen bestehen können. Bis auf das *Light Propagation Module* sind alle Module nach außen hin „geschlossen“ und können als „Blackbox“ mit Ein- und Ausgang betrachtet werden.

Eine weitere sehr wichtige Funktionalität der *Frame Control Unit* ist das Daten- und Speichermanagement. Die Simulationsszenarios, die Bewegungsdateien, die Parameterdateien, die Zwischen- und die Endergebnisse werden von ihr verwaltet und für einzelne Berechnungsschritte vorbereitet.

Abbildung 8.1 zeigt das Zusammenwirken der *Frame Control Unit* mit weiteren Simulator-Komponenten.

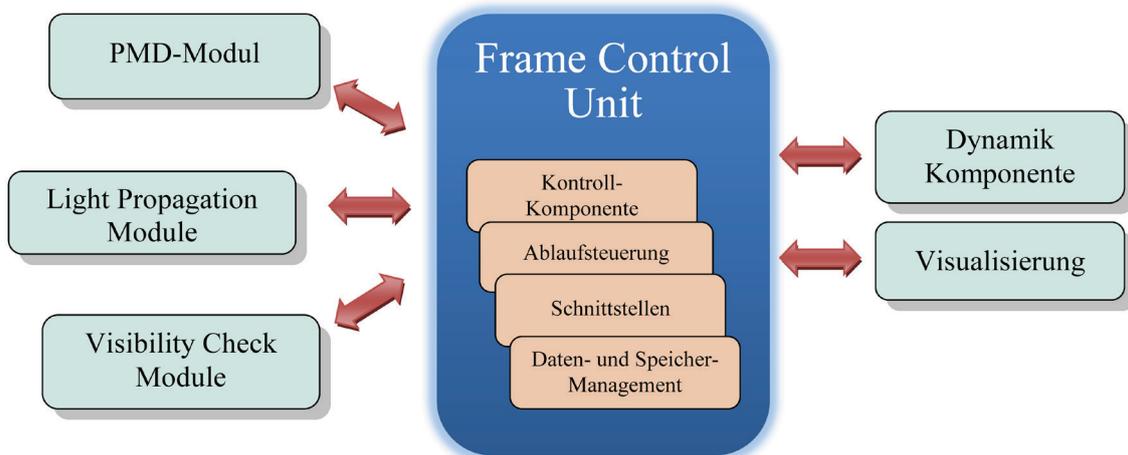


Abbildung 8.1: Aufgaben- und Kommunikationsbereiche der Frame Control Unit

8.2 Visibility Check Module

Das *Visibility Check Module* setzt die in Kapitel 6 behandelten Ansätze um. Als Eingangsdaten dienen die von der *Frame Control Unit* aufbereiteten Objektdateien (Informationen der 3D Szene), Position und Blickfenster des Betrachters und die gewünschte/erforderliche Methode (Backface Culling + Ray Casting, Backface Culling + angepasstes Ray Casting, Subpixel Methode oder alternative Subpixel Methode nach Ab-

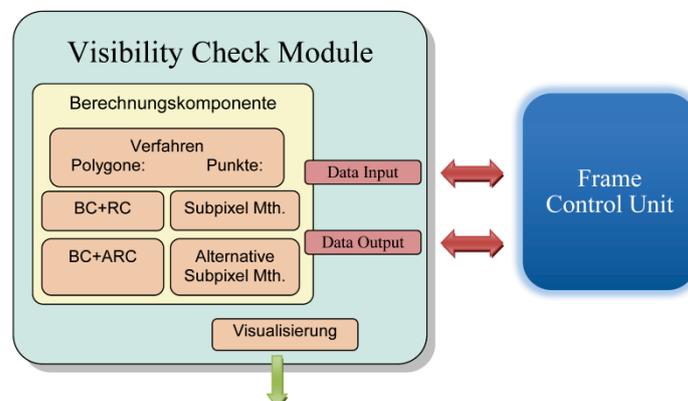


Abbildung 8.2: Interaktion zwischen dem Visibility Check Module und der Frame Control Unit

schnitt 12.1). Im Falle der Berechnung der Detektierbarkeit seitens eines Sensors findet während der gesamten Berechnung zusätzlich noch eine kontinuierliche Interaktion mit der FCU zwecks Aktualisierung der momentanen Blickstrahlrichtung statt. Ausgangsseitig stellt das VCM die separierten sichtbaren Bereiche der Szene zur Verfügung. Optional können die berechneten Daten grafisch dargestellt werden. Abbildung 8.2 veranschaulicht die Interaktion zwischen der *Frame Control Unit* und dem *Visibility Check Module*.

8.3 Light Propagation Module

Das *Light Propagation Module* vereint zwei Funktionalitäten. Die eine ist der in Kapitel 3.3 bereits erwähnte Independent Mode (unabhängiger Betrieb). Bei dieser Betriebsart kann das LPM unabhängig vom PMD-Simulator ausgeführt werden und erlaubt Berechnungen der Lichtverhältnisse in einer vordefinierten Szene sowie die Darstellung der Rückstreuung in eine vordefinierte Richtung. Diese Funktionalität kann sehr nützlich sein bei Bestimmung und theoretischer Auslegung neuer Beleuchter, die aus mehreren Beleuchtern bestehen können und bzw. oder das Licht gebündelt in eine bestimmte Richtung abgeben sollen, wodurch durch eine gestiegene höhere örtliche Nutzlichtintensität größere Messbereiche überbrückt werden können. Eine umfassende Untersuchung verschiedener Beleuchter-Konfigurationen wurde im Zentrum für Sensorsysteme durchgeführt und kann in [Lott 10] nachgeschlagen werden. Gemeinsame Erkenntnisse, sowie ein Vergleich der Simulationsergebnisse mit realen Messwerten wurden auf dem „7th EUROSIM Congress on Modelling and Simulation“ 2010 in Prag vorgestellt und publiziert, siehe [Pet 10a], [Pet 10b].

Abbildung 8.3 stellt das Zusammenwirken zwischen der *Frame Control Unit* und dem *Light Propagation Module* dar.

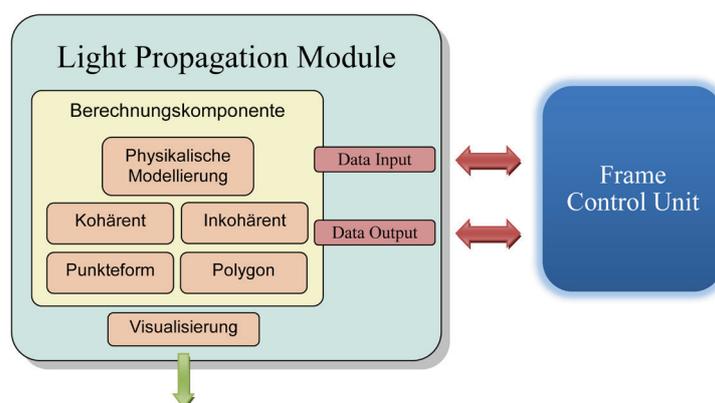


Abbildung 8.3: Interaktion zwischen dem Light Propagation Module und der Frame Control Unit

Die zweite Betriebsart des *Light Propagation Module* ist die Berechnung der für die Simulation der theoretischen Antwort des Sensors notwendigen Lichtintensitätswerte. In

einer Interaktion über die FCU mit dem VCM werden für die sichtbaren Bereiche der Szene die Rückstrahlintensitäten in eine vorgegebene Richtung berechnet. Eingangsseitig werden in das Modul die Objektdateien, Betriebsart der Lichtquellen und ihre Parameter (Modulationsfrequenz, Abstrahlcharakteristik, Position und Lage) eingespeist. Anhand dieser Daten und Vorgaben werden entsprechend der Art des vorliegenden Objektes (polygonale Form oder Punkteform) die Lichtintensitäten in den einzelnen Punktzielen berechnet und im bistatischen Fall gemäß der Blickrichtung zum Betrachter hin (z.B. Sensor) und der Rückstreuungsfunktion korrigiert.

Die Lichtintensitäten in den Punktzielen oder die Intensität der von der Oberfläche zurückgestrahlten Infrarotstrahlung in eine vorgegebene Richtung kann visuell dargestellt werden.

8.4 PMD-Modul

Die Hauptaufgabe des PMD-Moduls liegt in der Berechnung der synthetischen Tiefen- und Amplitudenbilder. Von der *Frame Control Unit* werden die Daten für die Berechnungen aufbereitet. Die Szenenbereiche, die beleuchtet wurden und potentiell detektierbar sind, liegen mit den entsprechenden Lichtintensitätskenntnissen eingangsseitig vor. Im PMD-Modul werden einige aufeinander folgende Schritte durchgeführt. Von der FCU wird ein Kamera-Objekt übergeben, dieses beinhaltet eine Modellierung der zu simulierenden Kamera mit allen während der Initialisierungsphase (siehe Kapitel 8.7) festgelegten Parametern. Die Szenenobjekte liegen bereits im kamerafesten Koordinatensystem vor. Die Punktziele werden entsprechend der optischen Abbildung auf den Sensor projiziert und den entsprechenden Pixeln zugeordnet. Anschließend werden je nach Betriebsart *Frame* oder *Zwischenframe*, die von der FCU für jede Berechnung durchgegeben wird, entsprechend die Tiefen- und Amplitudenbilder oder die Pixelmatrizen mit Integrationsparametern (siehe Kapitel 5.8) berechnet und an die FCU wieder zurückgegeben. Die Werte dieser Pixelmatrizen werden so lange im Zwischenspeicher gehalten, bis von der FCU ein *Frame* initialisiert wird. Nur zu diesen Simulationszeiten wird die endgültige theoretische Antwort des Sensors berechnet. Ist die Funktion der Simulation der Bewegungsunschärfe deaktiviert, dann entspricht jedes *Zwischenframe* einem *Frame*. Die Tiefen- und Amplitudenbilder werden dann bei jeder Ansteuerung des PMD-Moduls generiert.

Abbildung 8.4 veranschaulicht das Zusammenwirken zwischen der *Frame Control Unit* und dem *PMD Modul*.

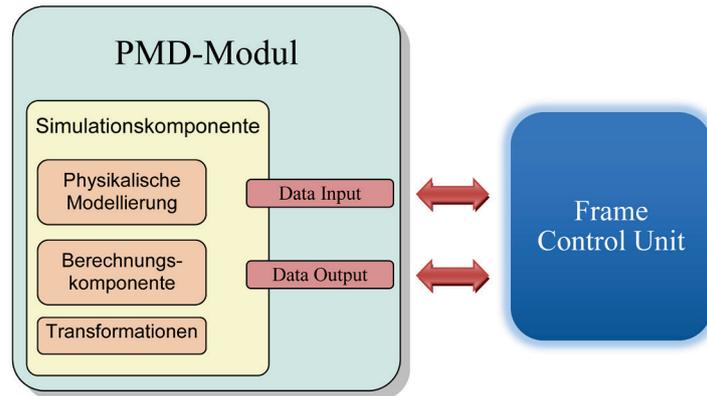


Abbildung 8.4: Interaktion zwischen dem PMD-Modul und der Frame Control Unit

8.5 Visualisierung

Die Visualisierungskomponente dient zur Verwaltung und Ausführung der gewünschten Funktionalitäten der optischen Darstellung und Eingabehilfen. So kann die Parametrisierung des Simulators wahlweise entweder mit Hilfe einer Initialisierungsdatei oder mittels einer grafischen Oberfläche erfolgen. Mit Hilfe einer Checkbox und einer Auswahlleiste können außerdem einige während der Simulation generierten Zwischenergebnisse dargestellt werden.

Abbildung 8.5 veranschaulicht das Zusammenwirken zwischen der *Frame Control Unit* und dem Visualisierungsmodul.

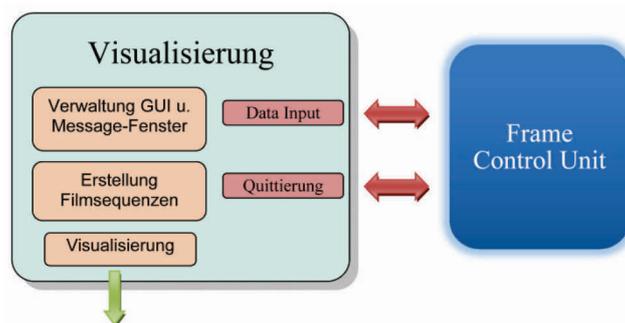


Abbildung 8.5: Interaktion zwischen der Visualisierungskomponente und der Frame Control Unit

8.6 Dynamikkomponente

Die Dynamikkomponente ist im softwaretechnischen Sinne nicht komplett von der *Frame Control Unit* losgelöst. Ein Teil davon ist in ihr integriert. Dieser Teil ist verantwortlich für die Bereitstellung der aktuellen Position und Orientierung der Simulati-

onsteilnehmer, welche sich aus den Trajektoriendateien ergeben. Anhand der aktuellen Position, Lage und des Zeitpunktes wird die nachfolgende Position und Lage berechnet. Diese Daten dienen später als Eingangsdaten bei oben beschriebenen Interaktionen.

Der zweite Teil der Dynamikkomponente ist autonom ausgeführt und ist zuständig für die im Laufe der Simulation durchzuführenden Koordinatentransformationen.

8.7 Simulationsablauf

Die in den vorangegangenen Abschnitten behandelten Komponenten und deren Zusammenspiel im Simulator sollen nun anhand eines typischen Simulationsvorgangs näher betrachtet werden. Der Simulationsprozess lässt sich in fünf Hauptschritte unterteilen, die in Abbildung 8.6 dargestellt sind.

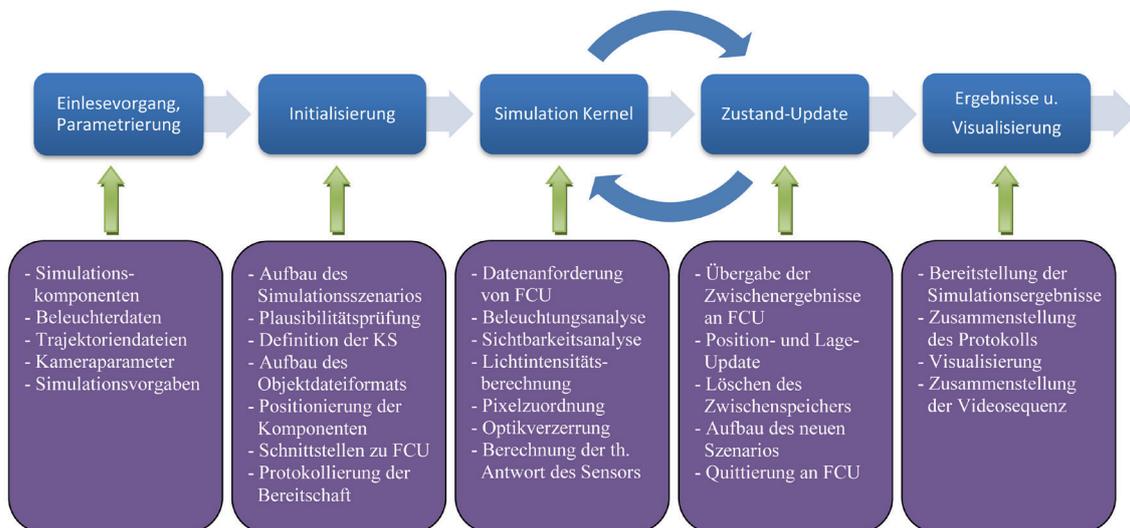


Abbildung 8.6: Simulationsablauf und die wichtigsten Arbeitsschritte

8.7.1 Einlesevorgang und Parametrierung

Die Simulation beginnt mit dem Aufbau des Simulationsszenarios. Dafür werden die vordefinierten Teilnehmer eines Simulationsprozesses aus den entsprechenden Objektdateien geladen. Für häufig verwendete Simulationsanordnungen liegen diese Daten im Dateiarchiv bereits vor und können von da aus direkt eingebunden werden. Sollen neue Komponenten simuliert werden (neue Beleuchter, Objektumgebung usw.), dann müssen vor der Simulation die entsprechenden Dateien erstellt werden. Dabei ist zu beachten, dass die Abarbeitung der einzelnen Beleuchter und Objekte sequentiell durchgeführt wird. Die Reihenfolge der eingelesenen Daten bestimmt somit die Reihenfolge des späteren Ablaufs der Berechnungen – Überlagerung der Objekte im 3D Raum oder Hinzufügen neuer Beleuchter in die Simulationsumgebung geschieht in der gleichen Reihen-

folge. Werden im Ordner-Set Trajektoriendateien vorgefunden, dann werden diese an die entsprechenden Objekte „angehängt“, wodurch die Objekte später bei Simulation anhand der Bewegungsvorgaben positioniert und orientiert werden. Anderenfalls werden die Objekte als starr betrachtet. Eine Kombination der beiden Möglichkeiten ist ebenfalls gegeben.

Im gleichen Arbeitsschritt wird die PMD-Kamera definiert. Die Parameter der Kamera können wahlweise aus einer Kamera-Datei eingelesen, oder in der integrierten GUI (Graphical User Interface) eingegeben werden.

8.7.2 Initialisierung

Nachdem im ersten Schritt die Simulationskomponenten und -parameter definiert wurden, wird der Initialisierungsvorgang ausgelöst. Während dieser relativ kurzen Phase wird das Ausgangssimulationsszenario erstellt. Nach einer kurzen Plausibilitätsprüfung werden die wichtigsten Koordinatensysteme definiert und die einzelnen Komponenten entsprechend der Vorgaben an ihre Startpositionen versetzt und ausgerichtet. Die Zielobjekte werden in das programminterne Objektdateiformat konvertiert und zusammengefasst. Das in Kapitel 5.1 behandelte Format ermöglicht einerseits das zeit- und speicheroptimierte Management großer Datenmengen und andererseits das Zusammentragen der wichtigsten Parameter der Objektumgebung wie ihre Bestandteile, die Rückstreuungseigenschaften der einzelnen Objekte, ihre Bewegungsabläufe und die objektinternen Knotenbeziehungen zueinander.

Nach einem erfolgreichen Initialisierungsprozess werden die für die Simulation aufbereiteten Daten an die *Frame Control Unit* übergeben sowie weitere Schnittstellen zwischen der FCU und den einzelnen Simulationsmodulen realisiert.

8.7.3 Simulation

Nach einer erfolgreichen Quittierung des abgeschlossenen Initialisierungsvorgangs kann die eigentliche Simulation beginnen. Der gesamte Simulationsprozess wird ab nun von der *Frame Control Unit* gesteuert. Es werden der aktuelle Momentzustand der Objekt-daten sowie die Position des Sensors an das *Visibility Check Module* übergeben und eine Sichtbarkeitsanalyse veranlasst. Die von VCM berechneten sichtbaren (potenziell detektierbaren) Punktziele werden von FCU in die *Erkenntnisebene* übertragen. Anschließend wird überprüft, welche der berechneten detektierbaren Punktziele von der jeweiligen Lichtquelle aus beleuchtet werden können. Die FCU steuert mit aktualisierten Positionsdaten wiederholt das VCM an. Im gleichen Zug wird von dem *Light Propagation Module* die Lichtintensität im Punktziel bestimmt und entsprechend der relativen Position des Sensors korrigiert (Rückstrahlung in Richtung Sensor ist wichtig). Die Interaktion VCM \Leftrightarrow LPM findet für jede Lichtquelle der Simulationsszene statt.

Nach der Berechnung der von dem Sensor detektierbaren Punktziele mit ihren entsprechenden integrierten Lichtintensitäten wird das *PMD-Modul* dazu veranlasst, die theoretische Antwort des Sensors zu bestimmen. Entsprechend der optischen Abbildung werden die Punktziele auf die einzelnen Pixel abgebildet. Anschließend werden die Tiefen- und die Amplitudenbilder berechnet und an die *Frame Control Unit* übergeben. Dies geschieht jedoch nur zu den Ablaufzeitpunkten der Frame-Berechnung. Wird gerade ein Zwischenframe berechnet, dann werden die gewonnenen Daten in einen *Stack* – eine Art Stapelspeicher zwischen zwei Frames – übertragen und so lange dort „gesammelt“ bis ein Frame berechnet werden kann.

8.7.4 Zustand-Update

Nach der Berechnung eines Frames bzw. Zwischenframes werden die entsprechenden Daten für Positionen und Lagewinkel der Simulationsteilnehmer aktualisiert sowie sämtliche nicht mehr benötigten Informationen zurückgesetzt. Das Simulationsszenario wird für das nächste Frame/Zwischenframe neu aufgebaut. Der Zustand nach diesem Update entspricht mit der Ausnahme des *Stacks* bei Zwischenframes, der Zeitvariablen, der Frame-Historie (enthält die berechneten Tiefen- und Amplitudensequenzen) und einiger wenigen weiteren globalen Variablen dem Ausgangszustand nach der Initialisierung. Alle weiteren Datensätze und Zwischenergebnisse werden mit dem Ziel der Minimierung des Datenflusses und des erforderlichen Speicherbedarfes gelöscht, wobei eine komplette Protokollierung nach Wunsch zugeschaltet werden kann. Der Datenbestand steigt jedoch bei einer solchen Protokollierung mit jedem Zwischenframe linear an – nach 100 Zwischenframes entsteht so näherungsweise die 100-fache Datenmenge.

Nach dem Update des Szenarios erfolgt eine Quittierung an die *Frame Control Unit*, das nächste Frame/Zwischenframe kann berechnet werden.

8.7.5 Ergebnisse und Visualisierung

Nach Beendigung der gesamten Simulation werden die Ergebnisse in die entsprechenden Simulationsdateien geschrieben. Sie enthalten den zeitlichen Ablauf der synthetischen Sensordaten. Optional dazu können Filmsequenzen generiert werden, die mit einer vordefinierten Bildwiederholungsrate abgespielt werden können. An dieser Stelle muss jedoch erwähnt werden, dass die zumindest in der Matlab Version 7.11 integrierten Codecs keine hochqualitative und gleichermaßen flexible Konvertierung der einzelnen Frames zu einem Video zulassen. Um dem Benutzer die Möglichkeit zu bieten, das Video in einem gewünschten Format selber anzufertigen, werden die einzelnen Frames zusätzlich zu dem Video im verlustfreien TIFF-Format mit abgespeichert.

9 Beispielszenarien

Nach der Behandlung der theoretischen Aspekte der Modellierung und Simulation sollen im nachfolgenden Kapitel ein paar Simulationsszenarien betrachtet sowie Vergleiche mit realen Messdaten angestellt werden. Bei vielen dieser Simulationen handelt es sich um relativ einfache Szenarien. Das hat zwei Hintergründe. Erstens sind solche Abläufe auch in der Realität einfacher nachzubilden und zu wiederholen, und zweitens kann eine sehr komplizierte 3D Umgebung mit vielen Details nur mit einem enormen Aufwand virtualisiert werden. Prinzipiell spricht jedoch nichts auch gegen solche Simulationen.

9.1 Bi- bzw. Multistatischer Fehler

Der bi- bzw. multistatische Fehler beruht auf der Annahme, dass der Abstand von Kamera zum Objekt der halben Signallaufzeit entspricht.

$$\tau_L = \frac{2R}{c_0} \quad \text{bzw.} \quad R = \frac{\tau_L \cdot c_0}{2} \quad (9.1)$$

Diese Formel gilt jedoch nur dann, wenn der Hin- und Rückweg gleich lang sind. Dies trifft nur dann zu, wenn der Sender und Empfänger an derselben Stelle positioniert sind. Bei realen PMD Kameras kann diese Anforderung nicht erfüllt werden, da sich der im Verhältnis zum Sensor meistens größere und räumlich ausgedehnte Beleuchter baubedingt immer außerhalb der Kamera befindet. Daher sind die auf der Gleichung (9.1) basierenden Abstandsberechnungen im Falle einer PMD-Kamera grundsätzlich fehlerbehaftet. Die Größenordnung des Messfehlers bei einer Lichtquelle kann relativ leicht abgeschätzt werden, wenn die relative Position des Beleuchters bezüglich der Kamera bekannt ist, siehe zur Veranschaulichung das 2D Modell in der Abbildung 9.1. Die von der Lichtquelle tatsächlich zurückgelegte Strecke setzt sich aus den Teilstrecken Beleuchter-Objekt und Objekt-Beleuchter zusammen. Für den tatsächlichen und den gemessenen Abstände gilt:

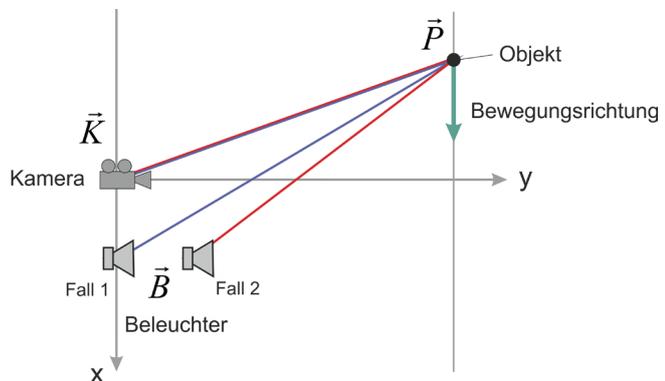


Abbildung 9.1: Illustration bistatischer Anordnungen

$$R_{\text{tatsächlich}} = \overline{KP} \quad \text{bzw.} \quad R_{\text{gemessen}} = \frac{1}{2} (\overline{BP} + \overline{KP}) \quad (9.2)$$

Für den bistatischen Messfehler folgt demnach:

$$\text{Messfehler}_{Bi} = R_{\text{gemessen}} - R_{\text{tatsächlich}} = \frac{1}{2}(\overline{BP} - \overline{KP}) \quad (9.3)$$

Der Fehler hängt somit einerseits von der Position des Beleuchters und andererseits von der Position des Objektes bzw. Punktzieles ab. Die relative Position des Beleuchters ist in den meisten Anwendungen fest bezüglich der Kamera. Die Position des Objektes ist unbekannt, somit ist auch die Länge der Teilstrecke \overline{BP} unbekannt. In Abbildung 9.2 sieht man den Verlauf des Messfehlers für die in Abbildung 9.1 dargestellte Bewegung des Objektes parallel zur x-Achse im Bereich von $[-0.5, 1]$ bis $[0.5, 1]$ m. Der rote und der blaue Verlauf (Abbildung links) stellen den berechneten Abstandsverlauf für zwei unterschiedliche Beleuchterpositionen $[0, 0.2]$ und $[0.2, 0.2]$ m dar, der schwarze Verlauf – den tatsächlichen Kamera-Objekt-Abstand. Alle Größen sind in Metern dargestellt. Die Abbildung rechts zeigt den Verlauf des Messfehlers.

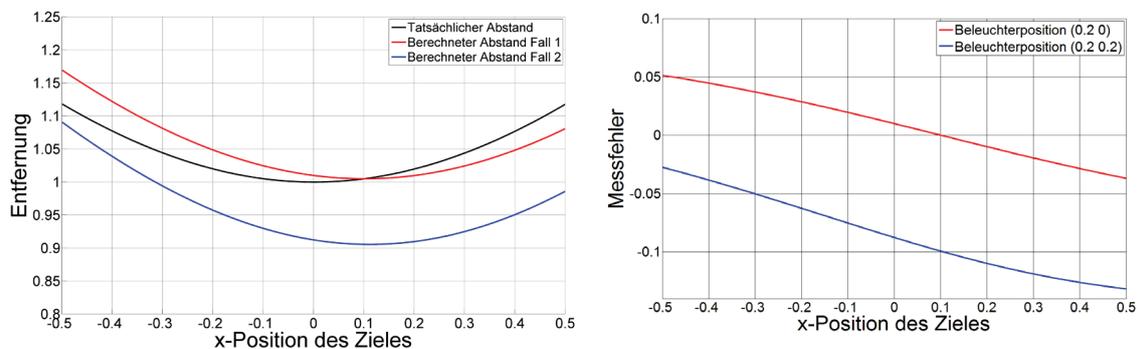


Abbildung 9.2: Berechnete und tatsächliche Entfernung (links); Entfernungsmessfehler bei einer Lichtquelle an unterschiedlichen Positionen (rechts)

Abbildung 9.3 zeigt den ortsabhängigen Entfernungsmessfehler aufgrund des bistatischen Aufbaus. Ein Punktstrahler ist an der Position $[0, 0.2]$ m platziert, der Sensor liegt im Ursprung.

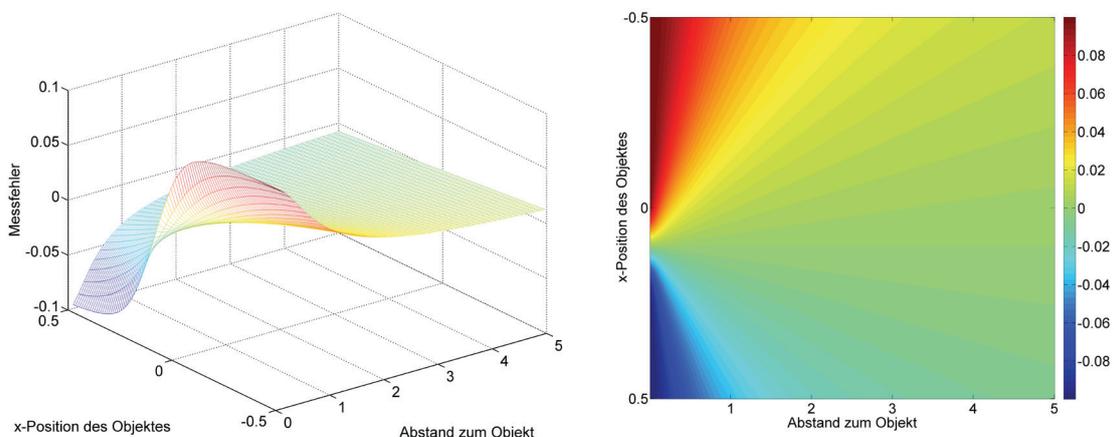


Abbildung 9.3: Entfernungsmessfehler als Funktion des Ortes im Fall einer Lichtquelle im Punkt $[0, 0.2]$ bei unterschiedlichen Objektpositionen im Raum (links – 3D Darstellung, rechts - Farbdarstellung)

Dargestellt ist der Fehler im Bereich $x \in [-0.5, 0.5]$ m, $y \in [0.1, 5.0]$ m. Abbildung 9.4 zeigt den Verlauf des Messfehlers für den zweiten Fall (Punktstrahler an der Position $[0.2, 0.2]$ m).

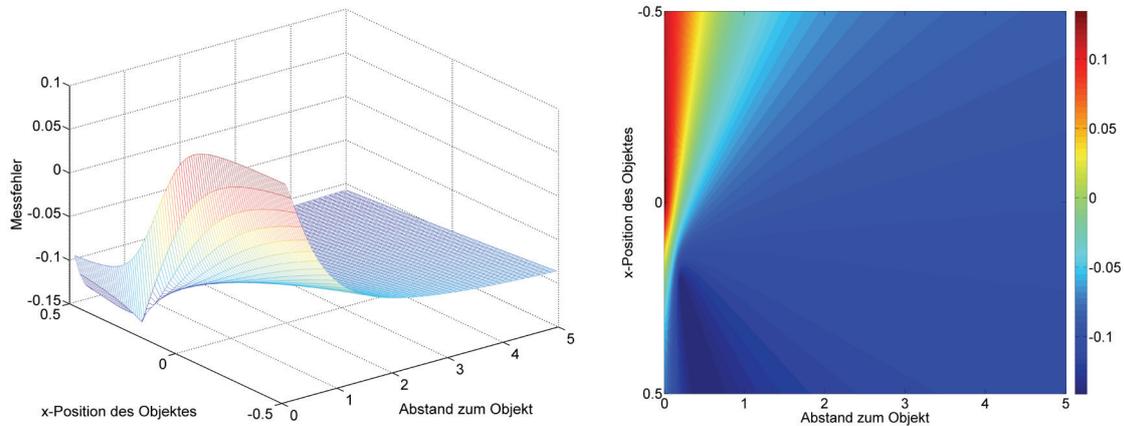


Abbildung 9.4: Entfernungsmessfehler als Funktion des Ortes im Fall einer Lichtquelle im Punkt $[0.2, 0.2]$ bei unterschiedlichen Objektpositionen im Raum (links – 3D Darstellung, rechts - Farbdarstellung)

Wie man in den Abbildungen gut erkennen kann, liegt der systematische Messfehler aufgrund des bistatischen Aufbaus im ungünstigsten Fall bei über 10 cm. Wie zu erwarten war, sinkt der Messfehler mit steigendem Abstand zum Objekt hin. Liegt der Beleuchter im gleichen Tiefenbereich, dann läuft der Fehler im Unendlichen gegen Null. Bei Beleuchtern, die einen Tiefenversatz gegenüber der Kamera aufweisen, spiegelt sich dieser auch im Unendlichen bzw. bei größeren Abständen in einem signifikanten Messfehler wieder (ist gleich dem halben Tiefenversatz).

Im Folgenden soll der Einfluss der Beleuchtergeometrie von mehreren Lichtquellen mit bestimmten Abstrahlcharakteristika untersucht werden. Als Simulationsmodell dient die PMD CamCube mit folgenden technischen Daten:

- CMOS Sensor mit einer Auflösung von 200x200 Pixel
- Pixeldimensionen $45 \mu\text{m} \times 45 \mu\text{m}$, Sensorgröße 9x9 mm
- Optischer Füllfaktor 30%
- Öffnungswinkel 32° bei Brennweite $f=16$ mm
- Eingestellte Modulationsfrequenz 20 Mhz, Wellenlänge Beleuchter 870 nm
- Video Bildrate bis zu 40 Bilder/Sek

Der Aufbau der beiden kameraeigenen Beleuchter wird simuliert, wie in Abbildung 9.5 dargestellt.

Als Szenario dient eine diffus reflektierende Wand.

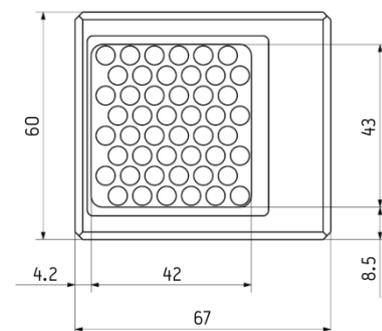


Abbildung 9.5: Aufbau des Beleuchters in einer PMD CamCube (Quelle: © PMD Technologies GmbH)

Die Kamera ist parallel zur Wand ausgerichtet, der Abstand zur Wand beträgt 1.7 m. Beide Beleuchter sind zur Erhöhung des Einflusses des multistatischen Aufbaus nicht direkt an der Kamera, sondern in einem Abstand von 30 cm links und rechts von der Kamera angebracht (gemessen Mitte Kamera bis Mitte Beleuchter). Die Beleuchter sind auf der gleichen Höhe mit der Kamera und haben den gleichen Abstand zur Wand. Abbildung 9.6 zeigt die verwendete Kamera.

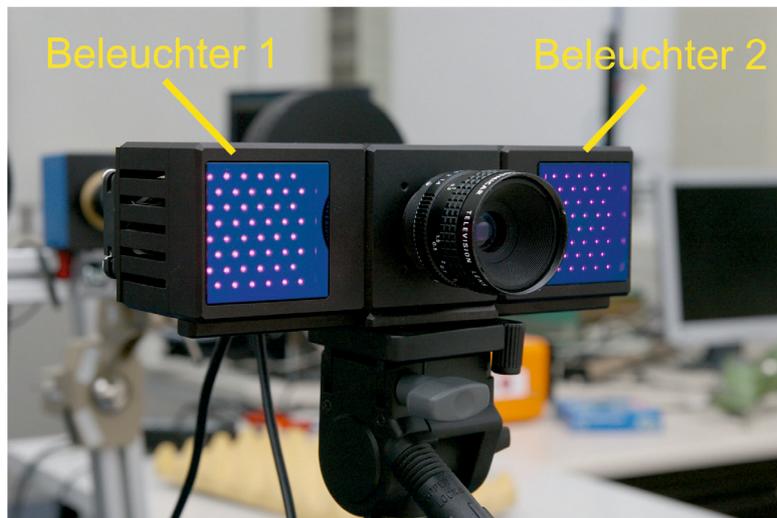


Abbildung 9.6: Die für die Aufnahme eingesetzte PMD CamCube

Für die Erzeugung möglichst rauschfreier Bilder wurden Aufnahmen mit verschiedenen Integrationszeiten gemacht. Doch selbst bei den Aufnahmen mit dem geringsten Rauschen lag die Standardabweichung der Messwerte in einem Bereich von einigen Zentimetern. Um das statistische Rauschen zu minimieren, wurden deshalb jeweils 50 Bilder gemittelt. In Abbildung 9.7 ist der Vergleich eines Tiefenbildes mit einem gemittelten Tiefenbild dargestellt.

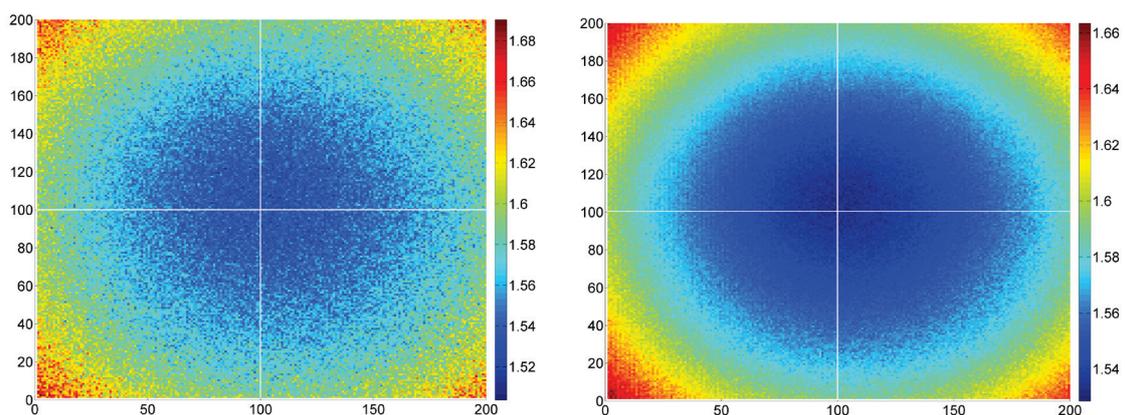


Abbildung 9.7: Farbkodierte Tiefenbilder : Links – Tiefenbild eines Frames; Rechts – Tiefenbild über 50 Frames gemittelt

Wie man gut erkennen kann, lässt sich das statistische Rauschen auf diese Weise signifikant reduzieren. Eine solche Mittelung lässt sich natürlich nur bei statischen Aufnahmen anwenden, was in diesem Fall gegeben ist.

Nun wird die Wand mit Hilfe der Kamera einmal mit beiden Beleuchtern und einmal mit nur einem Beleuchter aufgenommen. Die Messwerte werden mit den Simulationsergebnissen verglichen. Jegliche Art von Rauschen wurde in der Simulation abgeschaltet, es handelt sich um ungestörte ideale Datensätze. Die Wand wurde als Lambert-Rückstrahler angenommen.

Experiment 1: Beide Beleuchter aktiv

Abbildung 9.8 zeigt einen Vergleich der Simulationsergebnisse mit den Messdaten.

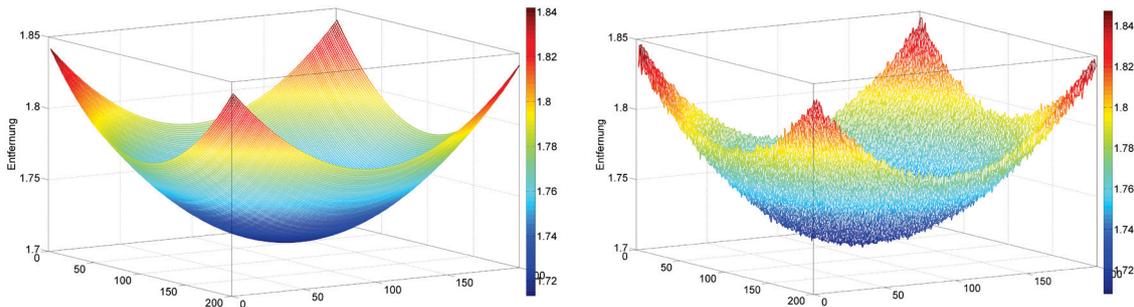


Abbildung 9.8: Farbkodierte Tiefenbilder : Links – Simulationsergebnisse; Rechts – Messdaten (Tiefenbilder über 50 Frames gemittelt), Integrationszeit = 3 ms

Die simulierten Tiefenbilder stimmen recht gut mit den gemessenen überein. Die Entfernungswerte sind in relativen Sensorkoordinaten dargestellt. Mit steigendem Seitenwinkel wird die Entfernung zur Wand hin größer. Der Verlauf der Differenz zwischen den simulierten und den gemessenen Tiefenbildern ist in Abbildung 9.9 dargestellt. Alle Tiefenwerte sind in Metern angegeben.

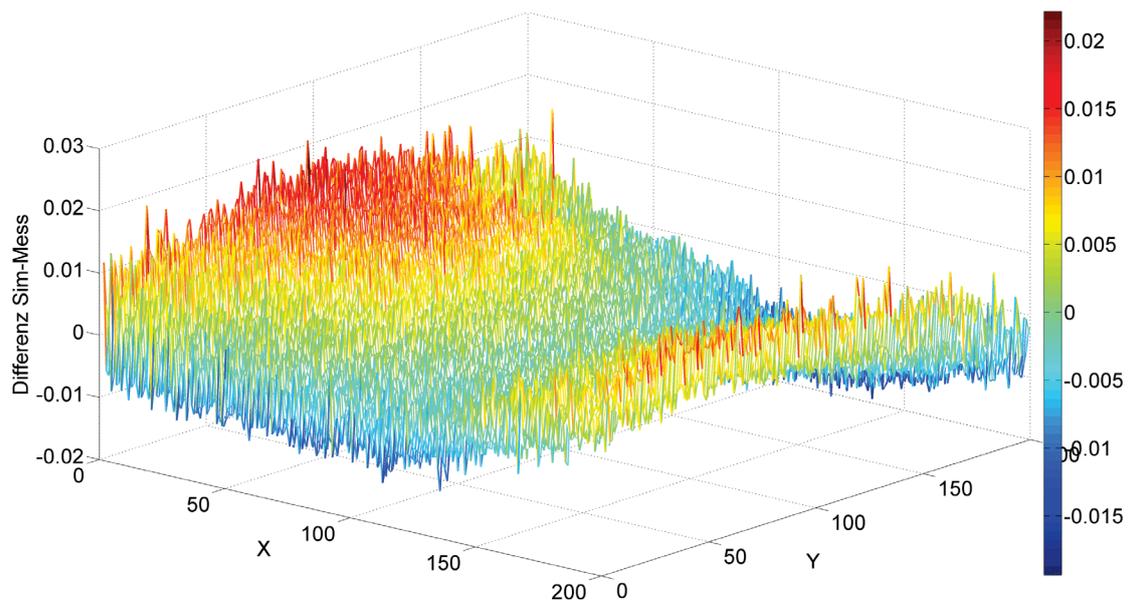


Abbildung 9.9: Differenz zwischen den simulierten und den gemessenen Tiefenbildern

Besonders im mittleren Bereich des Sensors liegt der Fehler unterhalb eines Zentimeters. Seitlich zur Kamera nimmt das Rauschen etwas zu. Der Grund dafür kann eine am

Rand nicht mehr optimale Beleuchtung für die eingestellte feste Integrationszeit von 3ms sein. Außerdem fällt auf, dass der Verlauf entlang der x-Achse etwas gekippt zu sein scheint. Am rechten Rand des Sensors (links im Bild) sind der Fehler und auch das Rauschen etwas größer. Eine Erklärung dazu kann u.a. eine nicht 100%-ge parallele Ausrichtung des Sensors zur Wand sein. Ein Ausrichtungswinkelfehler von ca. einem Grad führt bei dieser Anordnung bei äußeren Pixeln des Sensors zu einer Entfernungsverzerrung im Bereich eines Zentimeters. Abbildung 9.10 zeigt die Verläufe der gemessenen Tiefenbilder der einzelnen Zeilen und Spalten.

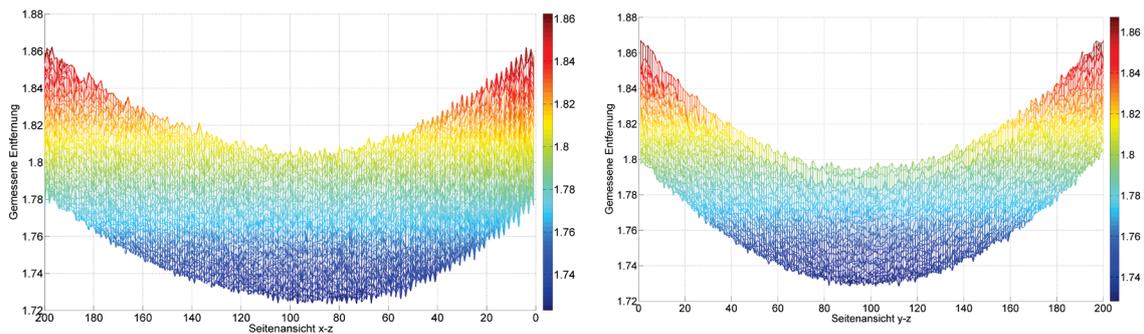


Abbildung 9.10: Links – gemessene Verläufe der Tiefenwerte der einzelnen Zeilen (Seitenansicht x-z); Rechts – gemessene Verläufe der Tiefenwerte der einzelnen Spalten (Seitenansicht y-z)

In Abbildung 9.11 ist der Verlauf der simulierten Tiefenwerte der einzelnen Zeilen dargestellt. Da es unmöglich ist, die Verläufe aller 200 Zeilen und Spalten darzustellen, sollen die Abbildungen nur die gemessene Symmetrie in den Verläufen verdeutlichen.

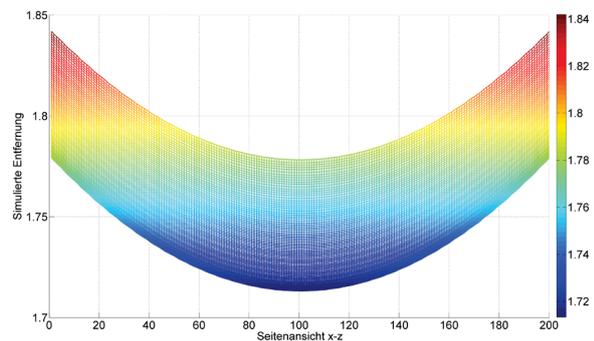


Abbildung 9.11: Verlauf der simulierten Tiefenwerte der einzelnen Zeilen

Deutlich zu erkennen ist der minimale Tiefenoffset in dem mittleren Bereich der Szene, der in der Simulation bei 13.3 mm liegt. Bei den aufgenommenen Daten liegt er in etwa im gleichen Bereich.

Experiment 2: Rechter Beleuchter 1 aktiv

Nun wird der linke Beleuchter abgeschaltet und die Szene nur mit einem Beleuchter aufgenommen. Um die gleiche Qualität der Belichtung zu erzielen, wird die Integrationszeit auf 6 ms angehoben (doppelt so hoch wie zuvor).

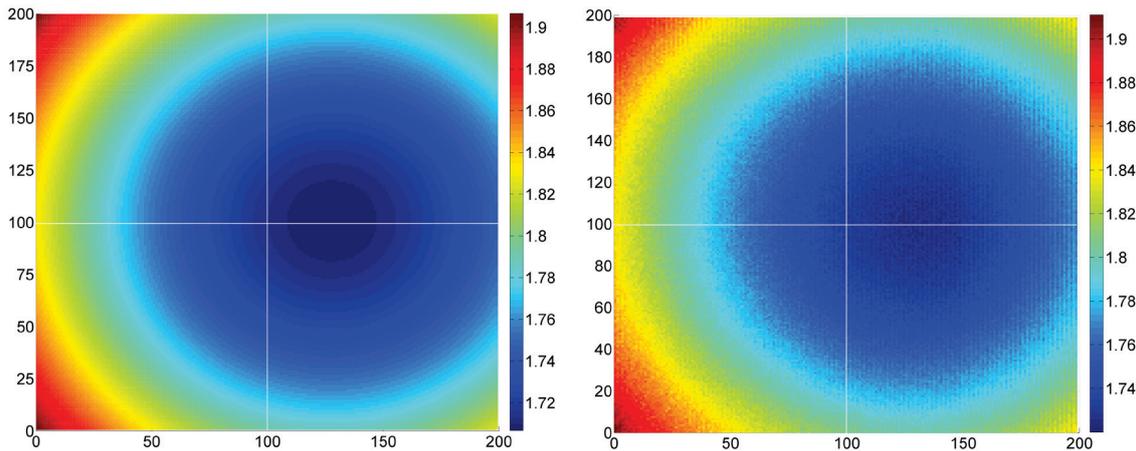


Abbildung 9.12: Farbodierte Tiefenbilder : Links – Simulationsergebnisse; Rechts – gemessene Tiefenbilder (über 50 Frames gemittelt)

Abbildung 9.12 zeigt die simulierten und die gemessenen Werte der asymmetrischen multistatischen Beleuchter-Kamera Anordnung. In Abbildung 9.13 sind die entsprechenden Verläufe der Tiefenwerte der einzelnen Zeilen dargestellt.

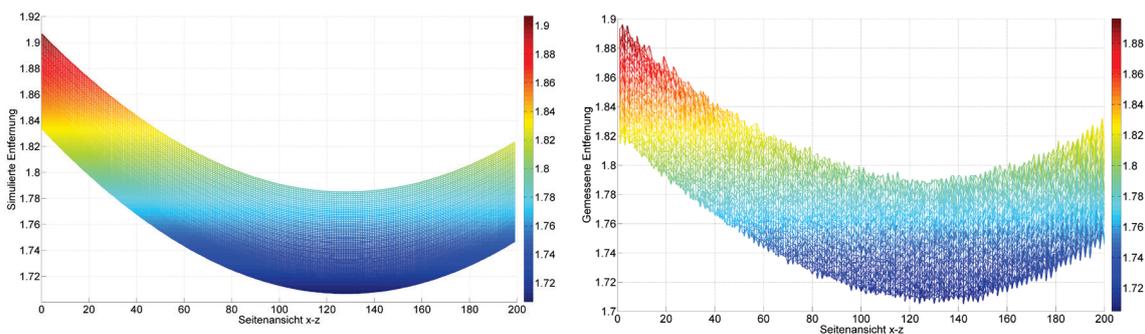


Abbildung 9.13: Verlagerung des Zentrums des minimalen Abstandes aufgrund des multistatischen Aufbaus: Links – Verläufe der Tiefenwerte der einzelnen Zeilen des simulierten Sensors (Seitenansicht x-z); Rechts – gemessene Verläufe der Tiefenwerte der einzelnen Zeilen (Seitenansicht x-z)

Man kann sehr gut erkennen, dass bei nur einem Beleuchter die gemessene Entfernung bezüglich der Mitte des Sensors in x -Richtung, also zur Seite hin, nicht mehr symmetrisch ist. Es resultiert daraus, dass sich die minimale von dem Signal durchlaufende Strecke $\min\{R_{\text{Beleuchter-Objekt}} + R_{\text{Objekt-Sensor}}\}$ nun in der senkrechten Projektion des Mittel-

punktes zwischen dem Beleuchter und der Kamera auf die Wand befindet. Alle Entfernungswerte werden von dieser Geometrie beeinflusst. Der vertikale Tiefenunterschied für einzelne Spalten ist dagegen, wie Abbildung 9.14 verdeutlicht, weitgehend symmetrisch, in der Simulation ist er sogar absolut symmetrisch, da es in y -Richtung keine Asymmetrie in der

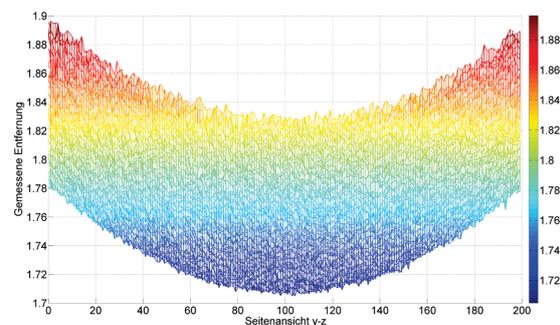


Abbildung 9.14: Gemessene Verläufe der Tiefenwerte der einzelnen Spalten (Seitenansicht y-z)

Anordnung gibt. Die Differenz zwischen den gemessenen Entfernungsbildern aus beiden Experimenten (Beleuchter 1+2 AN und nur Beleuchter 1 AN) ist in Abbildung 9.15 illustriert.

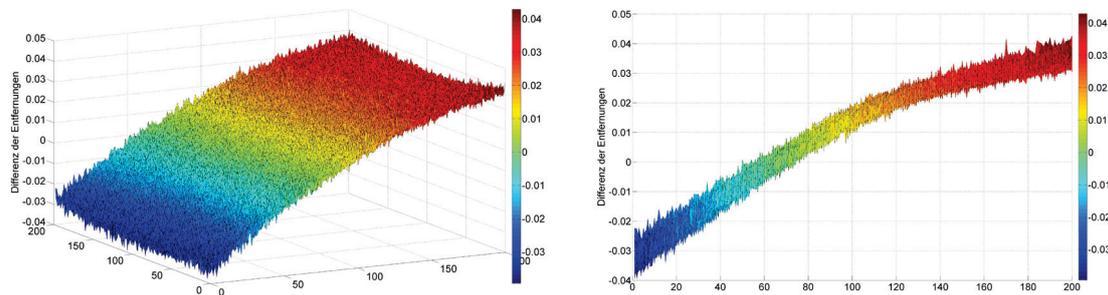


Abbildung 9.15: Differenz der gemessenen Tiefenbilder bei einem und zwei aktiven Beleuchtern (Differenz zwischen den Aufnahmen der beiden Experimente): Links – 3D Darstellung; Rechts – Seitenansicht der gleichen Darstellung (Seitenansicht x-z)

Zum Vergleich ist in Abbildung 9.16 die Differenz der simulierten Tiefenbilder dargestellt. Beide Verläufe zeigen den gleichen Trend und stimmen gut überein. Die durchschnittliche gemessene Differenz liegt bei ca. 3.2cm links im Bild und bei ca. 3.6 cm rechts. Die maximalen Abweichungen in der Simulation liegen beidseitig im Bereich 4.1 cm.

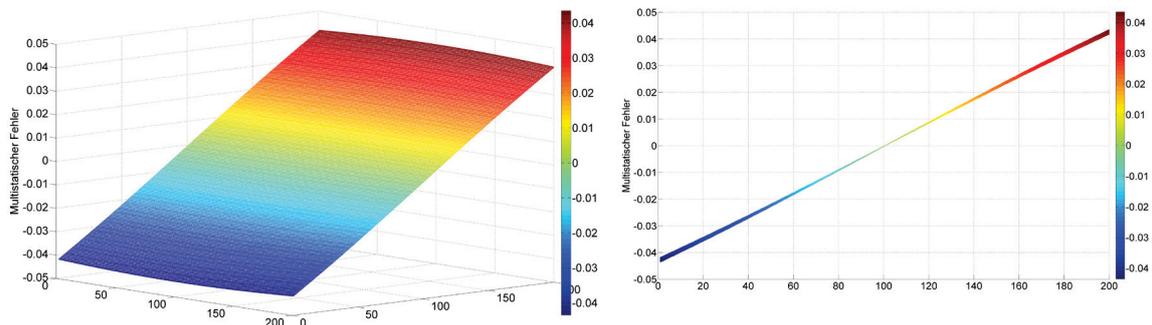


Abbildung 9.16: Differenz der simulierten Tiefenbilder bei einem und zwei aktiven Beleuchtern (Differenz zwischen den Simulationsergebnissen der beiden Experimente): Links – 3D Darstellung; Rechts – Seitenansicht der gleichen Darstellung (Seitenansicht x-z)

Offenbar sind die gemessenen durchschnittlichen Abweichungen sogar noch etwas kleiner als die der Simulation. Ein Grund dafür kann u.a. in der nicht exakt übereinstimmenden Abbildung des Lochkammermodells liegen. Stimmt der Öffnungswinkel des Objektivs nicht exakt mit dem in der Simulation angenommenen überein, wird in der Simulation ein etwas größerer Bereich abgebildet, wodurch der Fehler am Rand zunimmt. Zusätzlich dazu kommt noch das Problem der radialen Verzerrungen bei realen Objektiven.

Das im 2. Experiment dargestellte Szenario stellt mit nur einem aktiven Beleuchter relativ weit von der Kamera positioniert eher einen Sonderfall dar. Doch in manchen Anwendungen, wie z.B. im Automotivbereich, kann es aus technischen und optischen Gründen notwendig sein, die recht großen aktiven Beleuchter weit weg von der Kamera

einzubauen. In einem solchen Fall muss eine möglichst optimale Anordnung gefunden werden. Wie zahlreiche andere Experimente und Simulationen gezeigt haben, führen symmetrische Anordnungen der Beleuchter bezüglich der Sensorhauptachsen zu einem deutlich konstanteren und „berechenbaren“ Fehlerverlauf. Einen Vergleich für unterschiedliche Beleuchteranordnungen und entsprechende Distanzbilder findet man in [NeRa 11]. Eine symmetrische Anordnung führt zu einem bezüglich der Sensormitte symmetrischen multistatischen Fehler, dessen größter Teil aus einem Tiefenoffset besteht. Unsymmetrische Anordnungen führen dagegen zu starken Seitenunterschieden (siehe dazu auch [Pet 07b]).

Betrachtet man den theoretischen Fehler, hervorgerufen durch den multistatischen Aufbau bei einer CamCube mit den Beleuchtern direkt an der Kamera, dann liegt der multistatische Fehler bei einer Entfernung von 1,7 m unter einem Millimeter und somit weit unterhalb der Auflösungsgrenze. Abbildung 9.17 illustriert den simulierten Verlauf des Messfehlers. Bei einem Abstand von 0,5 m steigt der Fehler auf über 3 mm an, was bereits in dem Bereich der Entfernungsauflösung der Kamera liegt.

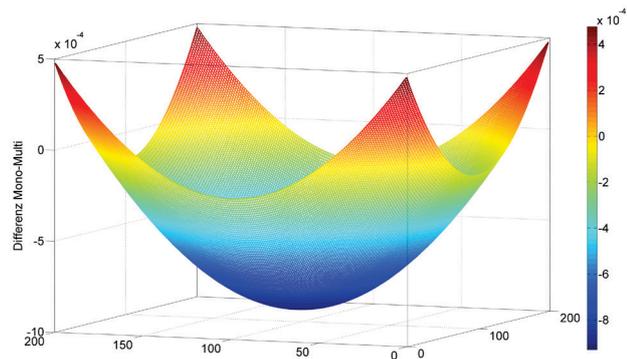


Abbildung 9.17: Simulierter multistatischer Fehler bei zwei aktiven Beleuchtern einer Standard CamCube-Anordnung

Abschließend muss noch bemerkt werden, dass aufgrund der unterschiedlichen Positionen der einzelnen Pixel in der Pixelmatrix grundsätzlich eine gewisse Asymmetrie zu den einzelnen Beleuchtern entsteht. Bei dem in der CamCube eingesetzten 9x9 mm großen Sensor beträgt der maximale Versatz eines Pixels gegenüber der Sensormitte ca. 6,4 mm. Der dadurch theoretisch entstehende Fehler ist jedoch in der Praxis nicht relevant und zu vernachlässigen.

9.2 Simulation der Bewegungsunschärfe

Nachfolgend soll die in Kapitel 5.8 behandelte Simulation der Bewegungsunschärfe illustriert werden. Wie bereits erwähnt, kann sich dieser Effekt bei hohen Integrationszeiten im Zusammenhang mit schnellen Bewegungen zeigen, z.B. beim Bewegen der Kamera quer zu einer Kante. Solche Bewegungen spielen besonders im Nahbereich einer Kamera eine große Rolle. Mit steigender Entfernung sinkt die relative Änderung der Position eines sich mit konstanter Geschwindigkeit an der Kamera vorbei bewegendes Objektes und somit der Effekt der Bewegungsunschärfe.

Bei der Pixelgröße von $45\ \mu\text{m}$ der zu Verfügung stehenden CamCube, löst ein Pixel in der Mitte des Sensors bei dem verwendeten Objektiv in einem Abstand von einem Meter einen ca. $9\ \text{mm}^2$ großen Bereich auf (am Rand ist der Bereich deutlich größer). Bei solchen Entfernungen liegen die optimalen Integrationszeiten oft im Bereich 1-2 ms. Bei Abblenden kann die Integrationszeit zwar erhöht werden, dies erfordert aber eine stärkere Beleuchtungsstärke, die mit Standard-Beleuchtern nicht zu erreichen ist. Um die Änderungen der Werte in einem Pixel festzustellen, muss die Kantengeschwindigkeit für das oben genannte Beispiel im Bereich 3 m/s liegen. Solche recht hohe Geschwindigkeiten mit der Genauigkeit im Millimeterbereich sind für eine kontrollierte Aufnahme unter Laborbedingungen recht schwer zu verwirklichen. Dazu kommt noch die Schwierigkeit der Synchronisation des Ablaufes. Der Sensor soll genau dann auslösen, wenn die Kante an einer bestimmten Position steht.

Unkontrolliert, bzw. nur zum Teil kontrolliert lässt sich die Bewegungsunschärfe dennoch relativ leicht aufnehmen, indem man ein Objekt an der Kamera schnell vorbeiführt, z.B. wenn man den freien Fall einer Holzplatte aufnimmt, deren Geschwindigkeit sich sogar in gewissen Grenzen bestimmen lässt. So bleibt die Kamera lagestabil und wird keinen weiteren Einflüssen ausgesetzt. In Abbildung 9.18 ist beispielhaft eine solche Aufnahme präsentiert. Zur besseren Übersichtlichkeit wurden nur die mittleren 5 Spalten im Bereich der Kante ausgeschnitten.

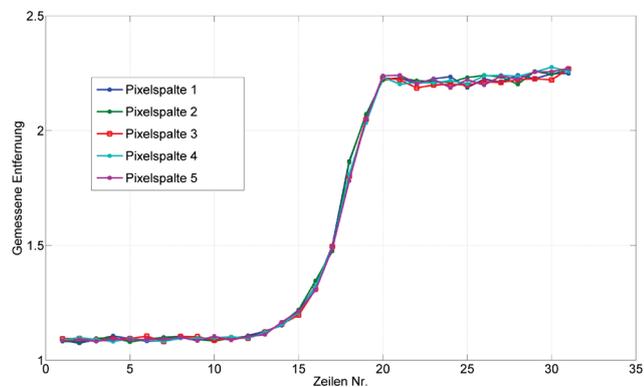


Abbildung 9.18: Aufnahmen: Kantenglättung infolge der Bewegungsunschärfe

Zur besseren Übersichtlichkeit wurden nur die mittleren 5 Spalten im Bereich der Kante ausgeschnitten. Die Kamera stand vor einer Wand in einem Abstand von ca. 2,25 m. Die sich schnell bewegende Platte im Vordergrund der Kamera hatte einen Abstand von ca. 1,1 m zur Kamera. Zwischen der 15. und 20. Zeilen sieht man eindeutig den Effekt der „Kantenglättung“. Die Reflektionen des Hintergrundes vermischen sich während der Integrationszeit mit den Reflektionen der Platte. Aufgrund der so entstehenden Mischphasen wird, je nach Anteil der einzelnen Reflektionen, eine Entfernung gemessen, die zwischen den beiden Werten liegt.

Die folgenden Abbildungen zeigen die Simulationsergebnisse bei unterschiedlichen Einstellungen der Zwischenframe/Frame-Rate (entspricht der simulativen Nachbildung der Integrationszeit) und Objektgeschwindigkeit. Die Simulation ist nicht absolut komplementär zu den in Abbildung 9.18 gezeigten Aufnahmedaten, da es hierfür die exakte Kenntnis der Plattengeschwindigkeit erfordert. Es soll lediglich der Einfluss der Berechnung eines Frames aus Zwischenframes veranschaulicht werden. Als Simulationsszenario dienten zwei Platten – eine davon als Hintergrund (Objekt 2), die andere wurde mit einer konstanten Geschwindigkeit senkrecht zur optischen Achse bewegt (Objekt 1),

siehe Abbildung 9.19 links. Die Geschwindigkeit wurde so gewählt, dass die sich bewegende Platte in der Simulationszeit zwischen zwei Frames eine Strecke zurücklegte, die der Abbildungsstrecke von mindestens einigen Pixeln entsprach. Die Entfernungen sind als Grauwertbilder dargestellt, die gleich normiert sind. In Abbildung 9.19 ist ein Ausschnitt einer Momentaufnahme der Kante bei Zwischenframe/Frame-Rate=1 zu sehen. In Abbildung 9.20 sind zwei Frame-Aufnahmen bei einer Zwischenframe/Frame-Rate=4 mit der Bestimmung der Phasenverschiebung nach (2.19) (entspricht dem 4-Phasenalgorithmus) und unterschiedlichen Objektgeschwindigkeiten dargestellt. Im Bild rechts bewegte sich die Platte mit doppelter Geschwindigkeit. Man erkennt sehr deutlich den Effekt des Verwischens der Kante. Mit steigender Zwischenframe/Frame-Rate wird das Verwischen genauer nachgebildet, die Simulationszeit steigt jedoch um den gleichen Faktor an. Der Grund, dass die Kante in Abbildung 9.19 dennoch etwas verwischt aussieht, liegt darin, dass sie sich bei einer Momentaufnahme in der Regel mitten in einer bestimmten Zeile befindet. In dieser Zeile können alle Pixel sowohl die Reflektionen (Phasen) von Objekt 1, als auch die Phasen von Objekt 2 detektieren. Aufgrund des Phasenmischprozesses wird entsprechend der Anteile der einzelnen Objekte eine Mischphase berechnet, die zwischen den Signallaufphasen der beiden Objekte liegt. Die berechnete Entfernung liegt somit in der Regel ebenfalls zwischen den Entfernungen der beiden Objekte.

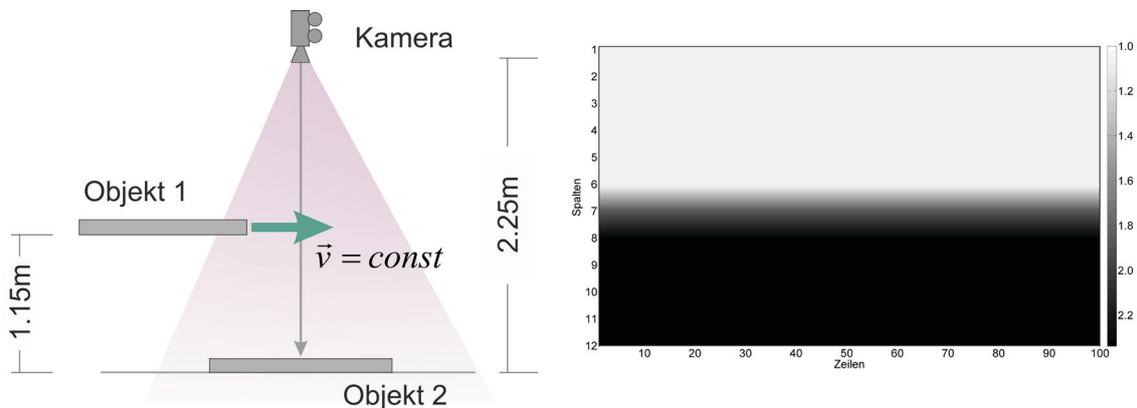


Abbildung 9.19: Links – Simulationsszenario; Rechts – Normiertes Tiefenbild bei Aufnahme einer Kante (keine Zwischenframes)

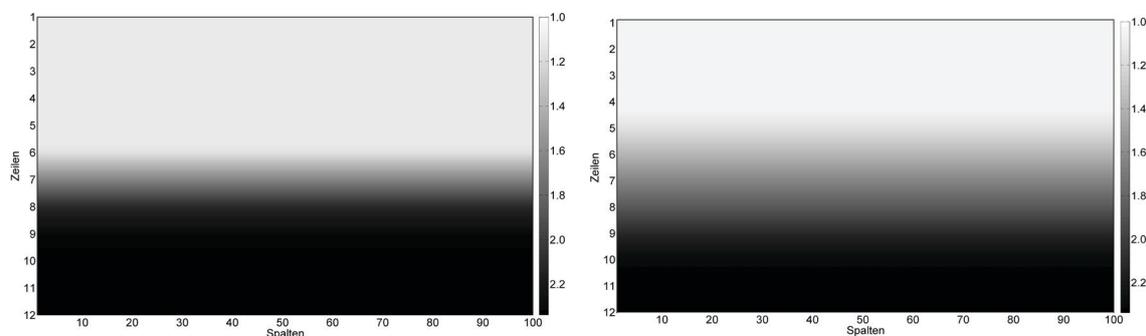


Abbildung 9.20: Normierte Tiefenbilder bei Berechnung mit Zwischenframes: Links – Normiertes Tiefenbild bei Aufnahme einer Kante; Rechts – gleiches Szenario mit doppelter Geschwindigkeit des Objekts 1

9.3 Simulation: Abstandsmessung an schrägen Oberflächen

Neben Objektkanten stellen unter realen Bedingungen zum Sensor hin geneigte Oberflächen die mit Abstand häufigste Form des Phasenmischprozesses dar. Reale Objekte sind selten parallel zur Kamera ausgerichtet, im Allgemeinen haben die von einem Pixel erfassten Oberflächen eine bestimmte Neigung zum Richtungsvektor in Richtung des Szenenausschnittes. Von großer Bedeutung sind hierbei nicht nur die Größe des von einem Pixel erfassten Szenenausschnittes und der Neigungswinkel der Oberfläche, sondern auch die Position und die Abstrahlrichtung des Beleuchters. Abbildung 9.21 veranschaulicht die Problemstellung.

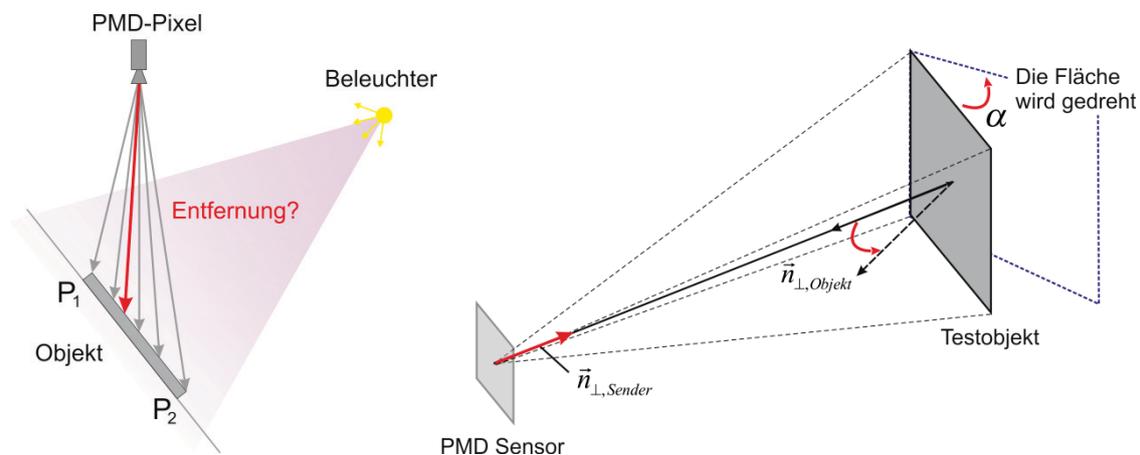


Abbildung 9.21: Simulationsszenario: Entfernungsberechnung bei einer schrägen Oberfläche

Das folgende Simulationsszenario soll den Unterschied der simulierten Entfernungswerte in Abhängigkeit des Neigungswinkels einer flachen Oberfläche klarstellen. Betrachtet werden die simulierten Entfernungsverläufe eines Pixels. Die Oberfläche wird in einem Winkelbereich von 0 bis 80 Grad gedreht und zwar an der äußeren Kante (siehe Abbildung 9.21, rechts). Der Abstand zur Fläche beträgt im Ausgangszustand, also bei $\alpha = 0$, 1 m. Der Öffnungswinkel eines Pixels ist so gewählt, dass dieses in einem Abstand von 1 m eine Fläche von 1 cm² auflösen kann (für einen rein qualitativen Vergleich ist diese Größe jedoch nicht entscheidend). Als Beleuchter dient ein \cos^{12} -Strahler, die Oberfläche wird als Lambert-Reflektor angenommen.

Zunächst wird ein monostatisches Szenario betrachtet. Der Hauptrichtungsvektor des Beleuchters fällt mit der optischen Achse des Sensors (Pixels) zusammen und steht in der Ausgangssituation senkrecht auf der Testfläche. In Abbildung 9.22 sind der simulierte Abstand und der Abstand zur Mitte der Testoberfläche dargestellt (blauer und schwarzer Verlauf). Man kann sehen, dass die Verläufe stark unterschiedlich sind. Das simulierte Ergebnis stellt also nicht einmal näherungsweise ein geometrisches Mittel aller in der Fläche befindlichen Punkte dar. Das ist auch verständlich. Die Bereiche der Szene, die näher zum Sensor sind, bringen eine höhere Gewichtung der Signallaufphasen (die Signalstärke ist dann bei realen Kameras höher). Somit beeinflussen sie das

Ergebnis stärker als die hinteren Bereiche der Platte. Aber wie bereits diskutiert, spielt nicht nur die Entfernung eine Rolle. Wie man in Abbildung 9.21 sehen kann, steigt der Winkel zwischen der Normalen auf die Objektfläche und dem Blickwinkel zu einem Flächenpunkt mit der steigenden Entfernung dieses Punktes am sich drehenden Rand durch die Drehung an (vergleiche die Winkel im P_1 und P_2). Aufgrund des photometrischen Entfernungsgesetzes wird somit die zurückgestrahlte Lichtleistung kleiner, was sich dann über die Amplitude der theoretischen Antwort auf die Gewichtung der Phasenwerte äußert. Zum Vergleich wird nun ein bistatischer Fall mit einem seitlich des Sensors versetzten Beleuchter betrachtet. Der bistatische Abstand beträgt 7,6 cm (entspricht dem Abstand einer CamCube), die restlichen Parameter sind die gleichen. Der rote Verlauf stellt die Ergebnisse für den bistatischen Fall dar.

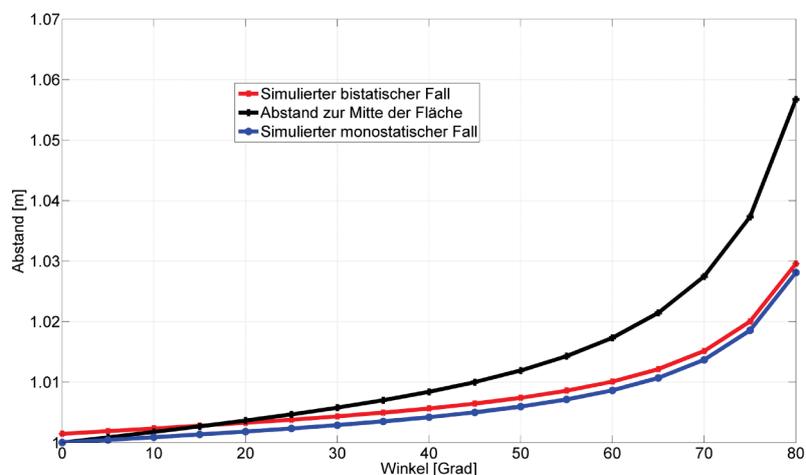


Abbildung 9.22: Winkelabhängigkeit des simulierten Abstandes bei schrägen Oberflächen

Zu guter Letzt spielt auch die Ausrichtung des Beleuchters eine wichtige Rolle. Die Lichtstärke des reflektierten Lichtes beeinflusst direkt den im Sensor stattfindenden Mischprozess, die Anteile der Szene mit hoher Reflektivität liefern über die folglich höhere Ladungsträgergenerierung einen höheren Beitrag zu den Korrelationsfunktionen. Der Einfluss der unterschiedlichen Lichtverhältnisse in einem von einem Pixel erfassten Szenenausschnitt ist jedoch im Allgemeinen aufgrund der kleinen Größen der Pixel relativ gering und liegt bei dem hier betrachteten Beispiel, wenn man den Beleuchter z.B. um 15 Grad verdreht, im Bereich 10^{-4} – 10^{-3} bezogen auf die maximale Helligkeit im monostatischen Fall.

Um diesen Effekt dennoch zu verdeutlichen wird nun ein weiteres Szenario betrachtet (siehe Abbildung 9.23). Der Abbildungsbereich eines Pixels wird übertrieben vergrößert

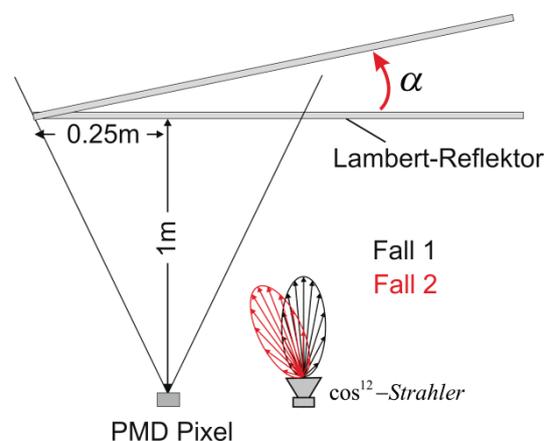


Abbildung 9.23: Das bistatische Simulationsszenario (Fall 1 und Fall 2)

bert und beträgt nun in einem Abstand von einem Meter $0,5 \times 0,5$ m. Der Beleuchter wird seitlich des Sensors um $0,5$ m versetzt. Im Fall 1 (schwarze Verläufe) ist der Beleuchter parallel zur Kamera ausgerichtet. Im 2. Fall wird er um 15 Grad zur Fläche hin gedreht. Die Abbildung 9.24 links zeigt die normierten Helligkeitsverläufe für unterschiedliche Rotationswinkel α . Man erkennt einen deutlichen Unterschied aufgrund der Neigung des Beleuchters. Im 2. Fall werden die zum Sensor näheren Bereiche der Fläche besser beleuchtet, wodurch sie nicht nur einen erhöhten Beitrag beim Helligkeitsverlauf, sondern auch beim Entfernungsverlauf leisten (siehe die Verläufe rechts). Die Unterschiede betragen in diesem übertriebenen Fall bis zu 20 cm. Unter realen Bedingungen unterliegen diese Einflüsse, wie bereits geschrieben, deutlich den Entfernungseinflüssen und den Einflüssen aufgrund des photometrischen Gesetzes.

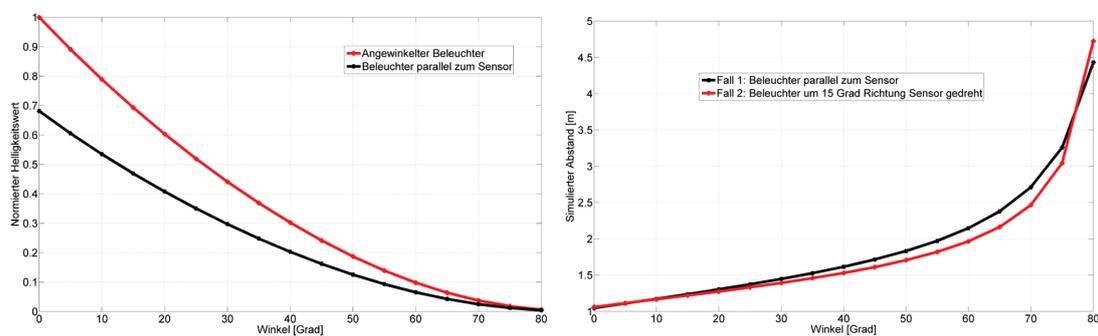


Abbildung 9.24: Die simulierten Helligkeitswerte für Fall 1 und Fall 2 (links); Die simulierten Entfernungswerte (rechts)

10 Zusammenfassung und Ausblick

10.1 Zusammenfassung

Das Ziel der vorliegenden Arbeit lag in der Entwicklung eines PMD-Simulators zur Generierung synthetischer Sensordaten. Solche Referenzdaten sind sehr hilfreich bei Untersuchungen der Auswirkungen unterschiedlicher Sensorgeometrien und konstruktiver Kameradesigns. Viele physikalische Effekte, auftretende Störphänomene und das Ausmaß ihres Einflusses auf die Messunsicherheit des gesamten Messsystems können mit Hilfe von Simulationen ermittelt, abgeschätzt und untersucht werden.

Besonderer Wert beim Entwurf des Simulators wurde gelegt, zum einen auf ein einfaches und überschaubares Grundkonzept und zum anderen auf die Realisierung eines weitgehend frei parametrisierbaren Modells, dessen wesentliche Funktionalitäten einzeln oder in einer Kombination untersucht werden können. So kann der Einfluss der möglichen Fehlerquellen, wie z.B. das Verrauschen der Korrelationsfunktionen, die Winkelabhängigkeit der Lichtintensitätsverteilung, das Verdecken der einzelnen Lichtelemente, die Neigung des Sensors, die optischen Verzerrungen usw. einzeln oder in einer festgelegten Kombination analysiert werden.

Desweiteren wurde bei dem Entwurf und der Umsetzung der Simulatorarchitektur der Frage der potentiellen Erweiterbarkeit des Simulators viel Beachtung geschenkt. So wurden viele methodenkritische oder erweiterungsfähige Teile des Simulators in weitgehend unabhängigen Teilkomponenten realisiert, die nach Bedarf erweitert oder geändert werden können. So kann z.B. ein weiteres (umfangreicheres) Modell der optischen Abbildung integriert werden, ohne dass sich der Aufbau und die Ablaufsteuerung des Simulators im Wesentlichen ändert.

Mit dem Ziel der Nachbildung der durch den 4-Phasenalgorithmus entstehenden physikalischen Effekte in den realen PMD-Kameras wurde ein solcher Ansatz auch im Simulator umgesetzt. Dabei werden die synthetischen Entfernungsbilder mittels der Berechnung aus den vier Signalphasenverschiebungen bestimmt. Die letzteren lassen sich entsprechend Vorgaben verrauschen, wodurch Einfluss auf die Messunsicherheit genommen werden kann. Da im Allgemeinen Simulationsszenen eine andere zeitliche Rasterung haben können als das im Simulator gewählte Intervall zwischen den einzelnen Phasenwerten, wurde im Simulator ein weiterer Ansatz implementiert, bei dem infolge eines Mischprozesses aus den einzelnen Zwischenframes synthetische Daten generiert werden können. Im Gegensatz zum 4-Phasenalgorithmus werden dabei die Helligkeits- und die Entfernungsbilder durch eine künstliche Integration über F Zwischenframes berechnet.

Eine der wichtigsten Vorgaben für den zu entwickelnden Simulator war die Möglichkeit der Simulation von unterschiedlichen Szenarien. Obwohl die meisten synthetisch erstellten 3D Umgebungen und Objekte vorwiegend polygonaler Natur sind, wurde die in 3D Grafik seltener zu findende Definition in Punkteform ebenfalls implementiert. Aufgrund der unterschiedlichen Möglichkeiten, gegeben durch diese beiden Formen, wurden im Simulator, je nach eingesetzter Form, entsprechende Algorithmen und Verarbeitungsmethoden entwickelt und implementiert. Beide parallel im Simulator existierenden Zweige unterscheiden sich grundsätzlich bei der Sichtbarkeitsanalyse und der Berechnung der Lichtintensität bzw. der Intensität der zurück reflektierten Lichtwelle. Zur synthetischen Erhöhung der Auflösung der polygonal vorliegenden Zielobjekte wurde mit dem Hintergrund der Nachbildung möglichst realitätsnaher Rückstreuungseigenschaften von Objekten eine optionale Objektinterpolationskomponente realisiert, die im Falle der Überschreitung der photometrischen Grenzentfernung die vorliegenden Daten für die Dauer der Berechnung eines Frames interpoliert.

Zur Modellierung der PMD-Kamera, der Signalgenerierung, -übertragung und -überlagerung wurden vektorbasierte Lösungsansätze gewählt. Sie bieten die Vorteile der genauen Berechnung, basierend auf analytischer Geometrie, kombiniert mit der Möglichkeit der Vereinfachung der oft umfangreichen und aufwendigen Berechnungen, indem man die Komplexität dieser in kleinere Teilprobleme zerlegt. So werden räumlich ausgedehnte Beleuchter durch eine Ansammlung von Punktbelechtern mit frei definierbaren Abstrahlcharakteristika nachgebildet, der PMD-Sensor wird als eine Pixelmatrix mit insgesamt drei möglichen Pixelgeometrien ausgelegt und die Optik wird durch ihre Eigenschaft der geometrischen Abbildung beschrieben.

Die in der Arbeit vorgestellten Ansätze basieren auf der Modellierung einer sinusförmigen Intensitätsmodulation. Wie gezeigt, lassen sich Überlagerungen solcher Wellen mit Hilfe komplexer Rechnung und somit vektorieell lösen, wodurch die Algorithmen schnell, effizient und präzise umgesetzt werden konnten. Prinzipiell können für die Ansteuerung der Beleuchtung auch andere Modulationsformen eingesetzt werden. Die Überlagerung solcher Wellen erfordert jedoch eine grundlegende Überarbeitung der eingesetzten theoretischen Ansätze.

Für eine einfachere Modellierung und systemtheoretische Beschreibung der einzelnen Zusammenhänge und Simulationsprozesse wurde eine Vielzahl von Koordinatensystemen eingeführt. Aufgrund der einfacheren Handhabung werden die Objekte und ihre Transformationen in erweiterten homogenen Koordinaten beschrieben und ausgeführt (siehe Anhang 12.2). Diese Koordinaten ermöglichen Vereinheitlichung und Kombination aller geometrischen Transformationen.

Auf Basis der geforderten Funktionalitäten und der vorgestellten Simulatorarchitektur wurde der PMD-Simulator in Form von einigen wichtigen Schlüsselmodulen entwickelt, umgesetzt und erprobt. Die komponentenweise Aufteilung des Simulators ermöglicht eine bessere Kontrolle der einzelnen Ablaufprozesse, sowie eine einfachere Opti-

mierung und Erweiterbarkeit des Systems. Im aktuellen Entwicklungsstadium beinhaltet der Simulator folgende Funktionalitäten und Ansätze:

- Generierung synthetischer Sensordaten für eine beliebige 3D Szene mit der Option von vordefinierten Reflektionseigenschaften
- Simulation von Bewegungsunschärfe infolge schnellerer Kamera- oder Szenenbewegungen
- Berücksichtigung der Sensorparameter wie Sensorgröße, Pixelgröße, Auflösung, optischer Füllfaktor, Sensorneigung und Sensor-Shift
- Berücksichtigung der Optikparameter wie Brennweite/Öffnungswinkel und Abbildungsgeometrie
- Berücksichtigung der Beleuchter- und Kameraparameter wie Modulationsfrequenz, Wellenlänge, Messwertquantisierung und Abstrahlcharakteristik
- Implementierung unterschiedlicher Ansätze zum Verrauschen der generierten Datensätze

Die Simulationsergebnisse zeigen in vielen Bereichen eine gute Übereinstimmung der simulierten Tiefenbilder mit den realen Messungen. In einigen Testreihen mussten die statisch aufgenommenen realen Daten dafür allerdings über einige Bilder gemittelt werden, da die einzelnen Bilder doch recht stark verrauscht waren. So konnte zumindest das statistische Rauschen in seiner Wirkung reduziert werden. Bei der Simulation der Amplitudenbilder gibt es allerdings folgendes zu beachten. Je nachdem ob die Objektumgebung polygonal oder in Punkteform vorliegt, ergeben sich, auch wenn die Unterschiede nicht gravierend sind, methodenbedingt unterschiedliche simulierte Intensitätsbilder. Die Polygonform erlaubt, eine gute Auflösung der Objekte vorausgesetzt, eine deutlich höhere Stufe an Details, da Reflektionseigenschaften von Objekten mitberücksichtigt werden können. Die Qualität der synthetischen Datensätze ist somit potentiell höher.

Die Qualität der Tiefenbilder ist bei beiden Methoden auf den ersten Blick scheinbar gleich. Ohne Störungen ergeben sich bis auf die Unterschiede, hervorgerufen durch die unterschiedlichen Sichtbarkeitsanalysen und das photometrische Entfernungsgesetz, theoretisch gesehen die gleichen Tiefenbilder. Die Messunsicherheit realer Kameras hängt jedoch, wie in der Arbeit beschrieben, im Wesentlichen von der Anzahl der in den Photogates vom modulierten Licht generierten Ladungen ab. Diese Anzahl ist, wie in (2.26) gezeigt, proportional zu der Menge des Nutzlichtes:

$$\sigma_R = \frac{\partial R}{\partial \varphi} \cdot \sigma_\varphi = \frac{c}{2\omega} \sqrt{\frac{H}{2M^2}} \quad (10.1)$$

Die im Simulator generierten Entfernungswerte lassen sich somit entsprechend dem berechneten Amplitudenwert verrauschen.

Wie bereits in Kapitel 5 behandelt, spiegelt die berechnete von einem Pixel empfangene Lichtmenge nicht exakt, sondern nur qualitativ den von der Kamera gelieferten Helligkeitsmesswert wieder. Wie die ersten Simulationen gezeigt haben, lassen sich synthetische Helligkeitsverläufe in reale mit Hilfe einer Normierung überführen. Diese lässt sich experimentell für eine bestimmte Kamera bestimmen. Inwieweit eine bestimmte Helligkeitsnormierung für andere Kameras und Szenen übernommen werden kann, muss in weiteren umfangreichen Simulationen und Versuchen analysiert werden.

Ein weiteres Konzept für Untersuchungen der Messunsicherheit des Abstandes bietet die Betrachtung des Einflusses der Messunsicherheit der detektierten Korrelationsfunktionen, welches im Simulator ebenfalls implementiert ist und bei Bedarf mit einer vordefinierten Standardabweichung für die einzelnen Messwerte der Korrelationsfunktionen hinzugeschaltet werden kann.

Der entwickelte Simulator stellt somit ein umfangreiches Werkzeug zur Simulation unterschiedlicher PMD-Kamera-Szenarien dar. Die dem Anwender gebotene große Flexibilität bietet ein hohes Potential besonders hinsichtlich Simulationen multistatischer Beleuchter-Sensor Anordnungen. Der Simulator stellt Test- und Referenzdaten zur Verfügung, mit deren Hilfe erweiterte Fehleranalysen durchgeführt, sowie Bildverarbeitungs- und Koregistrierungsansätze auf ihre Leistungsfähigkeit hin untersucht werden können. Die im Rahmen dieser Arbeit durchgeführten Testreihen und objektive Bewertungen der simulierten Daten bestätigen eine gute Übereinstimmung dieser mit den realen Messergebnissen.

10.2 Ausblick

Der aktuelle Entwicklungsstand und die umgesetzte Modellierung bieten eine gute Grundlage zu einer weitgehend flexiblen Simulation synthetischer Sensordaten. Während der Entwicklung und den Validierungsphasen haben sich jedoch einige weitere Fragestellungen und Verbesserungsmöglichkeiten ergeben.

Parallelisierung der Berechnungen

Der gesamte PMD-Simulator wurde mit dem Softwarepaket Matlab realisiert. In Bereichen der Vektor- und Matrizenrechnung bietet Matlab, im Vergleich zu anderen Programmiersprachen, eine recht gute Performance. Dennoch entstehen selbst bei mittleren Auflösungen des zu simulierenden Sensors, einer üblichen Konstellation der Lichtdioden und einer nicht sehr komplexen Objektumgebung eine enorme Anzahl an möglichen Interaktionen, die eine noch viel größere Anzahl oft sequentiell aufgebauter Berechnungen mit sich nachzieht. Dies spiegelt sich in hohen benötigten Rechenzeiten von typischerweise einigen Minuten für ein Frame wieder. Der größte Teil der Berechnun-

gen (>90%) fällt auf den Bereich der Sichtbarkeitsanalyse und lässt sich prinzipiell mit Hilfe der Berechnungen mittels eines Grafikchips (GPU) deutlich beschleunigen. Das rechenaufwendige Rendern ließe sich dann auf aktuellen Grafikkarten in Echtzeit durchführen. Das eingesetzte *Visibility Check Module* kann dann durch eine hocheffiziente Rendering-Komponente, realisiert z.B. in OpenGL, ersetzt werden, wodurch der rechenzeitliche Aufwand enorm reduziert werden kann. Ein noch weiter gehender Schritt wäre die Verwendung eines Grafikprozessors auch für weitere zeitintensive Berechnungen. In Laufe der letzten Jahre wurden Grafikchips im immer größeren Umfang programmierbar. Die Frage der Echtzeitfähigkeit stand bei der Entwicklung des PMD-Simulators nicht im Vordergrund. Die früheren Versionen von Matlab unterstützten außerdem keinen direkten Zugriff auf die von einer Grafikkarte gerenderten Daten. Diese Teilaufgabe wurde daher auf die CPU umgelegt. Im Hinblick auf die neuen Entwicklungen im Bereich GPGPU (General Purpose Computing on GPUs – „allgemeine Berechnungen auf Grafikprozessoren“) wird jedoch die Idee der Auslagerung der rechenintensiven Kernel auf den Grafikprozessor immer attraktiver.

Erweiterung des Kameramodells

Das im Simulator zugrunde liegende Modell der Kamera und des darauf aufbauenden Abbildungsvorgangs ist das Lochkameramodell. Dieses bietet entscheidende Vorteile bei der Modellierung der Optik, weil sich dadurch einfache Abbildungsvorschriften ergeben. Für kleine Blenden stellt diese Modellierung mit den implementierten radialen und tangentialen optischen Verzeichnungen eine recht gute Näherung für die Abbildung eines realen Optiksystems dar. Bei lichtstärkeren Objektiven, größeren Blenden und besonders bei Weitwinkelobjektiven kann das verwendete Modell unter Umständen eine ausreichende Genauigkeit der Abbildung nicht mehr gewährleisten. In solchen Fällen kann das Abbildungsmodell der Optik erweitert bzw. ersetzt werden (wird im Anhang ausführlicher behandelt). Die Simulation der Optik beruht im aktuellen Stand auf der Kenntnis des in die Optik einfallenden Strahlwinkels. Dazu wird der entsprechende austretende Strahl berechnet. Dieses Modul kann durch ein anderes auf dem gleichen Prinzip basierendes ausgetauscht werden.

Erstellung einer modellübergreifenden GUI

Der Simulator kann auf zwei Arten initialisiert und parametrisiert werden – mit Hilfe einer Initialisierungsdatei und mit Hilfe einer grafischen Benutzeroberfläche. Die Initialisierungsdatei muss alle Funktionalitäten und Parameter des Simulators berücksichtigen, die in der GUI definierten Inhalte können es, müssen aber nicht unbedingt. Programmtechnisch stellen die in der GUI gemachten Angaben nur Änderungen bezüglich der Initialisierungsdatei dar. Der Definitionsbereich der GUI ist also eine Art Editor für die Initialisierungsdatei. Im aktuellen Entwicklungszustand sind die meisten Parameter

der Kamera durch die GUI definierbar bzw. aktualisierbar. Sie bietet jedoch nicht den vollen Zugriff auf alle Funktionalitäten. So können z.B. die gewünschten Codecs und die Parameter für die Erstellung einer Videodatei der Simulation oder die grafische Ausgabe bestimmter Inhalte nur in der Initialisierungsdatei bzw. nur im Quellcode definiert werden. Über die GUI hat man den Zugriff nur auf die typischen Inhalte wie Simulationsszene, Abbildung der Szene auf dem Sensor, Tiefen und Amplitudenbilder und ein paar weitere. In einem weiteren Entwicklungsprozess wäre eine Erweiterung der einstellbaren Funktionalitäten sowie grafischer Ausgaben in der GUI denkbar.

Evaluation und Erweiterung des PMD-Simulators

Wie bereits geschrieben, ergeben sich gute Übereinstimmungen zwischen realen Aufnahmen und Simulationen besonders bei statischen Szenen. Bei Mittelung über einige Bilder kann in solchen Fällen auch das statistische Rauschen stark reduziert werden. Die Begutachtung bewegter Bilder gestaltet sich im Allgemeinen etwas schwieriger. Zum einen sind die einzelnen Bilder stärker verrauscht, zum anderen kann eine selbst einige wenige mm abweichende Simulationsumgebung besonders im Nahbereich zu Ergebnissen führen, die nicht mehr pixelgenau mit den gemessenen Werten korrespondieren. Bei schnellen Bewegungen kommen bei realen Kameras außerdem noch weitere Effekte wie Bewegungsartefakte, fliegende Pixel oder das so genannte „Pumpen“ der Kamera, bei dem im Nahbereich die Messwerte des gesamten Sensors von einem großen Objekt „mitgezogen“ werden, obwohl die Kamera noch nicht im Sättigungsbereich arbeitet, hinzu. Zur Erklärung des Ursprungs dieser Phänomene müssen weitere umfangreiche Tests und Simulationen durchgeführt werden. Der bestehende Simulator kann dann entsprechend der gewonnenen Erkenntnisse optimiert und erweitert werden.

Simulation der Sättigungseffekte

Die durchgeführten Messversuche zeigen eine deutliche Abhängigkeit der Messunsicherheit von der Signalstärke und dem Sättigungszustand des Sensors. Zu kurze Integrationszeiten oder schlecht reflektierende Materialien führen zu einer Verschlechterung der Signalstärke und somit zu einem höheren Rauschen der Abstandswerte. Bei zu langen Integrationszeiten tritt dagegen das Problem der Sättigung des Sensors auf. Bei Sensoren mit einer integrierten SBI-Schaltung (Unterdrückung der Hintergrundbeleuchtung) kann die Dynamik des Sensors zwar deutlich erhöht werden, aber auch sie hat ihre Grenzen. Trifft dann zu viel Licht auf den Sensor auf, sind die betroffenen Pixel stark verrauscht. Die Modellierung und Berücksichtigung dieses Phänomens kann in weiteren Entwicklungsstufen in Betracht gezogen werden.

11 Literaturverzeichnis

[Alex 01]	Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D. and Silva, C. T. „Point set surfaces” In IEEE Visualization, pages 21–28, October 2001
[Atki 96]	Atkinson, K. B. „Close Range Photogrammetry and Machine Vision” Whittle Publishing, 1996
[Baum 75]	Baumgart, B. G. „Winged-Edge Polyhedron Representation for Computer Vision” National Computer Conference, May 1975, Stanford University
[Bron 00]	Bronstein I. N., Semendjajew K. A., Musiol G. und Mühligh H. „ Taschenbuch der Mathematik ” Verlag Harri Deutsch, 5. Auflage, 2000, ISBN-10: 3817120052
[BuHe 01]	Buxbaum, B., Hess, H., Ringbeck, T., Schwarte, R. „The Photonic Mixer Device: New perspectives in optical wireless communication” Workshop "Optical MEMS and Integrated Optics", Dortmund, 2001
[BuSw 00]	Buxbaum, B., Schwarte, R., Schneider, B., Heß, H., Ringbeck, T., Tai, W., Grothof, M. „Optisches CDMA auf Basis der PMD-Technologie - Ein für unkorrelierte Datensignale blinder Empfänger“ ORT 2000, Hagen, 2000
[BuSw 01]	Buxbaum, B., Schwarte, R., Ringbeck, T., Grothof, M. „MSM-PMD as correlation receiver in a new multidimensional ranging system” SPIE- Laser Radar Techniques, Remote Sensing 2001, Toulouse, 2001
[BuSw 99]	Buxbaum, B., Schwarte, R. „Charge Transfer Simulation in PMDStructures” “Sensor’99 - Sensors, Transducers & Systems”, Vol. 2, S. 427-432, Nürnberg, 1999
[BuSw 99a]	Buxbaum, B., Schwarte, R. „Konzept einer hochparallelen optischen Schnittstelle auf Basis der PMD (Photonic Mixer Device)-Technologie“ 4. Workshop – Optik in der Rechentchnik, Jena, 1999
[BuSw 99b]	Buxbaum, B., Schwarte, R., Ringbeck, T. „PMD-PLL Receiver Structure for Incoherent Communication and

- Ranging Systems*
SPIE, Photonic East – Optical Wireless Communications II, Boston, 1999
- [BuSw 99c] Buxbaum, B., Schwarte, R., Xu, Z, Heinol, H., Ringbeck, T., Schulte, J., Tai, W.
„Charge Transfer Simulation in PMD-Structures”
Sensor 99 Proceedings, S. 427-432, Nürnberg, 1999
- [Ebel 92] Ebeling, K. J.
„Integrierte Optoelektronik“
Springer Verlag, Berlin, Deutschland, 1992
- [Graf 07] Graf, S.
„Kamerakalibrierung mit radialer Verzeichnung – die radiale essentielle Matrix“
Dissertation, Universität Passau, 2007
- [GrSw 02] Grothof, M., Schwarte, R., Buxbaum, B., Ringbeck, T., Frey, J.
„Product Monitoring with MSM-PMD-Array”
Kongressband OPTO'2002, AMA Fachverband für Sensorik, Erfurt, 2002
- [Hain 01] Heinol, H. G.
„Untersuchung und Entwicklung von modulationslaufzeitbasierten 3D-Sichtsystemen“
Dissertation, Siegen, 2001
- [Hart 05] Hartmann, K., Loffeld, O., Ghobadi, S. E., Peters, V., Prasad, T. D. A., Sluiter, A., Weihs, W., Lerch, T., Lottner, O.
„Klassifizierungsaspekte bei der 3D-Szenenexploration mit einer neuen 2D/3D-Multichip-Kamera“
19. Fachgespräch Autonome Mobile Systeme 2005 (AMS2005), Springer Verlag
- [Haso 06] Hasouneh, F., Knedlik, S., Peters, V., Loffeld, O.
„PMD-based Mobile Node Position Monitoring”
IEEE/ION PLANS2006 (Position Location and Navigation Symposium), April 24-27, San Diego, California, 2006
- [HeAl 02] H. Heß, M. Albrecht, R. Schwarte
„PMD - New Detector for Fluorescence Lifetime Measurement”
Kongressband OPTO'2002, AMA Fachverband für Sensorik, Erfurt, 2002
- [Hecht 89] Hecht, E.
„Optik“
Addison Wesley (Deutschland) GmbH, Bonn, Deutschland, 1989
- [Heri 05] Hering, E., Stohrer, M., Martin, R.
„Taschenbuch der Mathematik und Physik“

- Springer Verlag, Berlin, Auflage 4, 2005
- [Heri 08] Hering, E., Martin, R., Stohrer, M.
„*Physik für Ingenieure*“
Springer Verlag, Berlin, Auflage 10, 2008
- [Heß 06] Heß, H.
„*Empfang und Auswertung intensitätsmodulierter optischer Signale mittels Photonic-Mixer-Device (PMD) in Applikationen der Messtechnik und Kommunikation*“
Dissertation, Siegen, 2006
- [HeSw 99] Heinol, H.G., Schwarte, R.
„*Photomischdetektor erfasst 3D-Bilder*“
Elektronik 12/99; WEKA Fachzeitschriften Verlag, S. 80-90, München, 1999
- [HeXu 06] Heinol, H., Xu, Z., Schulte, J.
„*Laufzeitbasierte 3D-Kamerasysteme - Smart - Pixel -Lösungen*“
Tagungsband DGZfP/GMA – Optische Formerfassung, VDI/VDE-GMA, Stuttgart, 1999
- [HiDa 01] Hildebrand, M., David, K., Buxbaum, B., Grothof, M., Schwarte, R.
„*A New Approach for Indoor Wireless Internet based on Diffuse Infrared*“
Proceedings of the IASTED "Wireless and Optical Communications 2001" Conference, ACTA Press, Anaheim, Calgary, Zürich, ISBN 0-88986-287-7, 2001
- [Huß 00] Hußmann, S.
„*Schnelle 3D-Objektvermessung mittels PMD/CMOS-Kombizeilensensor und Signalkompressions-Hardware*“
Dissertation, Siegen, 2000
- [Jähne 05] Jähne, B.
„*Digitale Bildverarbeitung*“
Springer Verlag, Berlin, 2005
- [Kühl 07] Kühlke, D.
„*Optik. Grundlagen und Anwendungen*“
Verlag: Verlag Harri Deutsch GmbH, 2007
- [Lang 00] Lange, R.
„*3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*“
Dissertation, Siegen, 2000
- [LaSe 99] Lange, R., Seitz, P., Schwarte, R.
„*Time-of-flight-Entfernungskamera in CMOS/CCD-Technik mit pixelintegrierten Lock-in-Verstärkern*“
Tagungsband DGZfP/GMA – Optische Formerfassung, VDI/VDE-

- GMA, Stuttgart, 1999
- [LaSe 99] Lange R., Seitz, P., Schwarte, R.
„*Time-of-flight range imaging with a custom solid-state image sensor*”
SPIE-EOS: Sensors, Sensor Systems, and Sensor Data Processing,
München, 1999
- [Lied 03] Liedtke C.–E.
„*Rechnergestützte Szenenanalyse*“
Skript zur Vorlesung, Institut für Theoretische Nachrichtentechnik
und Informationsverarbeitung, Universität Hannover, 2003
- [Lott 10] Lottner, O.
„*Evaluation of Different Sensor / Sensor – and Illumination Configurations also in Dynamic Environments*”
Dissertation (Draft), Siegen, 2010
- [Luan 01] Luan, X.
„*Experimental Investigation of Photonic Mixer Device and Development of TOF 3D Ranging Systems Based on PMD Technology*“
Dissertation, Siegen, 2001
- [LuSw 01] Luan, X., Schwarte, R., Zhang, Z., Buxbaum, B., Ringbeck, T.,
Hess, H.
„*3D intelligent sensing based on PMD technology*”
SPIE- Laser Radar Techniques, Remote Sensing 2001, Toulouse,
2001
- [McCu 80] McCullagh, M. J., Ross, Ch. G.
„*Delaunay triangulation of a random data set for isarithmic mapping*”
The Cartographic Journal, 17(2): 93-99, 1980.
- [Misc 07] Mischler, A.
„*3D Reconstruction from Depth and Stereo Images for Augmented Reality Applications*”
Diplomarbeit, Technische Universität Berlin, 2007
- [NeRa 11] Nesper, S., Ranft, E.
„*Chancen und Herausforderungen von ToF-Kameras mit abgesetzten Lichtquellen*“
http://www.jade-hs.de/fileadmin/forschung/downloads/3DTage/Vortragsfolien_2011/Nesper.pdf
Oldenburger 3D-Tage 2011
- [Ohta 03] Ohtake, Y., Belyaev, A., Alexa, M., Seidel H.-P.
„*Multi-level Partition of Unity Implicits*”
ACM Transactions on Computer Graphics (SIGGRAPH 2003 Proceedings), pages 463-470, 2003

- [Pare 90] Pareigis, B.
„Analytische und projektive Geometrie für die Computer-Graphik“
Teubner, 1990
- [Pest 11] Pester, M.
„Mathematische Grundlagen der Computergeometrie“
Vorlesung Comutergeometrie, Technische Universität Chemnitz,
2011
- [Pet 06] Peters, V., Hasouneh, F., Knedlik, S., Loffeld, O.
„Simulation of PMD-based Self-localization of Mobile Sensor
Nodes or Robots”
ASIM2006 (19. Symposium Simulationstechnik), September 12-14,
Hanover, Germany, 2006
- [Pet 06] Peters, V., Knedlik, S., Loffeld, O.
„Modelbildung und Simulation eines hochauflösenden 3D-Sensors
am Beispiel einer PMD-Kamera“
3D-NordOst 2006, Berlin, Germany, Dec. 01, 2006
- [Pet 07a] Peters, V., Loffeld, O.
„A Bistatic Simulation Approach for a High Resolution 3D PMD-
Camera”
DAGM 2007 (29th Annual Symposium of the German Association
for Pattern Recognition) - Workshop on Dynamic 3D Imaging, Hei-
delberg, Germany, Sept. 11, 2007
- [Pet 07b] Peters, V., Loffeld, O., Hartmann, K., Knedlik, S.
„Modeling and Bistatic Simulation of a High Resolution 3D PMD-
Camera”
EUROSIM 2007 (6th EUROSIM Congress on Modelling and Simu-
lation), Ljubljana, Slovenia, Sept. 9-13, 2007
- [Pet 10a] Peters, V., Loffeld, O., Meckel, S., Lottner, O.
„Modeling and Simulation of Light Propagation for Multistatic 3D
PMD-Camera Illuminator Constellations”
EUROSIM 2010 (7th EUROSIM Congress on Modelling and Simu-
lation), Prague, Czech Republic, Sept. 6-10, 2010
- [Pet 10b] Peters, V., Loffeld, O., Meckel, S.
„Modular PMD Simulator for Multistatic Camera Systems”
EUROSIM 2010 (7th EUROSIM Congress on Modelling and Simu-
lation), Prague, Czech Republic, Sept. 6-10, 2010
- [RiBu 99] Ringbeck, T., Buxbaum, B.
„Layoutoptimierte PMD-Strukturen für die schnelle 3D-Erfassung“
Tagungsband DGZfP/GMA – Optische Formerfassung, VDI/VDE-
GMA, Stuttgart, 1999
- [Ring 07] Ringbeck, T., Hagebeuker, B.
„A 3D time of flight camera for object detection”

- Optical 3D Measurement Techniques, ETH Zürich, 2007
- [RiSw 01a] Ringbeck, T., Schwarte, R., Buxbaum, B., Luan, X., Tai, T., Zhang, Z.
 „A new Opto-electrical Phase Locked Loop for Phase / Frequency Detection in Rotation and Range Measurement Application”
 Prodeedings Sensor 2001, Nürnberg, 2001
- [RiSw 01b] Ringbeck, T., Schwarte, R., Buxbaum, B., Heß, H.
 „An Optical GMSK modem for Infrared Wireless Communication Based on a New Receiver Principle in CMOS-Technology”
 Photonic West - Optical Interconnects, San Jose, 2001
- [RiSw 99] Ringbeck, T., Schwarte, R., Buxbaum, B.
 „Introduction of a new electrooptical phase locked loop in CMOS-technology (PMD-PLL)”
 SPIE, Photonic East – Optical Wireless Communications II, Boston, 1999
- [Roth 03] Roth, H., Schwarte, R., Ruangpayoongsak, N., Kuhle, J., Albrecht, M., Grothof, M., Heß, H.
 „3D Vision Based on PMD-Technology for Mobile Robots”
 Proceedings SPIE Aerosense Conference 2003, Volume "Unmanned Ground Vehilce Technology", Paper 5083-66, Orlando, Florida, 2003
- [SchHe 99] Schwarte, R., Heinol, H., Buxbaum, B., Ringbeck, T., Xu, Z., and Hartmann, K.
 „Principles of Three-Dimensional Imaging Techniques”
 Handbook of Computer Vision and Applications, Sensors and Imaging, 1999
- [Schn 03] Schneider, B.
 „Der Photomischdetektor zur schnellen 3D-Vermessung für Sicherheitssysteme und zur Informationsübertragung im Automobil”
 Dissertation, Siegen, 2003
- [Schw 96] Schwarte, R.
 „Eine neuartige 3D-Kamera auf der Basis eines 2D Gegentaktkorrelator-Arrays”
 Symposium „Aktuelle Entwicklungen und industrieller Einsatz der Bildverarbeitung“, MIT GmbH, S. 273-278, Aachen 1996
- [Schw 98] Schwarte, R., Heinol, H., Buxbaum, B., Xu, Z., Ringbeck, T.
 „Neuartige 3D Visionsysteme auf der Basis Layout-optimierter PMD-Strukturen”
 tm-Technisches Messen 65 (1998); H. 7-8; Seite 264-271; 1998
- [Schw 99a] Schwarte, R.
 „3D-Bildsensorik”
 Tagungsband „10. Heidelberger Bildvearbeitungsforum“, Schwer-

- punktthema: 3D-Bildaufnahme und -verarbeitung‘; Siegen; 1999
- [Schw 99b] Schwarte, R.
„*Ein Jahrzehnt 3D-Bilderfassung – Zielsetzungen, Fortschritte, Zukunftsperspektiven*“
Tagungsband ‘Optische Formerfassung’, DGFfPVDI/VDE-GMA, GMA-Bericht 70, Beiheft; 1999
- [Stöck 98] Stöcker, H.
„*Taschenbuch der Physik*“
Verlag Harri Deutsch, Thun und Frankfurt am Main, 1998
- [SwBu 03] Schwarte, R., Buxbaum, B., Ringbeck, T., Xu, Z., Gollewski, T.
„*3-D-Kamera für Mobile Telerobotik und Telemanipulation*“
Telematik 2003, VDI-Berichte 1785, Fachtagung, Juni 2003, Siegen, ISBN 3-18-091785-7, S. 105 – 114
- [SwHe 00] Schwarte, R., Heinol, H., Xu, Z., Buxbaum, B., Schulte, J.
„*A New powerful sensory tool in Automotive Safety Systems Based on Photodetector Inherent Mixing by PMD*“
AMAA 2000, Berlin, Germany, 2000
- [SwHe 97] Schwarte, R., Heinol, H. G., Xu, Z., Olk, J., Tai, W.
„*Schnelle und einfache 3D Formerfassung mit einem neuartigen Korrelations-Photodetektor-Array*“
Tagungsband Optische Formerfassung, DGZfP und VDI-GMA, Langen 1997
- [SwHe 99] Schwarte, R., Heinol, H., Buxbaum, B., Ringbeck, T., Xu, Z., Hartmann, K.
„*Principles of 3-D Imaging Techniques*“
Computer Vision and Applications, Volume 1, Sensors and Imaging, Academic Press, S. 374-393, Boston 1999
- [SwXu 96] Schwarte, R., Xu, Z., Heinol, H.G.
„*Large Aperture Optical Modulators/- Demodulators for 3D-Cameras*“
OPTO 96 Kongressband, AMA Fachverband für Sensorik, S. 273-278, Leipzig 1996
- [Wang 08] Wang, J., Shi, F., Zhang, J., Liu, Y.
„*A new calibration model of camera lens distortion*“
Pattern Recognition 41, 2008, pp. 607–615
- [Win 09] Winner, H., Hakuli, S., Wolf, G.
„*Handbuch Fahrerassistenzsysteme Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*“
Verlag: Vieweg+Teubner, 2009
- [Xu 05] Xu, Z., Schwarte, R., Heinol, H., Buxbaum, B. and Ringbeck, T.
„*Smart pixel – photonic mixer device (PMD) new system concept of*

-
- a 3d-imaging camera-on-a-chip*
Technical report, PMDTec, 2005
- [Xu 99] Xu, Z.
„Investigation of 3D-Imaging Systems based on Modulated Light and Optical RF-Interferometrie (ORFI)“
Dissertation, INV, Universität Siegen, 1999
- [Zha 00] Zhang, Z.
„A flexible new technique for camera calibration“
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, 2000
- [Zha 03] Zhang, Z.
„Untersuchung und Charakterisierung von PMD (Photomischdetektor)-Strukturen und ihren Grundsaltungen“
Dissertation, Siegen, 2003
- [Zha 99] Zhang, Z.
„Flexible Camera Calibration By Viewing a Plane From Unknown Orientations“
International Conference on Computer Vision (ICCV'99), Corfu, Greece, pp. 666-673, 1999
- [ZhHe 02] Zhang, Z., Heß, H., Ringbeck, T., Schwarte, R.
„Phase-Lock techniques using new correlation receivers for TOF-laser ranging“
SPIE- Aero Sense 2002, Orlando, 2002
-

12 Anhang

12.1 Alternativer Ansatz zur Berechnung der Pixelzugehörigkeit eines Punktes

Alternativ zur Einführung eines Sensorkoordinatensystems ist es möglich, die Koordinaten der Pixelmatrix und die abgebildete Szene direkt im kamerafesten Koordinatensystem zu behandeln. Dieser Ansatz wird im Folgenden kurz erläutert.

Fall 1 (Idealfall): Die Pixelmatrix ist bezüglich des KKS nur verschoben

Betrachtet man die Relation der Sensorfläche zur Optik, so hat diese im Idealfall (Einbau des Sensors senkrecht zur optischen Achse, mit der Mitte auf der Achse und ohne Verdrehungen) eine feste Verschiebung d zum Ursprung des KKS, also zur Mitte der Optik, und zwar in die entgegengesetzte Richtung zur Achse \vec{z}_{KKS} (siehe Abbildung 12.1).

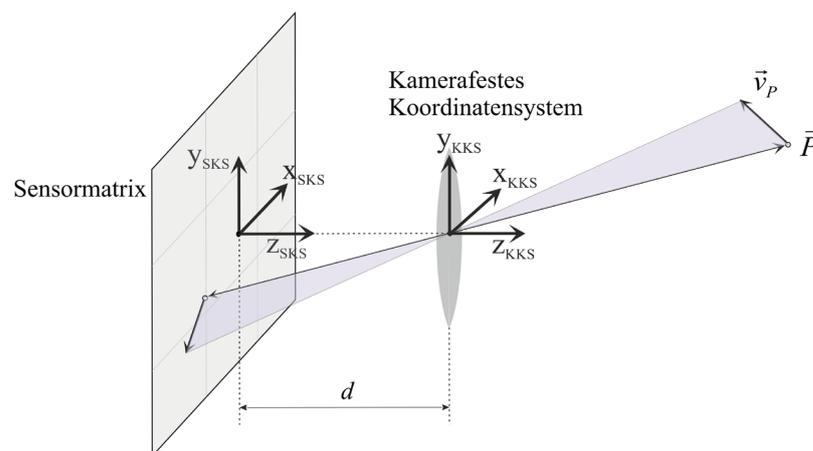


Abbildung 12.1: Abbildung eines Punktes im KKS auf die Pixelmatrix, die bezüglich des KKS nur verschoben ist

In so einem Fall kann ein im KKS gegebenes Punktziel mit Hilfe des Strahlensatzes, auf die Ebene, die durch die Sensorfläche aufgespannt wird, abgebildet werden. Danach kann mit dem gleichen Ansatz, wie bereits in Abschnitt 5.6 vorgestellt, das zu dem Punkt \vec{P} korrespondierende Pixel mit der Kenntnis der Sensorgröße und der Größe eines Pixels bestimmt werden. Da die z-Koordinaten aller Pixelknoten im KKS den gleichen Wert $-d$ haben und die restlichen Koordinaten gleich mit denen des Sensorkoordinatensystems sind, kann in einem solchen Fall auf die bei der Berechnung sehr oft vorkommende Transformation zwischen KKS und SKS verzichtet werden.

Fall 2: Die Pixelmatrix ist bezüglich des KKS verdreht und evtl. verschoben

Nun wird der Fall betrachtet, bei dem die Sensormatrix gegenüber dem kamerafesten Koordinatensystem verdreht und verschoben ist (siehe Abbildung 12.2).

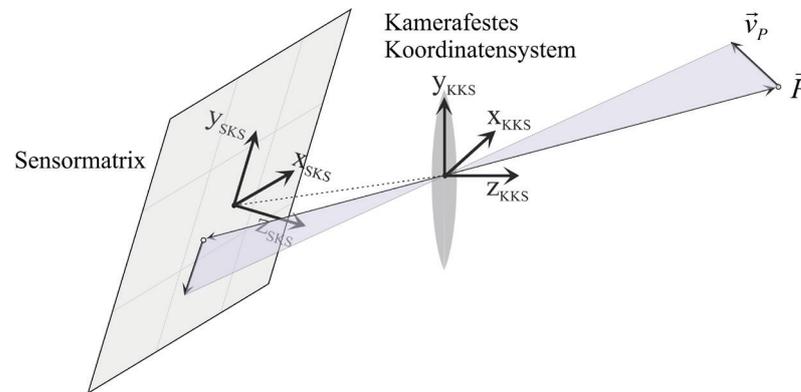


Abbildung 12.2: Abbildung eines Punktes im KKS auf die Pixelmatrix, die bezüglich des KKS verschoben und verdreht ist

Der Normalvektor auf die Sensorfläche liegt somit nicht parallel zur Achse Z_{KKS} . Betrachtet man die Ecken eines beliebigen Pixels, dann liegen alle vier Ecken in der von der Sensorfläche aufgespannten Ebene. Bei der Abbildung einer im Weltkoordinatensystem gegebenen Objektszene ins Kamerakoordinatensystem müssen alle sichtbaren auf die Sensorfläche abgebildeten Punkte den entsprechenden Pixeln zugewiesen werden. Betrachtet man einen einzigen Punkt im Raum und bildet diesen auf die Sensorebene ab, dann kann die Fragestellung in drei Schritte unterteilt werden:

- Liegt die Projektion des Punktes innerhalb des Sensors? Wenn „nein“, dann ist der Punkt nicht detektierbar, und wenn „ja“, dann ...
 - Auf welchem Pixel liegt genau die Projektion?
 - Liegt die Projektion im lichtempfindlichen Pixelbereich?

Im ersten und zweiten Schritt kann die Aufgabenstellung geometrisch betrachtet auf die Lösung folgender Fragestellung zurückgeführt werden:

Gegeben ist ein Pixel/Sensor-Viereck, welches durch die Eckpunkte $\vec{E}_1, \vec{E}_2, \vec{E}_3$ und \vec{E}_4 im kamerafesten Koordinatensystem gegeben ist. \vec{r}_{SP} ist der Schnittpunkt des Projektionsvektors des Punktes \vec{P} auf die Ebene, die mit dem Pixel/Sensor aufgespannt wird. Alle Punkte liegen somit auf einer Ebene. Es soll berechnet werden, ob die Projektion \vec{r}_{SP} innerhalb des Pixels/Sensors liegt oder außerhalb.

Einer der möglichen Lösungsansätze ist die Betrachtung der Flächen der Dreiecke, die mit je einer Kante des Pixels/Sensors und dem Schnittpunkt gebildet werden können (siehe Abbildung 12.3).

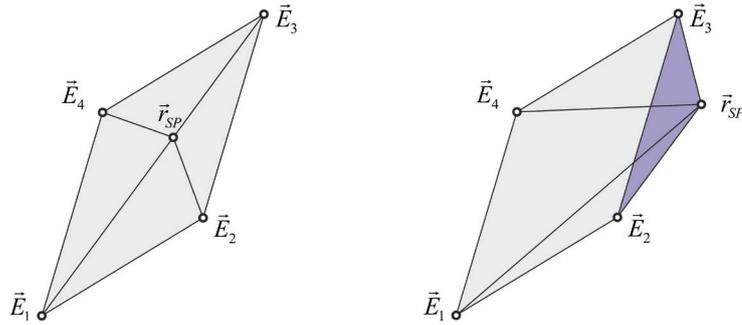


Abbildung 12.3: Veranschaulichung des Prinzip des Flächenvergleichs

Entspricht die Summe der Flächen der einzelnen Dreiecke der Fläche des Pixels/Sensors, dann muss der Schnittpunkt innerhalb des Pixels/Sensors liegen. Liegt der Schnittpunkt außerhalb des Pixels/Sensors, dann wird die Summe der Dreiecke größer sein. Die Abfrage lautet somit:

$$A(\vec{E}_1, \vec{E}_2, \vec{r}_{sp}) + A(\vec{E}_2, \vec{E}_3, \vec{r}_{sp}) + A(\vec{E}_3, \vec{E}_4, \vec{r}_{sp}) + A(\vec{E}_1, \vec{E}_4, \vec{r}_{sp}) \stackrel{?}{=} A(\vec{E}_1, \vec{E}_2, \vec{E}_3, \vec{E}_4) \quad (12.1)$$

Für die Flächeninhalte der Dreiecke mit den Eckpunkten \vec{E}_i , \vec{E}_j und \vec{r}_{sp} folgt mit Hilfe der Vektoralgebra:

$$A(\vec{r}_{sp}, \vec{E}_i, \vec{E}_j) = \frac{1}{2} \sqrt{\left| \left[(\vec{E}_j - \vec{E}_i) \times (\vec{r}_{sp} - \vec{E}_i) \right] \cdot \left[(\vec{E}_j - \vec{E}_i) \times (\vec{r}_{sp} - \vec{E}_i) \right] \right|} \quad (12.2)$$

Der entscheidende Vorteil dieser Methode liegt in seiner einfachen Umsetzung. Die Berechnung kann quasi in jedem orthogonalen Koordinatensystem durchgeführt werden. Der Nachteil verbirgt sich jedoch in der schrittweise aufgebauten Abfrage, die noch dazu für jedes einzelne Pixel der Matrix durchgeführt werden soll. Mit Hilfe der iterativ aufgebauten Methoden, z.B. indem man die Pixelfläche immer in zwei Hälften aufteilt und nach dem Ausschlussverfahren vorgeht, lässt sich die Anzahl der durchzuführenden Schritte auf $k+1$ reduzieren, mit $2^k \geq \text{Anzahl der Pixel}$ ⁷. Dennoch werden bei einer Kameraauflösung von 160x120 Pixel bereits 16 Schritte benötigt, um das gesuchte Pixel zu finden. Durch das Einsetzen effizienterer Iterationsverfahren ließe sich die Anzahl der benötigten Abfragen im Schnitt deutlich reduzieren, die würden dennoch an die Effizienz der durch die Transformation erzielten einfachen Lösung nicht herankommen. Ein weiteres Problem wäre (vgl. Abschnitt 5.3) die geometrisch gesehen nicht triviale Lösung bei ellipsenförmigen aktiven Bereichen des Pixels.

Zusammenfassend lässt sich sagen, dass der hier präsentierte alternative Ansatz nur in dem Fall sinnvoll und schneller ist, wenn die Pixelfläche senkrecht zur optischen Achse definiert ist. In allen anderen Fällen wird der in Kapitel 5.4 behandelte Ansatz empfohlen.

⁷ Bei Definition eines rechteckförmigen aktiven Bereichs des Pixels wird noch ein weiterer Schritt benötigt.

12.2 Homogene Koordinaten

Eine im Gegensatz zu kartesischen Koordinaten in der Regel bessere und einfach zu handhabende Behandlung der Koordinatentransformationen bieten die in der Computergrafik oft zu findenden Transformationen in homogenen Koordinaten. Dabei bekommen die Koordinaten bzw. Vektoren in kartesischen Koordinaten eine homogene Erweiterung. Ein Punkt P mit den affinen Koordinaten (x, y, z) besitzt dann als homogene Koordinaten (x, y, z, w) mit $w = 1$ bzw. (wx, wy, wz, w) mit $w \neq 0$. Bei der Darstellung der Vektoren in homogenen Koordinaten wird $w = 0$ gewählt. Für einen Ortsvektor gilt somit [Pest 11], [Par 90]

$$\vec{P} = (x \quad y \quad z \quad 1)^T \quad (12.3)$$

Ein Punkt mit den homogenen Koordinaten $(x \quad y \quad z \quad 0)^T$ besitzt keine affinen Koordinaten, die Betrachtung als Grenzwert liefert:

$$(x \quad y \quad z \quad 0)^T = \lim_{n \rightarrow \infty} (x \quad y \quad z \quad n)^T = \lim_{n \rightarrow \infty} n \cdot (x \quad y \quad z)^T \quad (12.4)$$

Ein Vektor kann somit als ein unendlich entfernter Punkt in der entsprechenden Richtung aufgefasst werden. Für einen Richtungsvektor gilt also:

$$\vec{V} = (x \quad y \quad z \quad 0)^T \quad (12.5)$$

Der Vorteil der homogenen Koordinaten liegt darin, dass sich sowohl Rotationen, als auch Translationen und Skalierungen mit Hilfe einer einheitlichen Behandlung durchführen lassen. Mehrere Transformationen lassen sich durch Multiplikation entsprechender Transformationsoperatoren kombinieren. Die Transformation eines Vektors \vec{u} in kartesischen Koordinaten kann mit Hilfe der Rotationsmatrix R , des Translationsvektors \vec{t} und des Skalierungsvektors \vec{s} allgemein wie folgt ausgedrückt werden

$$\vec{u} = (R \cdot \vec{u} + \vec{t}) \circ \vec{s} \quad (12.6)$$

◦ $\hat{=}$ komponentenweise Multiplikation

wobei die Rotation um eine beliebige Achse als ein Produkt der Rotationen um die Koordinatenachsen angesehen werden kann:

$$R = \underline{R}_x \cdot \underline{R}_y \cdot \underline{R}_z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \cdot \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix} \cdot \begin{pmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} = \quad (12.7)$$

$$\begin{pmatrix} \cos \beta \cos \gamma & \cos \beta \sin \gamma & -\sin \beta \\ \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \cos \beta \\ \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \cos \beta \end{pmatrix}$$

$$\vec{t} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}, \quad \vec{s} = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} \quad (12.8)$$

In homogenen Koordinaten können dagegen alle Transformationen in einer einzigen Matrix zusammengefasst werden:

$$\vec{u} = T \cdot \vec{u} = T_{rotation} \cdot T_{trans} \cdot T_{scale} \cdot \vec{u} \quad (12.9)$$

Mit den entsprechenden homogenen Transformationsmatrizen

$$T_{rotation} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad T_{trans} = \begin{pmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{und} \quad T_{scale} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Dabei ist zu beachten, dass die Matrizen von links operieren, also von rechts nach links nacheinander auf den Vektor \vec{u} angewandt werden. Bei gleichmäßiger Skalierung $\Rightarrow s_x = s_y = s_z$.

Ein weiterer Vorteil der homogenen Koordinaten ist die einfache Realisierung der Rücktransformationen. So werden bei Rückbewegung und der Rückskalierung die Inversen der Transformationsmatrizen gebildet und bei inverser Rotation die negativen Winkel in die Transformationsmatrizen eingesetzt:

$$\text{Inverse Skalierung: } T_{scale}^{-1} = \begin{pmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Rückbewegung: } T_{trans}^{-1} = \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Inverse Rotation: Einsetzen von $\underline{R}_x(-\alpha)$, $\underline{R}_y(-\beta)$, $\underline{R}_z(-\gamma)$

Die einzelnen Transformationsoperatoren liegen in der gesamten Transformationsmatrix in den entsprechenden Bereichen der Matrix vor:

$$T = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} \begin{array}{l} \text{Skalierung} \\ \text{Rotation} \\ \text{Translation} \\ \text{Perspektivische Projektion} \\ \text{Homogene Erweiterung} \end{array}$$

Eine Rücktransformation von homogenen Koordinaten in kartesische Koordinaten, kann durch eine perspektivische Abbildung beschrieben werden.

$$\begin{pmatrix} x' \\ y' \\ z' \\ w \end{pmatrix} \rightarrow \begin{pmatrix} x'/w \\ y'/w \\ z'/w \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (12.10)$$

Bei der Durchführung der im Abschnitt 4.8 behandelten Koordinatentransformationen, werden diese im Simulator entsprechend der Vorgaben für einen Orts- oder Richtungsvektor homogenisiert. Nach der Durchführung der Koordinatentransformationen werden diese wieder in kartesische Koordinaten überführt.

12.3 Betrachtung von dünnen bikonvexen Linsen

Eine ideale Lochkamera bildet alle Objektpunkte in der Objektebene mit Hilfe einer perspektivischen Projektion durch das optische Zentrum der Kamera auf entsprechende Punkte in der Bildebene ab. Das so entstehende Bild ist unabhängig von der Entfernung immer scharf. Die physikalischen Grenzen eines solchen Systems liegen in seiner schlechten Realisierbarkeit. Das Problem dabei ist, dass Kameras zur Belichtung eine bestimmte Menge an Licht brauchen, welches durch ein Lochkameramodell gegen Null minimiert wird. In der Realität werden daher zur optischen Abbildung in Kameras fast ausschließlich Linsensysteme verwendet. Ein wesentlicher Teil davon kann in guter Näherung durch Abbildungseigenschaften dünner Linsen bzw. einer Reihe davon besser beschrieben werden.

Als eine dünne Linse wird eine Linse bezeichnet, deren Dicke im Verhältnis zu dem Radius der Linse vernachlässigbar ist. Das Konzept der dünnen Linse ist ebenfalls eine Idealisierung, welche in der Praxis nie vollständig erfüllt werden kann. Es bietet jedoch eine gute Näherung besonders für Systeme mit großer Brennweite [Stöck 98]. Konvexlinsen sind Linsen mit einer positiven Brennweite. Als Bikonvexlinsen bezeichnet man Linsen, deren beide im Allgemeinen sphärischen Flächen nach außen hin gekrümmt sind (man spricht dabei von einer positiven Krümmung). Sind die Krümmungsradien gleich, dann spricht man von einer symmetrischen bikonvexen Linse. Betrachtet man eine solche Linse, dann lässt sich die Konstruktion I eines Objektes O im Abstand g von

der Hauptebene der Linse in der Bildebene durch mindestens zwei der folgenden linearen Strahlengänge nachbilden (siehe Abbildung 12.4):

- Parallelstrahl: der Strahl parallel zur optischen Achse wird bei der Abbildung so gebrochen, dass er durch den Brennpunkt f geht.
- Mittelpunktstrahl: ein Strahl durch den Schnittpunkt der optischen Achse mit der Hauptebene der Linse wird nicht gebrochen.
- Brennpunktstrahl: ein Strahl durch den Brennpunkt wird von der Linse so gebrochen, dass seine Richtung parallel zur optischen Achse verläuft.

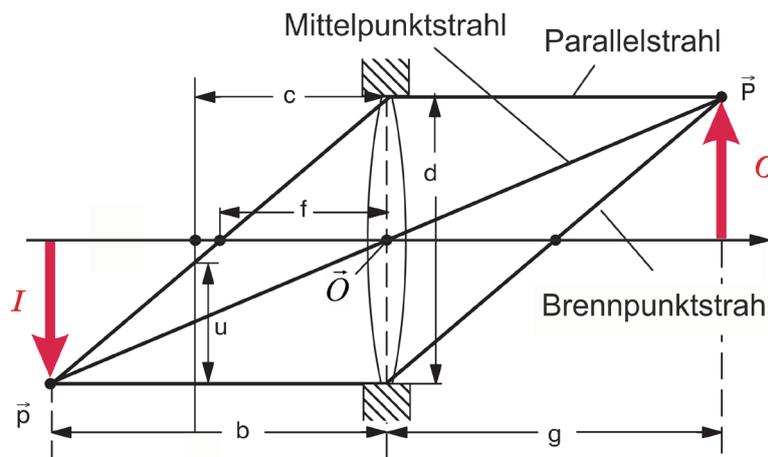


Abbildung 12.4: Konstruktion des Strahlengangs durch eine dünne Linse

Für dünne Linsen gilt näherungsweise die folgende Abbildungsvorschrift, die auch den Namen Gauß'sche Linsenformel trägt [Kühl 07]:

$$\frac{1}{f} = \frac{1}{b} + \frac{1}{g} \quad (12.11)$$

mit Objektweite g und Bildweite b

Für eine scharfe Abbildung muss diese Gleichung eingehalten werden. Eine Verschiebung des Objektes oder die Verschiebung der Kammerkonstante c (Abstand zwischen dem Projektionszentrum \vec{O} und seinem Lotfußpunkt auf die Sensorfläche) von der Bildweite b bewirkt eine Unschärfe bei der Abbildung. Ein Punkt im Raum wird dann nicht mehr auf einen Punkt auf dem Sensor, sondern auf eine kreisförmige Fläche mit dem Durchmesser u entsprechend Abbildung 12.4 abgebildet [Lied 03], [Hecht 89]. Liegt dieser Kreis innerhalb eines Pixels, dann spricht man von einer scharfen Abbildung. Neben der Geometrie entscheidet somit auch die Rasterung der Pixel, also ihre Größe, über den Schärfereich der Kamera. Feinere Pixel erfordern feinere Einstellung der Bildweite bzw. der Kammerkonstante.

Der Unschärfedurchmesser u lässt sich wie folgt bestimmen:

$$u = \frac{d}{f} \left[f - c \cdot \frac{g-f}{g} \right] = \frac{1}{k} \left[f - c \cdot \frac{g-f}{g} \right] \quad (12.12)$$

mit $k = \frac{f}{d}$ – Blendenzahl, d – Blende der Linse

Eine Vergrößerung der Blendenzahl bewirkt eine Erhöhung der Schärfentiefe, führt aber auf der anderen Seite zu einer Steigung der Unschärfe infolge der Beugungserscheinungen aufgrund der Wellennatur des Lichtes [Lied 03].

$$u_{th} = 2,44 \cdot k \cdot \lambda \quad (12.13)$$

mit Objektweite g und Bildweite b

Aufgrund der relativ einfachen geometrischen Zusammenhänge kann bei der Simulation der Tiefen- und Intensitätsbilder das zurückgestrahlte Licht entsprechend dem Unschärferradius auf die betroffenen Pixel „verteilt“ werden. Über die so abgeschwächte Signalstärke kann, entsprechend der Teilintensitäten von jedem einzelnen Punkt, über den Einfluss auf die Korrelationsabtwerte später die Messunsicherheit der Phasenbilder beeinflusst werden. Schwach beleuchtete Bereiche und Pixel erhalten dann eine höhere Messunsicherheit als stark beleuchtete Bereiche und Pixel.