



# Bachelorarbeit

Gerrit Bartsch

Kamera-Laserscanner-Sensorfusion für  
Kraftfahrzeuge

# **Gerrit Bartsch**

## Kamera-Laserscanner-Sensorfusion für Kraftfahrzeuge

Bachelorarbeit eingereicht im Rahmen Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Andreas Meisel  
Zweitgutachter : Prof. Dr. Franz Korf

Abgegeben am 26.04.2016

**Gerrit Bartsch**

**Thema der Bachelorarbeit**

Kamera-Laserscanner-Sensorfusion für Kraftfahrzeuge

**Stichworte**

Sensorfusion, Sensortechnik, Kamera, Laserscanner, Robot Vision, CoRE Projekt

**Kurzzusammenfassung**

Dieses Dokument beschäftigt sich mit einem Konzept zur Fusion von Sensordaten, für den Einsatz in Kraftfahrzeugen. Zwei Laserscanner zur Umgebungserfassung und eine Kamera dienen als Datenquellen. Die Kameradaten werden um die Informationen der Laserscanner erweitert. Für die Fusion der Sensordaten werden Techniken der Bildbearbeitung eingesetzt.

**Gerrit Bartsch**

**Title of the paper**

Camera-Laser Scanner-Sensor Fusion for automobiles

**Keywords**

Sensor Fusion, Sensor Technology, Camera, Laser Scanner, Robot Vision, CoRE Project

**Abstract**

This document deals with a concept for the fusion of sensor data, for use in automobiles. Two laser scanner for environment detection and a camera serve as data sources. The camera data is extended by the information of the laser scanner. For the fusion of the sensor data techniques of image processing are used.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>7</b>
<b>2</b>	<b>Der Systemaufbau .....</b>	<b>9</b>
2.1	Das Fahrzeug.....	9
2.1.1	Beschreibung.....	9
2.2	Die Kamera .....	10
2.2.1	Beschreibung.....	10
2.2.2	Technische Daten .....	10
2.3	Die Laserscanner.....	11
2.3.1	Beschreibung.....	11
2.3.2	Technische Daten .....	11
2.3.3	Format der Scanner-Daten.....	12
<b>3</b>	<b>Die Datenaufbereitung .....</b>	<b>15</b>
3.1	Unterstützende Programme und Softwarebibliotheken .....	15
3.2	Kameradaten .....	16
3.2.1	Transformationsansatz.....	17
3.2.2	Korrektur der Linsenverzeichnung.....	19
3.2.3	Mathematische Grundlagen zur Linsenverzeichnungskorrektur.....	21
3.3	Laserscandaten .....	24
3.3.1	Datenformat der Scaninformationen.....	24
3.3.2	Analyse der Scaninformationen .....	25
3.3.3	Grafische Darstellung der Scaninformationen.....	25
3.3.4	Mathematische Grundlagen zur Laserdatenanalyse .....	30

<b>4 Die Sensorfusion.....</b>	<b>32</b>
4.1 Perspektivische Transformation.....	32
4.2 Anwendung der Perspektivischen Transformation auf die Sensorfusion ....	34
4.3 Unterschiede in der Objektwahrnehmung zwischen Kamera und Laserscanner.....	35
4.4 Objektrelationen zwischen Kamera und Laserscanner .....	36
4.5 Konzepte verschiedener Messvorrichtungen.....	38
4.5.1 Aufbau 1 .....	38
4.5.2 Aufbau 2 .....	39
4.5.3 Aufbau 3 .....	39
4.5.4 Aufbau 4 .....	40
4.5.5 Auswertung.....	41
4.6 Umsetzung des Konzeptes.....	41
4.7 Lagebestimmung der Scanebenen .....	44
4.8 Durchführung der Sensorfusion .....	47
4.9 Mathematische Grundlagen zur Perspektivischen Transformation .....	50
<b>5 Zusammenfassung und Ausblick .....</b>	<b>51</b>
5.1 Zusammenfassung.....	51
5.2 Einsatzmöglichkeiten.....	52
5.3 Bedeutung für die Zukunft .....	53
<b>6 Begleitende Software .....</b>	<b>54</b>
6.1 Program Sensorfusion .....	54
6.1.1 Funktionen .....	54
6.1.2 Klassendiagramm .....	58
6.2 Modul Sensorfusion.....	59
6.2.1 Funktionen .....	59
6.2.2 Klassendiagramm .....	62
<b>7 Quellen .....</b>	<b>63</b>
7.1 Wissenschaftliche Quellen .....	63
7.2 Internetquellen.....	63
7.3 Buchquellen .....	64

7.4	Bildquellen .....	64
<b>8</b>	<b>Anhang.....</b>	<b>65</b>
8.1	Tabellen .....	65
8.1.1	Punktliste.....	65
8.1.2	Gleichungssysteme.....	66
8.1.3	Transformationsparameter.....	70

# 1 Einleitung

In der Automobiltechnik werden heute immer mehr technische Hilfsmittel zur Fahrerassistenz eingesetzt, mit dem Ziel Sicherheit und Fahrkomfort zu steigern.

In vielen modernen Kraftfahrzeugen werden Scanner- und Kamerasysteme installiert. Sie werden aber oft getrennt oder für unterschiedliche, nicht zusammenarbeitende Systeme eingesetzt. Ein gutes Beispiel ist hier die Kombination aus Rückfahrkamera und Abstandssensor. Beide Systeme dienen dem Fahrer als Parkhilfe aber die Daten des einen Systems werden nicht zur Erweiterung der Daten des anderen Systems verwendet.

Kameras bieten als Sensorelemente, in einem Fahrzeug installiert, weitreichende Einsatzmöglichkeiten. Es ist möglich Straßenschilder auszuwerten, Hindernisse oder Personen auf der Fahrtrute zu erkennen und dem Straßenverlauf zu folgen. Im einfachen Falle einer Rückfahrkamera dienen sie als Fahrerassistenz.

Um diese Möglichkeiten zu realisieren, ist es erforderlich mit einem Analyseprogramm im Kamerabild Straßenschilder oder Hindernisse zu finden und zu erkennen. Hierzu werden verschiedene Techniken der Mustererkennung und der Bildbearbeitung eingesetzt. Oft ist es notwendig mehrfach jedes einzelne Pixel eines Bildes zu bearbeiten. Bei hochauflösenden Bildern benötigen diese Bearbeitungsprozesse sogar mit den heutigen Computersystemen viel Rechenzeit.

Das Leistungsvermögen einer Kamera als Sensor ist somit abhängig von der eingesetzten Analyse- und Bildbearbeitungssoftware, den technischen Parametern des eingesetzten Aufnahmeapparates und der Rechenkapazität des Fahrzeugcomputers.

Laserscanner ermöglichen einem Fahrzeug eine recht genaue Tiefenabtastung seiner Umgebung. Die verschiedenen Modelle bieten unterschiedliche Abtaststraten, Erfassungsreichweiten und Auflösungen. Ein solcher Scanner erlaubt eine schnelle Erfassung von Objekten im Sensorbereich. Der Einsatz von Analysesoftware erlaubt hier die

Erkennung von Kollisionsgefahren, die Bestimmung des Abstandes zum Hindernis und auch die Messung von Relativgeschwindigkeiten zu bewegten Objekten.

Ein Laserscanner erstellt Abstandsinformationen, zwischen Scanner und Objekt, in einer horizontalen Schnittebene. Interpretiert man die Abstandsinformationen grafisch, erhält man eine skizzenhafte Draufsicht der Fahrzeugumgebung. Mit einem Laserscanner alleine kann man gut das Vorhandensein von Objekten detektieren. Es ist aber teilweise nur schwer möglich festzustellen um was für ein Objekt es sich genau handelt.

Eine Fusion dieser beiden unterschiedlichen Sensorsysteme würde einen Mehrwert der Sensordaten erlauben und eine Entlastung der jeweiligen Analysesoftware ermöglichen.

Für eine Bildauswertung könnte man so auf die Analyse des gesamten Bildes verzichten, da man die Objekte über den Laserscanner bereits gefunden hat. Im Bild entstehen so „Regions of Interest“ (ROI), welche bevorzugt ausgewertet werden können. Dies spart Rechenzeit, wodurch sich der praktische Einsatzbereich erweitert.

Weiterhin können Bilddaten die Laserdaten ergänzen. So gewinnt die Umgebungserfassung an Details, was eine bessere Interpretation der Sensordaten ermöglichen kann.

## 2 Der Systemaufbau

Die Quelldaten, welche für die Kamera-Laserscanner-Sensorfusion verwendet werden, entstammen einem realen Versuchsfahrzeug. Das Fahrzeug ist mit einer Vielzahl von Modifikationen an der Bordelektronik und im Speziellen dem internen Kommunikationsnetz versehen. Im Besonderen verfügt das Versuchsfahrzeug über ein Kamerasystem und über Laserscanner zur Umgebungserfassung. Das Fahrzeug ist mit einer internen Datenbank ausgestattet. Dies ermöglicht einen einfachen Zugriff auf Testdaten. Für die Kamera-Laserscanner-Sensorfusion werden die Rohdaten dieser Sensoren als Beispiel- und Arbeitsdaten verwendet.

### 2.1 Das Fahrzeug

#### 2.1.1 Beschreibung

Das Versuchsfahrzeug gehört zu der Communication over Real-Time Ethernet (CoRE) Arbeitsgruppe der HAW-Hamburg. Heutige Fahrzeuge verwenden unterschiedliche Hilfssysteme wie Stabilitätsprogramme (ESP), Einparkhilfen oder automatische Abstandsregelungen (ACC) um Sicherheit und Fahrerassistenz im Straßenverkehr zu erhöhen. Diese verwenden zurzeit unterschiedliche Kommunikationstechnologien, was Datenaustausch und Vernetzung erschwert.

Die CoRE-Forschungsgruppe arbeitet mit dem Versuchsfahrzeug an zukünftiger Vernetzungstechnik und Kommunikationsarchitektur in Automobilfahrzeugen. Das Ziel ist die einfache Integration neuer Funktionen und Assistenzsysteme. Um den zukünftigen Datenaustausch zu realisieren, werden diese neuen Kommunikationsarchitekturen benötigt.

In Zusammenarbeit mit dem Industriepartner IAV wurde in das Testfahrzeug ein echtzeitbasiertes Ethernet-Kommunikationsnetz installiert. Die Arbeiten am Versuchsfahrzeug und um die CoRE-Forschungsgruppe sind Gegenstand vieler Bachelor- und Masterarbeiten.

## 2.2 Die Kamera

### 2.2.1 Beschreibung

Das Fahrzeug ist an der Front und an dem Heck mit einer Kamera vom Type: USB UEye MI ausgerüstet. Die Frontkamera ist zusätzlich mit einem Fish-Eye Objektiv bestückt, welches den Bildbereich und damit die Bildqualität erhöht. Die Kamera wird mit einer Auflösung von 1280 x 1024 Pixeln bei ca. 9 – 10 fps betrieben. Für die Bearbeitung der auftretenden Linsenverzeichnung sind die technischen Daten der eigentlichen Sensorfläche wichtig.

### 2.2.2 Technische Daten

<b>Kamera</b>	
Model	UI-1240ML-C-HQ
Sensortechnologie	CMOS Color
Hersteller	e2v
Auflösung (MPixel)	1,31
Auflösung (h x v)	1280 x 1024
ADC	10 Bit
Farbtiefe (Kamera)	8 Bit
Auflösungsklasse	SXGA
Optische Klasse	1/1.8
max. fps im Freerun Modus	25,8
Binning Modi	M/C automatisch
Sensorbezeichnung	EV76C560ACT
Pixelgröße	5,3 µm
Optische Fläche	6,784 mm x 5,427 mm
8,69 mm (1/1,84")	8,69 mm (1/1,84")

<b>Objektiv</b>	
Modell	FISH-EYE LENSE FE185C086HA-1
Focal Length (mm)	2,7
Iris Lenght	F1,8 ~ F16
Operation (Focus)	Fixed
Operation (Iris)	Manual
Angel of View (1")	185° X 185° (Ø8,6mm)
Angel of View (2/3")	185° X 140° 35'

Angel of View (1/2")	185° X 102° 19'
Focusing Range (Front of the Lens) (m)	∞ ~ 0,2
Back Focal Distance (in air) (mm)	9,75
Exit Pupil Position (Front Image Plane) (mm)	-49

## 2.3 Die Laserscanner

### 2.3.1 Beschreibung

Das Fahrzeug verwendet zwei ibeo LUX (model 2010) Laserscanner um seine Umgebung zu erfassen. Die beiden Laserscanner sind in der Front des Fahrzeugs eingesetzt und erfassen je die linke bzw. die rechte vordere Umgebung des Fahrzeugs. Aufgrund dieser Sensoranordnung ist die Umgebungserfassung direkt vor dem Auto ungenau. Die Scanbereiche überlappen sich frontal mit steigender Entfernung.

Für die Fusion der Kamerabilder mit den Laserscanner-Daten müssen die Rohdaten des Scanners in ein bearbeitbares Format gebracht werden. Hierfür sind die technischen Spezifikationen des Scanners und der Aufbau der Rohdaten entscheidend.

### 2.3.2 Technische Daten

<u>Laserscanner</u>	
Modell	IBEO LUX (model 2010)
Scan Data type	0x2202
Range	200 m / 650 ft (average distance) 50 m / 164 ft @ 10 % remision
Horizontal field of view	2 layers: 110 deg (50 deg to -60 deg) 4 layers: 85 deg (35 deg to -50 deg)
Vertical field of view	3,2 deg
Multi layer	4 parallel scanning layers
Multi echo	Up to 3 distance measurements per laser pulse (allow measurements through atmospheric clutter like rain and dust)
Data update rate	12.5/ 25.0/ 50.0 Hz
Accuracy (distance independent)	10 cm/ 3.9 in
Angular resolution	Horizontal: up to 0.125 deg / Vertical: 0.8 deg
Distance resolution	4 cm/ 1.57 in

### 2.3.3 Format der Scanner-Daten

Die Laserscanner generieren bei jedem Abtastvorgang einen Datensatz. Diese Informationen werden in der Fahrzeugdatenbank abgelegt. Die Informationen in den einzelnen Datenfeldern des Datensatzes sind vom verwendeten Modell abhängig oder werden bei der Installation konfiguriert. Die Scanpunktliste und die Begleitinformationen wie die Punktzahl oder die Datensatzgröße werden vom dem Laserscanner erstellt und eingetragen.

#### Ibeo data header

(big-endian Format)

<u>Bytes</u>	<u>Offset</u>	<u>Ibeo data header</u>	<u>Data Type</u>	<u>Description</u>
4	0	Magic word (0xAFFEC0C2)	UINT32	The "magic word" is used for searching Ibeo messages and to distinguish between different versions
4	4	Size of previous messages	UINT32	Helps to navigate backwards through a file. Unused in live data
4	8	Size of this message	UINT32	Helps to read the message data. Size of message content without this header
1	12	Reserved	UINT8	
1	13	DeviceID	UINT8	ID of the connected device. Unused in data received directly from Ibeo LUX sensors
2	14	Data type	UINT16	Specifies the data type within this message
8	16	NTP time	NTP64	Time when this message was created
	24	Message data		Depending on data type

#### Ibeo LUX scan data: Data type 0x2202

(little-endian Format) Angles are given in the ISO 8855 / DIN 70000 scanner coordinate system

<u>Bytes</u>	<u>Offset</u>	<u>Ibeo data header</u>	<u>Data Type</u>	<u>Description</u>
2	0	Scan number	UINT16	The number of this scan. The number will be increased from scan to scan
2	2	Scanner status	bit field 16 bits	0x0007: reserved 0x0008: set frequency reached 0x0010: external sync signal detected 0x0020: sync ok 0x0040: sync master (instead of slave)

				0xFF80: reserved
2	4	Sync phase offset	UINT16	Phase difference (conversion factor 409.6 ns) between sync signal and scanner mirror crossing the synchronization angle
8	6	Sync phase offset	UINT16	NTP time when the first/last measurement was done
8	14	Scan end time NTP	NTP64	
2	22	Angle ticks per rotation	UINT16	Number of angle ticks per rotation
2	24	Start angle	INT16	Start/end angle in angle ticks of this scan
2	26	End angle	INT16	
2	28	Scan points	UINT16	Number of scan point transmitted in this scan
2	30	Mounting position yaw angle	INT16	Rotation of the scanner around the axes of the reference coordinate system. All angles are given in angle ticks Order of translation and rotation is essential: Yaw->Pitch->Roll->Translation Scan data is given in the scanner coordinate system without any transformation
2	32	Mounting position pitch angle	INT16	
2	34	Mounting position roll angle	INT16	
2	36	Mounting position x	INT16	Mounting position of the scanner relative to the reference coordinate system (ISO 8855 / DIN 70000 coordinate system). The origin is located on flat ground under the center of the rear axle. X-axis faces to the vehicle front resp. straight driving direction. Y-axis faces left The mounting position is needed for ego motion compensation (only available if scanner x-y-plane is almost parallel to the ground) All coordinates are given in centimeters. Order of translation and rotation is essential (Rotation -> Translation) The mounting position is used for ego motion compensation, not to transform scan data but is available for further processing steps
2	38	Mounting position y	INT16	
2	40	Mounting position z	INT16	
2	42	Flags	UINT16	Bit 0: ground labeled Bit 1: dirt labeled

				Bit 2: rain labeled Bit 3: reserved Bit 4..6: internal Bit 7..9: reserved Bit 10: mirror side (0=front, 1=rear) <sup>1</sup> Bit 11..15: reserved
	44	Scan Point List	Scan Point	Array of scan points. See number of scan points above and point information below

**ibeo LUX scan data: Scan Point List**

(little-endian Format) Angles are given in the ISO 8855 / DIN 70000 scanner coordinate system

<u>Bytes</u>	<u>Offset</u>	<u>Scan point:</u>	<u>Data Type</u>	<u>Description</u>
1	0	Layer	UINT4	Scan layer of this point (zero-based). Use the low nibble / bits 0...3 of this byte
		Echo	UINT4	Echo number of this point (zero-based). Use the high nibble / bits 4...7 of this byte.
1	1	Flags	Bit field 8 bits	0x01: transparent point 0x02: clutter (atmospheric) 0x04: ground 0x08: dirt 0xF0: reserved
2	2	Horizontal angle	INT16	Angle of this point in angle ticks in the scanner coordinate system
2	4	Radial distance	UINT16	Distance of this point in the scanner coordinate system in cm
2	6	Echo pulse width	UINT16	Detected width of this echo pulse in cm
2	8	Reserved	UINT16	
	10			

## 3 Die Datenaufbereitung

Die Daten des Versuchsfahrzeuges sind in ihrer rohen Form nicht für eine Sensorfusion geeignet. Die verschiedenen Sensordaten müssen zunächst in eine überführbare Form gebracht werden. Diese Form ist von dem Ziel der Sensorfusion abhängig.

Hier ist das Fusionsziel eine Aufwertung der Kameradaten durch die Laserscanner. Die Kamerabilder sollen um die Objektdaten der Scanner erweitert werden. Ein mögliches Ergebnis ist die Reduzierung des Rechenaufwands für spätere Bildbearbeitungsoperationen.

Somit sind Bilddaten das Fusionsziel. Es müssen hierzu die verzeichneten Kamerabilder entzerrt und die Rohdaten der Laserscanner in ein übertragbares Format gebracht werden.

### 3.1 Unterstützende Programme und Softwarebibliotheken

Für die Bearbeitung von Bilddaten ist eine Vielzahl von Verarbeitungsalgorithmen erforderlich. In der begleitenden Software dieser Bachelorarbeit wird hierfür die freie Programmbibliothek Open Source Computer Vision Library (OpenCV) verwendet.

Die OpenCV Bibliothek bietet nach Angabe der Entwickler über 2500 optimierte Algorithmen, welche bei der Bildbearbeitung hilfreich sind. OpenCV bietet Interfaces für mehrere Programmiersprachen wie C/C++ Java oder Python.

Für die Kamera-Laserscanner-Sensorfusion wird OpenCv hauptsächlich für das Laden und Speichern der Bilddaten, den Zugriff auf einzelne Bildelemente und die Bildmanipulation verwendet.

Für die Berechnung einiger Parameter wird das Programm MATLAB eingesetzt. Die Parameter werden in der Software als Konstanten verwendet

MATLAB bietet laut Angabe der Entwickler eine Vielzahl von Werkzeugen zur Visualisierung von Daten und der Erstellung interaktiver Umgebungen zu Analysezwecken. Zur Lösung verschiedener Problemstellungen stehen unterschiedliche Mathematische Funktionen zur Verfügung.

MATLAB erleichtert hier die Lösung einiger Berechnungen. Es wird speziell für die Bestimmung der Korrekturparameter für die Berichtigung der Linsenverzeichnung und zur Berechnung der Transformationsparameter bei der eigentlichen Sensorfusion eingesetzt.

Das Versuchsfahrzeug arbeitet zur Datenspeicherung mit einer PostgreSQL Datenbank. PostgreSQL wird als freies, objektrelationales Datenbankmanagementsystem beschrieben. Das Datenbanksystem ist ACID konform und hält sich zum Großteil an den SQL-Standard ANSI-SQL 2008.

Für diese Bachelorarbeit wird ein PostgreSQL Datenbanksystem verwendet um eine Kopie der Fahrzeugaufzeichnungen zu verwalten. Das Programmpaket pgAdmin ermöglicht den Zugriff auf das Datenbanksystem.

## 3.2 Kameradaten

Nahezu jede Kamera weist, bedingt durch den Einsatz einer Kameralinse, eine leichte Bildverzeichnung auf. Durch die zusätzliche Verwendung eines Fish-Eye Objektivs im Testfahrzeug besitzen die Kamerabilder, als Konsequenz, eine starke Verzeichnung. Diese ist als kissen- oder tonnenförmige Verzerrung im Bild zu sehen (Abb.1).

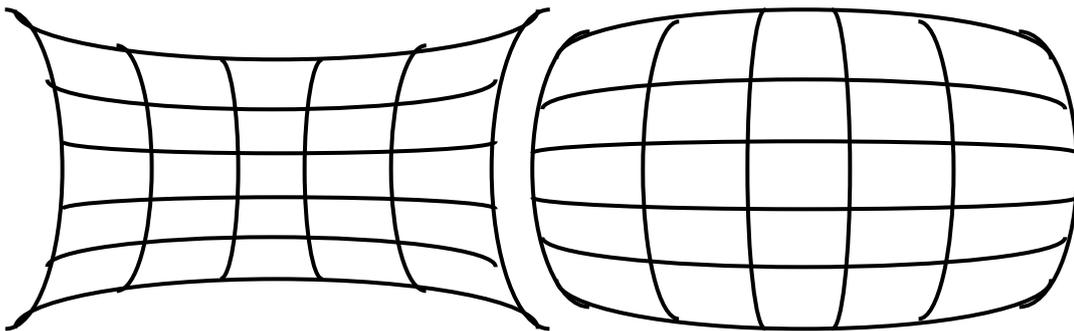


Abb. 1: Kissenförmige Verzerrung

Tonnenförmige Verzerrung

Mit steigender Entfernung vom Bildmittelpunkt werden gerade Linien immer stärker als Kurven dargestellt. Die vier Kanten eines Bildes sind somit am stärksten von dieser Verzerrung betroffen. Die Verzeichnung muss vor der eigentlichen Sensorfusion, durch entsprechende Techniken der Bildbearbeitung, korrigiert werden.

Hierzu sind Kenntnisse über die technischen Spezifikationen der eingesetzten Kamera sowie eventuell zusätzlich verwendeter Objektivs notwendig. Die Kenndaten des Sensorelementes, mit dem die Kamera ausgestattet ist, sind hierbei von entscheidender Bedeutung.

Die Korrektur der Bilddaten ist wichtig für die weiteren Bearbeitungsschritte der Sensorfusion. Eine Bildverzeichnung führt zu einer fehlerhaften Darstellung des Kamerabildes. Abstände, Objektkanten und Objektformen werden verzerrt. Die Korrektur der Verzeichnung erleichtert die Integration der Laserscanner-Daten.

Auch spätere mögliche Applikationen wie die Objekterkennung und –auswertung profitieren von einer Korrektur der Verzeichnung. Viele Bildverarbeitungstechniken sind mit diesen Vorgängen verbunden. Diese Techniken setzen ein verzeichnungskorrigiertes Bild voraus

### **3.2.1 Transformationsansatz**

Bei der Bildbearbeitung wird generell zwischen zwei Bearbeitungsansätzen unterschieden. Diese Ansätze sind die Source to Target Transformation und die Target to Source Transformation. Die Transformationsansätze bieten unterschiedliche Vorteile.

#### Source to Target Transformation

Bei der Source to Target Transformation werden die einzelnen Elemente des Quellbildes (Source) über entsprechende Transformationsgleichungen in das Zielbild überführt. Es wird hierbei durch das Quellbild iteriert. Die Transformationsgleichung bestimmt die Position des Quellbildelementes im Zielbild. Hierbei kann es vorkommen, dass einzelne Elemente außerhalb vom Abbildungsbereich des Zielbildes platziert werden. Es ist auch möglich, dass nicht jede mögliche Position des Zielbildes mit einem Quellbildelement besetzt wird. Es entstehen sichtbare Fehlstellen. Hierdurch leidet die Bildqualität. Abhängig von der angestrebten Bildtransformation ist der Rechenaufwand und damit auch die Rechenzeit geringer als bei der Target to Source Transformation.

#### Target to Source Transformation

Die Target to Source Transformation verfolgt den umgekehrten Ansatz zur Source to Target Transformation. Es wird für jede Position des Zielbildes ein Quellbildelement ermittelt. Hierfür wird durch die einzelnen Positionen des Zielbildes iteriert und mittels einer Transformationsgleichung ein entsprechendes Quellbildelement ermittelt. Abhängig von der angestrebten Bildtransformation ist es möglich, dass durch die Transformationsgleichung kein eindeutiges Quellbildelement bestimmt werden kann. Für die Besetzung einer Position des Zielbildes können so zwei oder vier mögliche Quellbildelemente ermittelt werden. Dieser Umstand erfordert oft die Durchführung zusätzlicher Rechenoperationen zur endgültigen Bestimmung des Quellbildelements. Die Target to Source Transformation hat den Vorteil, dass ein transformiertes Bild keine Fehlstellen aufweist. Die Qualität des transformierten Bildes ist damit wesentlich besser. Abhängig von der Bildtransformation steigen Rechenaufwand und damit auch Rechenzeit.

---

In einem Fahrzeug, welches nur über begrenzte Rechenkapazität verfügt, sind Rechenaufwand und Rechenzeit ein wichtiger Faktor. In dieser Bachelorarbeit wird somit für die Linsenverzeichnungskorrektur der Source to Target Transformationsansatz verwendet.

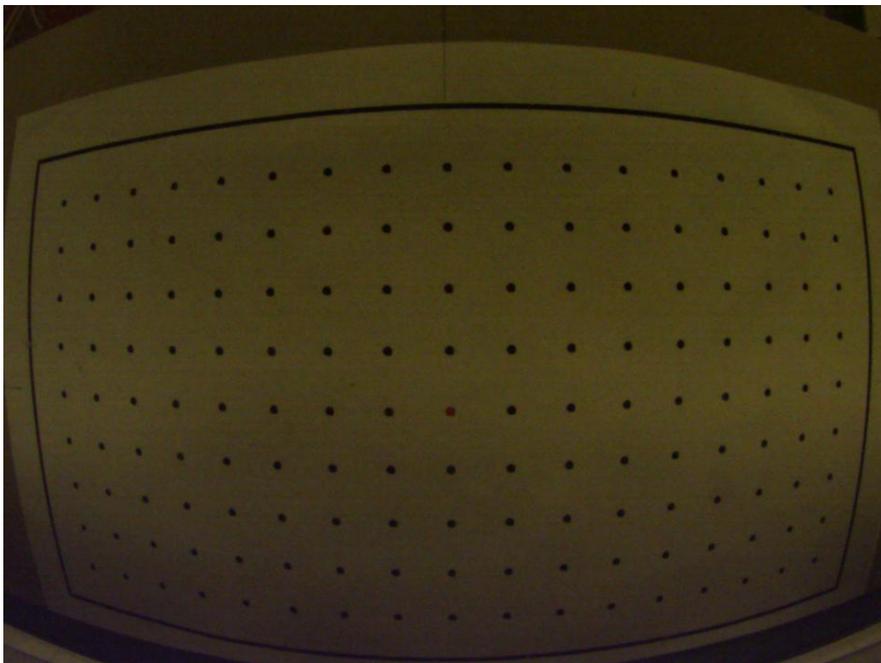
### 3.2.2 Korrektur der Linsenverzeichnung

Zur Berichtigung der Linsenverzeichnung ist es notwendig die kameraspezifischen Verzeichnungsparameter zu bestimmen. Die Verzeichnungsparameter lassen sich mittels unterschiedlicher Verfahren bestimmen.

Es ist möglich die Parameter über die Kameralinse, das eingesetzte Objektiv oder dem Abgleich mit einer Kalibrierungsplatte zu bestimmen. Die Parameter sind abhängig von der verwendeten Transformationsformel.

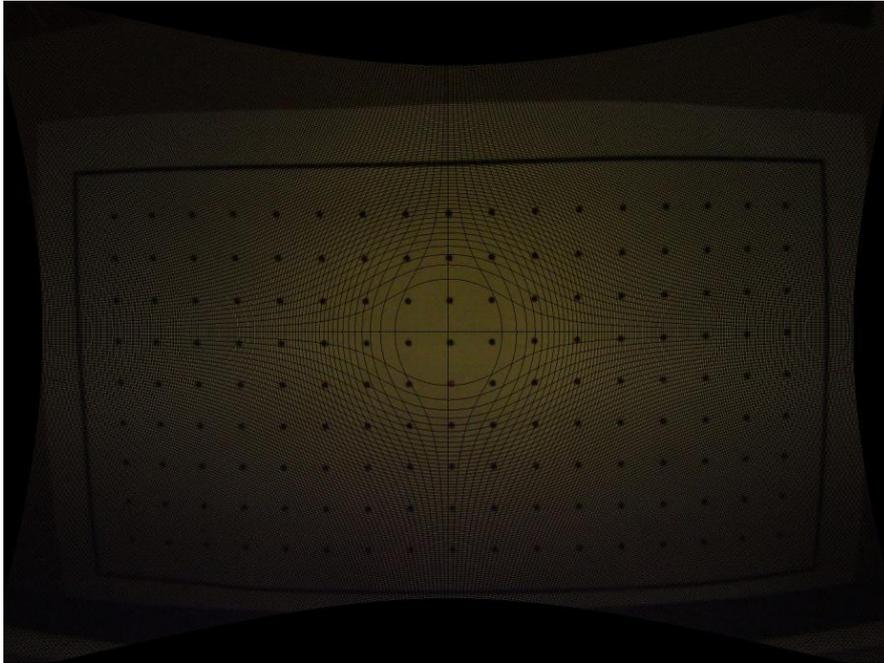
In dieser Bachelorarbeit wird zur Parameterbestimmung eine Kalibrierplatte verwendet. Hierbei wird eine Platte mit einem erkennbaren Raster parallel zum Kamerasensor positioniert. Die Platte sollte hierbei das Kamerabild ausfüllen.

Das aufgenommene Bild weist jetzt die Linsenverzeichnung auf (Abb.2). Die Elemente des Bildes können jetzt auf die einzelnen Fotozellen des Kamerasensors umgerechnet werden. Die Maße des Fotosensors und die Größe der Fotozellen finden hier ihre Anwendung. Sie werden den entsprechenden Datenblättern entnommen oder aus diesen abgeleitet. Die Verzeichnungsparameter lassen sich ermitteln, durch einen Vergleich der aktuell verzeichneten Position der Rasterelemente mit den realen Positionen des Rasters. Die realen Positionen werden der Kalibrierplatte entnommen und auf die Sensorfläche umgerechnet. Es werden nur wenige Vergleichspunkte benötigt, um die Parameter der Transformationsgleichung zu bestimmen.



**Abb. 2:** Linsenverzeichnetes Bild

Da als Transformationsansatz die Source to Target Transformation gewählt wurde, weist das korrigierte Bild Fehlstellen auf (Abb.3). Das entstandene Muster ist typisch für die Korrektur eines verzeichneten Bildes. Es ist möglich die Fehlstellen durch die Verwendung entsprechender Bildbearbeitungstechniken aufzufüllen. Dies würde zusätzliche Rechenzeit benötigen und die Bildqualität beeinflussen. In dieser Arbeit wird auf diese Möglichkeit verzichtet.



**Abb. 3:** Korrigiertes Bild mit Fehlstellen

Solange der aktuelle Aufbau des Kamerasystems nicht verändert wird, reicht eine einmalige Bestimmung der Verzeichnungsparameter. Diese Parameter werden als Konstanten in der Begleitsoftware verwendet. Nach der Korrektur der Linsenverzeichnung können jetzt weitere Bildtransformationen notwendig werden.

Abhängig vom Aufbau des Kamerasystems und dessen Einsatz kann es erforderlich sein Aufnahmewinkel auszugleichen oder Filter anzulegen. Für die Sensorfusion am Versuchsfahrzeug ist dies nicht erforderlich.

### 3.2.3 Mathematische Grundlagen zur Linsenverzeichnungskorrektur

#### Verwendeter Verzeichnungsansatz:

Mathematischer Ansatz zur Korrektur der Linsenverzeichnung für eine Source to Target Transformation.

$$u_c = u_d * (1 + K_1 * r^2 + K_2 * r^4)$$

$$v_c = v_d * (1 + K_1 * r^2 + K_2 * r^4)$$

$$r = \sqrt{u_d^2 + v_d^2}$$

$u_c, v_c$ : korrigierte Bildkoordinaten

$u_d, v_d$ : verzeichnete Bildkoordinaten

$K_1, K_2$ : Verzeichnungsparameter

#### Kalibrierung:

Einmalige Bestimmung der Verzeichnungsparameter für die Verzeichnungskorrektur.

Eingabewerte:

$u_d^{\#}, v_d^{\#}$ : verzeichnete unzentrierte Bildkoordinaten [Pixel]

$u_{pp}, v_{pp}$ : korrigierte Bildkoordinaten auf der Passpunktplatte [ mm]

Umrechnung der Bildkoordinaten der Passpunkte in zentrierte Sensorkoordinaten:

$$u_d = (u_d^{\#} - u_H) / s_x$$

$$v_d = (v_d^{\#} - v_H) / s_y$$

$u_d, v_d$ : zentrierte Sensorkoordinaten [Pixel]

$u_H, v_H$ : Bildhauptpunktverschiebung [Pixel]

$s_x, s_y$ : Sensorelemente pro mm

Berechnung der Wunschkoordinaten:

$$u_{pp,c} = u_{pp} - u_{pp,0}$$

$$v_{pp,c} = v_{pp} - v_{pp,0}$$

$u_{pp,c}, v_{pp,c}$ : zentrierte Passpunktkoordinaten

$u_{pp,0}, v_{pp,0}$ : Durchstoßpunkt der opt. Achse durch Passpunktplatte

$$u_w = u_{pp,c} * m$$

$$v_w = v_{pp,c} * m$$

$$m = \frac{\text{Bildgröße}}{\text{Gegenstandsgröße}}$$

$u_w, v_w$ : Wunschkoordinaten der Passpunkte

$m$ : optischer Abbildungsfaktor

Bestimmung der Verzeichnungsparameter:

$$\vec{L} = \begin{bmatrix} u_{w1} - u_{d1} \\ \vdots \\ u_{wn} - u_{dn} \\ v_{w1} - v_{d1} \\ \vdots \\ v_{wn} - v_{dn} \end{bmatrix}$$

$$r = \sqrt{u_d^2 - v_d^2}$$

$$\underline{A} = \begin{bmatrix} u_{d1} * r_1^2 & u_{d1} * r_1^4 \\ \vdots & \vdots \\ u_{dn} * r_n^2 & u_{dn} * r_n^4 \\ v_{d1} * r_1^2 & v_{d1} * r_1^4 \\ \vdots & \vdots \\ v_{dn} * r_n^2 & v_{dn} * r_n^4 \end{bmatrix}$$

$$\vec{L} = \begin{pmatrix} K_1 \\ K_2 \end{pmatrix} * \underline{A}$$

$r$ : Radius [Pixel]

Verzeichnungskorrektur:

Positionskorrektur der einzelnen Pixel jedes Bildes

$$u_d = (u_d^\# - u_H) / s_x$$

$$v_d = (v_d^\# - v_H) / s_y$$

$$r = \sqrt{u_d^2 + v_d^2}$$

$$u_c = u_d * (1 + K_1 * r^2 + K_2 * r^4)$$

$$v_c = v_d * (1 + K_1 * r^2 + K_2 * r^4)$$

$$u_c^\# = u_c * s_x + u_H$$

$$v_c^\# = v_c * s_y + v_H$$

$u_c^\#, v_c^\#$ : korrigierte unzentrierte Bildkoordinaten [Pixel]

## 3.3 Laserscandaten

Das Fahrzeug erfasst seine Umgebung mit zwei Laserscannern. Die Scanner sind sowohl rechts als auch links in der Front des Versuchsfahrzeugs, unterhalb der Lichtanlagen, montiert. Die Sensoren werden durch ein gemeinsames Taktsignal synchronisiert. Die verwendeten Laserscanner bieten mehrere Optionen für den Aufbau der Datenformate an. Im Versuchsfahrzeug wird der Datentyp 0x2202 des ibeo LUX Laserscanners verwendet. Die gesammelten Daten werden in der Datenbank des Fahrzeugs abgelegt. Jeder Laserscanner liefert pro Scanvorgang einen eigenen Datensatz. Zur einfachen Speicherung in der Datenbank werden die Scandaten zuvor in Stringketten konvertiert. Numerische sowie hexadezimale Werte bleiben als entsprechende Stringelemente erhalten.

Für die Kamera-Laserscanner-Sensorfusion werden die Scandaten aus der Datenbank verwendet.

### 3.3.1 Datenformat der Scaninformationen

Das Ziel dieser Sensorfusion ist eine Integration der Laserscanner-Daten in die Kamerabilder. Es ist notwendig die Scandaten in eine übertragbare Form zu überführen. Hier werden, für die Sensorfusion, die Scandaten in Bilddaten umgewandelt.

Der Datensatz ist in drei Elemente aufteilbar. Der erste Abschnitt ist im big-endian-Format codiert. Er bildet den Header des Datensatzes. Der Header enthält unter anderem Informationen über die Nachrichtengröße, die Identität des Ursprungsgerätes und den Zeitpunkt der Nachrichtenerstellung. Wichtig für die weitere Auswertung sind die Informationen zur Bestimmung des Formattyps des folgenden Datenabschnittes.

Im Testfahrzeug wird für die Codierung des zweiten Abschnittes der Datentyp 0x2202 des Scannermodels verwendet. Der Abschnitt ist damit im little-endian-Format codiert. Eine Vielzahl der Informationen dieses Abschnittes ist, für die Aufwertung der Scandaten, notwendig. Die Montageposition und die Montagewinkel des Scanners sind in diesem Abschnitt zu finden. Die Koordinaten von Position und Winkel werden auf das Fahrzeugkoordinatensystem bezogen. Durch die ISO 8855 / DIN 70000 wird das Koordinatensystem definiert.

Der letzte Abschnitt enthält die Aufzählung der Scanpunkte. Die einzelnen Punkte enthalten ihre jeweiligen Informationen über Winkel und Distanz zum erfassten Objekt, die Echopulsweite sowie die nähere Informationen zum Echopuls. Dieser Abschnitt ist, wie der zweite Abschnitt, im little-endian-Format codiert.

### 3.3.2 Analyse der Scaninformationen

Die weitere Bearbeitung der Scandaten erfordert die Zerlegung der String Kette und die Verarbeitung der Informationen. In der erstellten Software wird hierfür ein Strukt zur Informationsverwaltung verwendet. Das Zerlegen und Verarbeiten der Informationen in diesen String Ketten, wird durch verschiedene Umstände erschwert.

Ein Großteil des Datensatzes ist im little-endian-Format codiert. Für die weitere Verwendung ist es erforderlich die Codierung anzupassen.

Die Scandaten werden, in dieser Bachelorarbeit, der Datenbank des Versuchsfahrzeuges entnommen. Im Fahrzeug selbst werden die Informationspakete für die Sicherung zuvor vom Bussystem gesammelt und in String Elemente konvertiert. Im Anschluss werden die gesammelten Daten in die Fahrzeugdatenbank geschrieben. Während dieses Prozesses entstehen gelegentlich Speicherunregelmäßigkeiten. Es kommt vor, dass eine String Kette um wenige Elemente zu lang ist. Dieser Umstand erfordert eine zusätzliche Kontrolle der Datensätze.

Zuletzt variieren die Größe des Datensatzes und damit die Länge der String Kette stark. Die Länge der Kette ist abhängig von der Anzahl der Scanpunkte, welche vom entsprechenden Laserscanner ermittelt wurden.

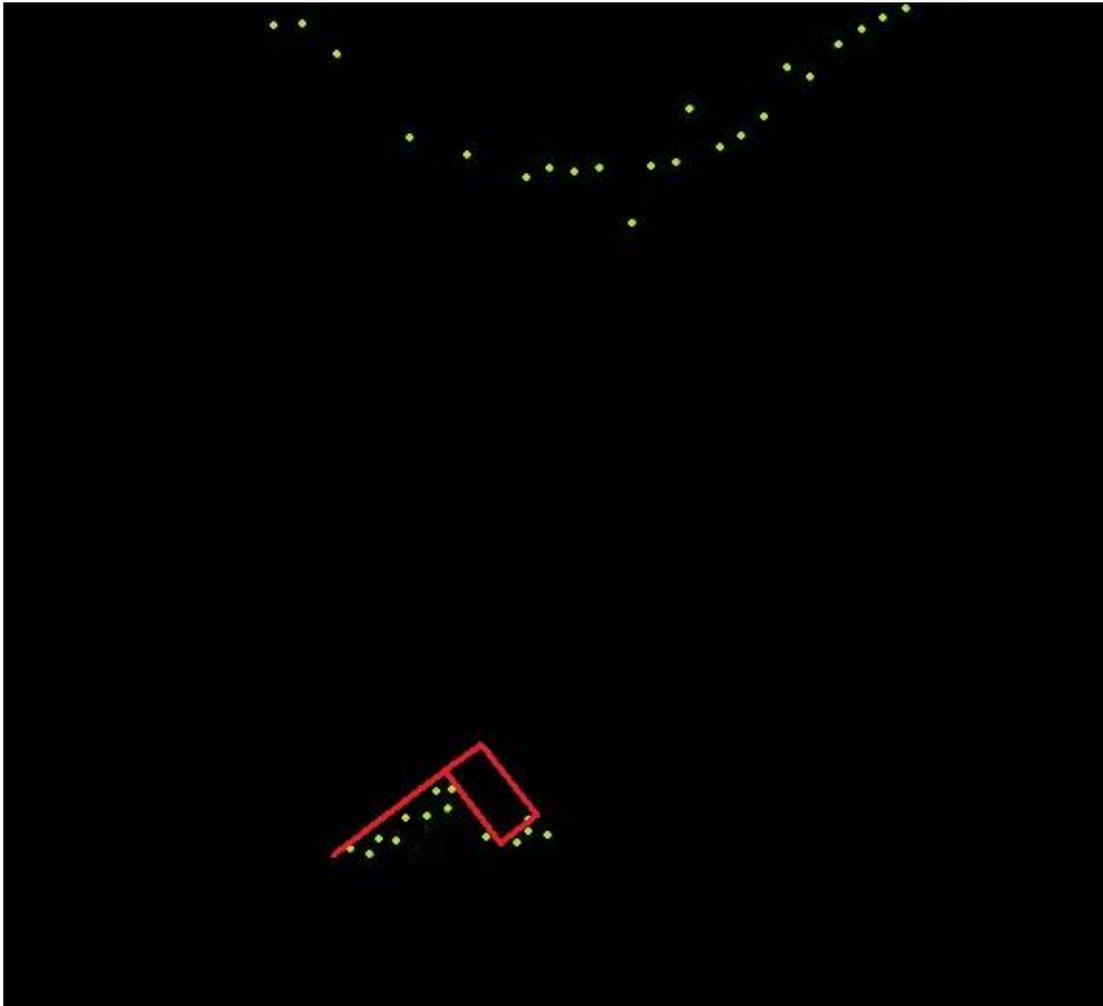
Nach dem Zerlegen der Zeichenkette und der Korrektur des Formates, können die einzelnen Zeichenketten in mathematische Größen konvertiert werden. Hierfür bieten die C++ String Bibliothek viele Methoden an. In der Begleitsoftware werden hexadezimal dargestellte Zeichenfolgen, mit der Hilfe verschiedener Stringstream Funktionen, in verwertbare Größen übertragen. Es ist nun möglich die Scandaten auszuwerten.

### 3.3.3 Grafische Darstellung der Scaninformationen

Die Auswertung der Scandaten beginnt bei den einzelnen Scanpunkten. Die Informationen der Datenfelder „Layer“, „Horizontalangle“ und „Radial distance“, aus dem Abschnitt der Scanpunkte, sowie das Feld „angle ticks per rotation“, aus dem Abschnitt Datentyp, werden hierbei benötigt.

Das Datenfeld Layer ordnet den Punkt einer Scanebene zu. Der eingesetzte Laserscanner verfügt über vier Scanebenen. Für Jede Scanebene entsteht so ein eigenes Umgebungsabbild. Die Informationen der Felder „Horizontalangle“ und „Radial distance“ beschreiben die Position eines Objektes, welches von Laserstrahl getroffen wurde. Das Feld „Radial distance“ wird hier in Zentimeter und das Feld „Horizontal angle“ in „angle ticks“ angegeben. Die „angle ticks“ erlauben zusammen mit der Information aus dem Feld „angle ticks per rotation“ die Bestimmung des horizontalen Winkels.

Betrachtet man eine Scanebene als zweidimensionale Fläche, erlaubt uns die Verwendung verschiedener Winkelfunktionen das Setzen eines Punktes auf dieser Ebene. Bei der Notation des Punktes ist die Lage des Koordinatenursprungs zu beachten. Als Koordinatenursprung wird hier die linke, untere Bildecke gewählt. Dies erleichtert die Interpretation der Grafik. Durch die Wiederholung dieses Vorgangs, für jeden erfassten Scanpunkt, erhält man vier Umgebungsabbilder für den jeweils linken und rechten Laserscanner. Jedes Umgebungsabbild ist auf das Koordinatensystem des Scanners bezogen (Abb.4). Die Abbilder müssen, zur weiteren Bearbeitung, in das Fahrzeugkoordinatensystem überführt werden.



**Abb. 4:** Ausschnitt Umgebungsabbild im Scanner-Koordinatensystem

Ein fahrzeuggebundenes Koordinatensystem ist ein rechtshändiges dreidimensionales Koordinatensystem (Abb.5). Die Position der Fahrzeugkomponenten wird hier auf einen Fahrzeugmittelpunkt bezogen. Bei Kraftfahrzeugen wird dieser Punkt meist auf die Mitte der Abstände beider Fahrzeugachsen gelegt. Mit Ausnahme der positiven x-Achse, welche immer in die primäre Bewegungsrichtung zeigt, ist die Ausrichtung der restlichen Achsen vom Fahrzeugtyp abhängig. Die positive y-Achse ist bei Landfahrzeugen nach links gerichtet. Die positive z-Achse zeigt nach oben und steht senkrecht zur x-y-Ebene.

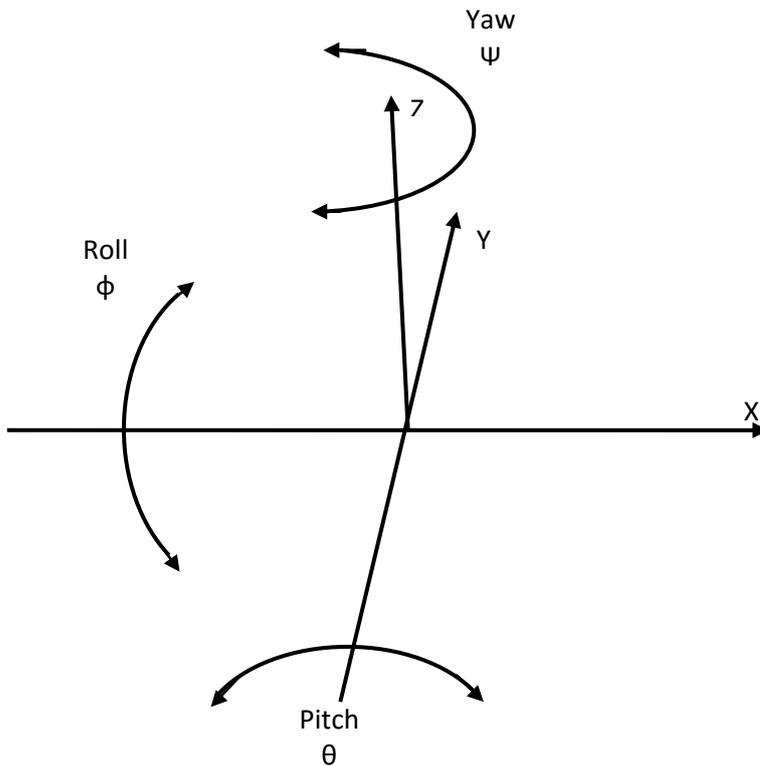


Abb. 5: Fahrzeugkoordinatensystem

Um die Umgebungsbilder der Scanner auf das Fahrzeugkoordinatensystem zu übertragen werden mehrere Informationen aus dem zweiten Abschnitt des Datensatzes benötigt. Die drei Felder „Mounting position yaw, pitch, roll angle“ beinhalten Daten über die Montagewinkel. Durch eine entsprechende Rotation der Punkte um ihren Ursprungspunkt, wird die Übertragung ins Fahrzeugkoordinatensystem durchgeführt. Die Felder „Mounting position x, y, z“ beschreiben die Position des jeweiligen Scanners im Fahrzeugkoordinatensystem. Diese Informationen erlauben es die Punkte vom Koordinatensystem des Scanners in das System des Fahrzeuges zu übertragen. Dies kann mittels einer Translation der Koordinaten bewerkstelligt werden.

Die Abfolge von Rotation und Translation sind zu beachten. Es muss mit der Rotation der Punkte in der Reihenfolge „yaw-angle“, „pitch-angle“ und „roll-angle“ begonnen werden. Die Translation erfolgt zum Schluss.

Im Versuchsfahrzeug sind nur die beiden Datenfelder „Mounting position y“ und „Mounting position yaw angle“ mit Informationen versehen. Der Nullpunkt der x-Achse des Fahrzeugkoordinatensystem wurde damit auf die Frontstoßstange des Fahrzeugs gelegt, in der die Scanner montiert sind. Dies vereinfacht die Kombination der Umgebungsabbilder beider Laserscanner zu einem Gesamtabbild, für jede der vier Scanebenen (Abb.6).



**Abb. 6:** Ausschnitt Umgebungsabbild im Fahrzeugkoordinatensystem

Die Umgebungsabbilder sind eine graphische Interpretation der Scandaten. Die Bilder gestatten einen Einblick in die Sensorwahrnehmung. Beim Betrachten der Bilder fällt auf, dass nur wenige Scanpunkte, im Verhältnis zur nötigen Bildgröße, existieren. Für die Fusion der Umgebungsabbilder mit den Kameraaufnahmen wäre eine Suche nach den Bildkoordinaten der Scanpunkten erforderlich. Dies würde zusätzliche Rechenzeit beanspruchen.

Es werden daher, im Sensorfusionsmodul, nur die Positionen der Scanpunkte im Umgebungsabbild berechnet. Die Positionen werden im Anschluss in einer einfachen Liste vermerkt. Die Liste dient damit als virtuelles Bild. Es ist somit nicht notwendig, für die eigentliche Sensorfusion, grafische Abbilder der Umgebung zu erstellen. Die benötigte Zeit zum Auffinden einzelner Scanpunkte im Bild, wird so auf die Zugriffszeit der einzelnen Listenelemente reduziert.

Die Umgebungsabbilder werden aber dennoch einmalig benötigt. Sie dienen der Bestimmung vom verschiedenen Werten und damit der Berechnung mehrerer Transformationsparameter.

### 3.3.4 Mathematische Grundlagen zur Laserdatenanalyse

#### Scanwinkelbestimmung:

Ermittlung des Winkels zwischen Scanner und Reflexionspunkt des Laserstrahls im Koordinatensystem des Scanners.

$$\varphi = 2\pi \frac{a_t}{a_{tpr}}$$

$\varphi$ : Scanwinkel [rad]

$a_t$ : Anzahl der Scannerrotationsschritte (angle ticks)

$a_{tpr}$ : Gesamtzahl der Scannerrotationsschritte (angle ticks per rotation)

#### Umgebungsabbildgenerierung für Scanner:

Erstellung des Umgebungsabbildes im Koordinatensystem des Scanners. Der Scanner wird als Koordinatenuhrsprung angenommen.

$$x = r * \cos \varphi$$

$$y = r * \sin \varphi$$

$x, y$ : Kartesische Koordinaten

$r$ : Entfernung zum Scanpunkt (radial distance) [cm]

Umgebungsabbildgenerierung für Fahrzeug:

Übertragung des Umgebungsabbildes vom Scanner-Koordinatensystem in das Fahrzeugkoordinatensystem.

Rotation:

$$u_r = y * \sin \varphi + x * \cos \varphi$$

$$v_r = y * \cos \varphi - x * \sin \varphi$$

$u_r, v_r$ : Koordinaten eines Punktes im neuen System

Translation;

$$u_t = u_r + a$$

$$v_t = v_r + b$$

$u_t, v_t$ : Endkoordinaten nach Translation

$a, b$ : Verschiebung der Achsen (mounting position x) (mounting position z)

## 4 Die Sensorfusion

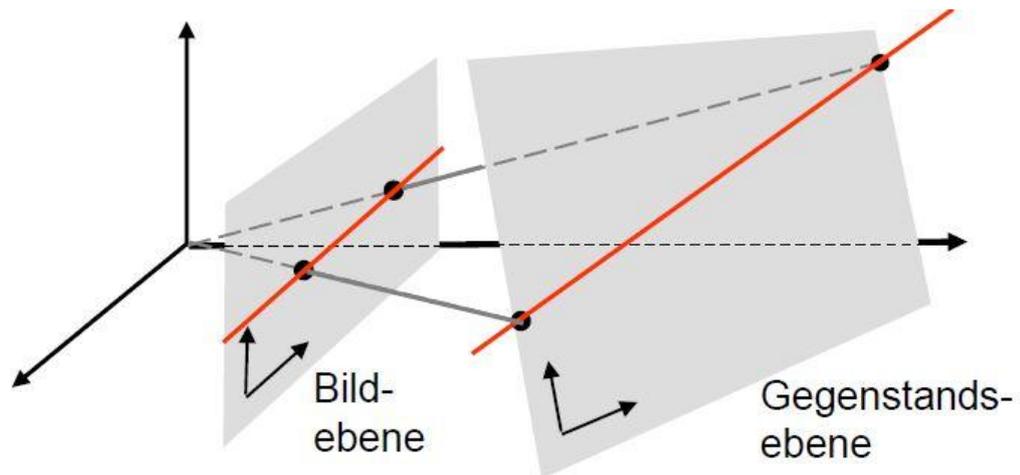
Die Kamera-Laserscanner-Sensorfusion hat das Ziel die Sensordaten der Laserscanner mit den Kamerabildern zu verbinden. Hierbei sollen die Kamerabilder um die Informationen der Laserscanner erweitert werden. Die Sensordaten der Laserscanner wurden hierfür in Bilddaten überführt. Die Kamerabilder wurden von der Linsenverzeichnung befreit. Um die Daten zu verbinden sind Kenntnisse über die Lage der Scanebenen im Raum erforderlich. Die Perspektivische Transformation, eine Technik der Bildbearbeitung zur nichtlinearen Formwiederherstellung, erlaubt die Übertragung der Scanpunkte in das Kamerabild.

### 4.1 Perspektivische Transformation

Die Perspektivische Transformation ist eine Technik aus dem Bereich der geometrischen Bildtransformation. Dem Begriff der geometrischen Bildtransformation werden verschiedene Techniken, wie Verschiebung oder Scherung zur Bearbeitung eines Bildes, zugeordnet. Die Techniken Verschiebung, Drehung, Scherung oder Skalierung gehören zur Gruppe der affinen Transformation. Die Basis der affinen Transformation sind lineare Polynome.

Die polynomale Transformation bildet eine weitere Gruppe der geometrischen Bildtransformation. Diese Gruppe beinhaltet weitergehende Techniken zur Verformung eines Bildes. Die polynomale Transformation ist damit eine Erweiterung der affinen Transformation. Die polynomale Transformation ist, anders als die affine Transformation, nicht immer Parallelen erhaltend.

Die Perspektivische Transformation ist eine polynomale Transformationstechnik zur nichtlinearen Formwiederherstellung. Die Idee der nichtlinearen Formwiederherstellung verfolgt die Verwendung zweidimensionaler Transformationen, zur Annäherung an die Lösung dreidimensionaler Probleme. Die Perspektivische Transformation verwendet einen Ebene zu Ebene Transformationsansatz. Mit der Perspektivische Transformation wird eine Senkrechtsicht einer ebenen Fläche aus einer Schrägansicht bestimmt (Abb.7). Dabei werden zwei zweidimensionale Flächen miteinander verrechnet. Die Umgebungsabbilder des Laserscanners bilden aufgrund ihrer Natur, als Schnitt durch den Raum, eine zweidimensionale Fläche. Die Kameraaufnahmen stellen eine Sicht in einen dreidimensionalen Raum da. Nach der Korrektur der Linsenverzeichnung, kann die Kameraaufnahme auch als zweidimensionale Fläche betrachtet werden. Es wird hier der Target to Sorce Transformationsansatz verwendet.



**Abb. 7:** Perspektivische Transformation

Anders als die Affine Transformation ist die Perspektivische Transformation nicht parallelerhaltend. Sie ist dennoch geradenerhaltend.

## 4.2 Anwendung der Perspektivischen Transformation auf die Sensorfusion

Die perspektivische Transformation errechnet aus einer Schrägansicht einer ebenen Fläche eine Senkrechtansicht. Als Eingangsparameter werden ein Quellbild und acht Transformationsparameter benötigt. Die perspektivische Transformation ist eine Ebene zu Ebene Transformation.

Die Umgebungsabbilder der Laserscanner und die bearbeiteten Kameraaufnahmen bilden die Quelldaten der Sensorfusion. Eine Zuordnung zwischen einem beliebigen Scanpunkt und einer entsprechenden Position in der Kameraaufnahme, ist das gewünschte Ergebnis. Die Kameraaufnahmen stellen hier die primären Quelldaten, welche erweitert werden sollen. Die Umgebungsabbilder der Laserscanner sind die sekundären Quelldaten, mit denen die Erweiterung durchgeführt wird.

Die Laserscanner erzeugen, durch den Sensoraufbau des Versuchsfahrzeugs, vier Schnitte durch den Raum im Wahrnehmungsbereich der Kamera. Diese Schnitte bilden die vier Scanebenen. Sie enthalten die Objektinformationen, zur Integration ins Kamerabild, in der Form von Scanpunkten. Ein abschließendes Fusionsbild enthält somit die Objektdaten der Laserscanner als sichtbare Markierungen.

Betrachtet man die Umgebungsabbilder der Laserscanner aus der Sicht des Fusionsbildes, kann man diese leicht als das Ergebnis einer perspektivischen Transformation interpretieren. Das Fusionsbild, als Wunschergebnis der Sensorfusion, kann als das Quellbild der perspektivischen Transformation angesehen werden. Die einzelne Scanebenen im Fusionsbild würde die jeweilige Schrägansicht darstellen.

Von dieser Überlegung ausgehend sind die einzelnen Scanpunkte Positionen, die dem Quellbild entnommen wurden. Es wird der Target to Source Transformationsansatz verwendet. Bei diesem wird für jede Position im Zielbild ein Quellbildpunkt ermittelt, dem die entsprechenden Bildinformationen entnommen werden. Die Scanpunkte der Umgebungsabbilder bilden hier die Gruppe der Bildpositionen, für die eine Quellbildkoordinate innerhalb der Schrägansicht bestimmt wird. Die Position der Schrägansicht im Fusionsbild spiegelt die Lage der Scanebenen im Raum.

Um diesen Ansatz zur Sensorfusion umzusetzen, sind Information über die unterschiedliche Wahrnehmung verschiedener Objekte durch Kamera und Laserscanner erforderlich. Weiterhin ist es notwendig die Lage der Scanebenen im Raum zu ermitteln.

### 4.3 Unterschiede in der Objektwahrnehmung zwischen Kamera und Laserscanner

Laserscanner und Kamera erfassen dieselben Objekte unterschiedlich. Die Laserscanner-Daten erzeugen, nach ihrer Überführung in grafische Elemente, eine skizzenhafte Umgebungsdarstellung. Obwohl diese Darstellung wie ein Blick aus der Vogelperspektive auf die Umgebung wirkt, ist sie in Wahrheit recht unvollständig. Die einzelnen Punkte lassen die Formen verschiedener Objekte wie Fahrzeuge erkennen. Allerdings werfen die Elemente, abhängig von ihrer Position zum Scanner, einen Schatten auf mögliche dahinter liegende Objekte.

Diese Objekte, welche zum Teil oder auch komplett im Schatten anderen Elemente stehen, werden unvollständig oder gar nicht vom Scanner detektiert. Mit zunehmender Entfernung verlieren die Laser an Fokus. Damit wird die Positionsbestimmung ungenau. Es fällt auch beim Betrachten der Umgebungsabbilder auf, dass die Abstände der Scanpunkte zueinander mit steigender Entfernung zum Scanner zunehmen. Der Laser fächert sich horizontal auf. Schmale Objekte können auf großer Entfernung der Erfassung entgehen. Zusätzlich verlaufen die einzelnen Ebenen des Laserscanners nicht Parallel zueinander. Die Scanebenen fächern sich vertikal leicht auf.

Die Darstellung der Objekte auf den Kamerabildern ist abhängig von der Perspektive und der Entfernung zur Kamera. Weiter entfernte Objekte werden im Bildhintergrund kleiner dargestellt als nahe Objekte im Vordergrund. Abhängig von der Aufnahmeperspektive erschwert dies die Bestimmung der Maße einzelner Bildelemente, wenn keine bekannten Vergleichselemente vorhanden sind.

Objekte die abseits des Fokuspunktes positioniert sind, unterliegen einer Verzerrung die zur Unschärfe der Darstellung führt. Dem gleichen Effekt unterliegen Objekte, die sich vor oder hinter dem Fokuspunkt befinden. Dies ist abhängig von der verwendeten Linse und dem eingesetzten Objektiv sowie verschiedener technische Hilfen. Als Beispiel dient hier die bekannte Autofokus-Funktion verschiedener Kameramodelle.

Um die Fusion der Sensordaten durchzuführen, sind weitere Daten und Informationen über die Relationen der Objekte zueinander erforderlich.

## 4.4 Objektrelationen zwischen Kamera und Laserscanner

Um die Umgebungsabbilder des Laserscanners und die Aufnahmen der Kamera in Relation zu setzen, ist es notwendig einzelne Punkte in beiden Darstellungsformen zu identifizieren und zuordnen zu können. Betrachtet man die Umgebungsabbilder des Laserscanners, ist es möglich verschiedene Objekte zu identifizieren. Gegenstände wie Fahrzeuge oder Wände sind durch ihre charakteristischen Konturen einfach zu bestimmen (Abb.8). Vergleicht man diese Konturen mit den Kamerabildern, ist es möglich diese Objekte auch im Bild zu finden (Abb.9). Dies ist aber nicht ausreichend um einzelne Punkte korrekt zu bestimmen. In allen vier Scanebenen werden die Konturen einer Wand, eines Fahrzeuges oder anderen großer Gegenstände erfasst. Eine Zuordnung auf eine bestimmte Position im Raum oder im Kamerabild ist aber, aufgrund fehlender Informationen über die Lage der Scanebenen im Raum, nicht möglich. Es ist notwendig zuerst die Lage der Scanebenen zu ermitteln. Dies ist möglich, indem man die Position einzelner ermittelter Scanpunkte mit Höhenwerten ergänzt.

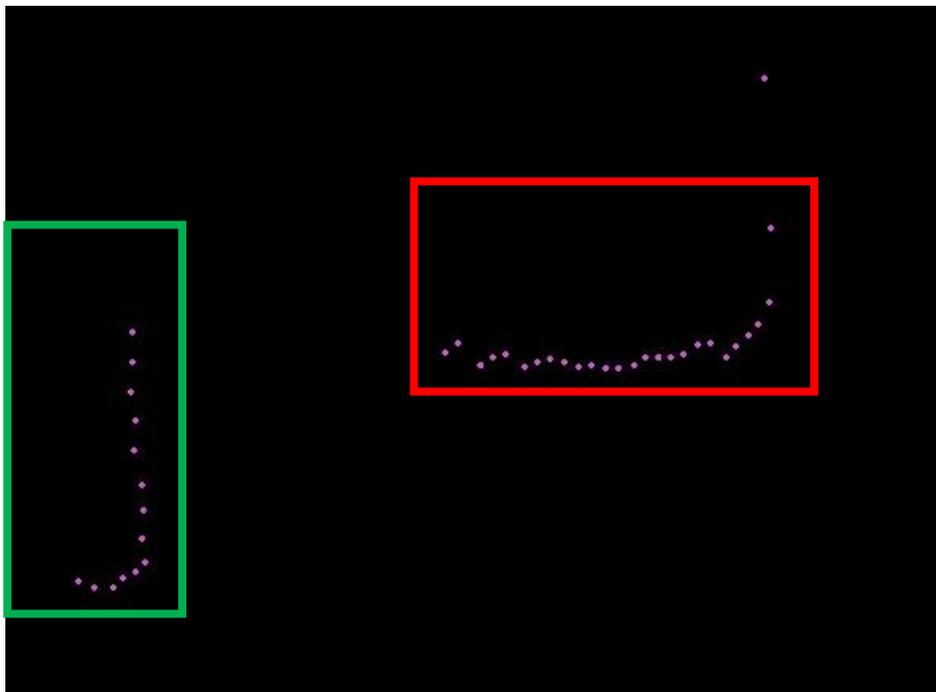


Abb. 8: Objektvergleich Laserscanner

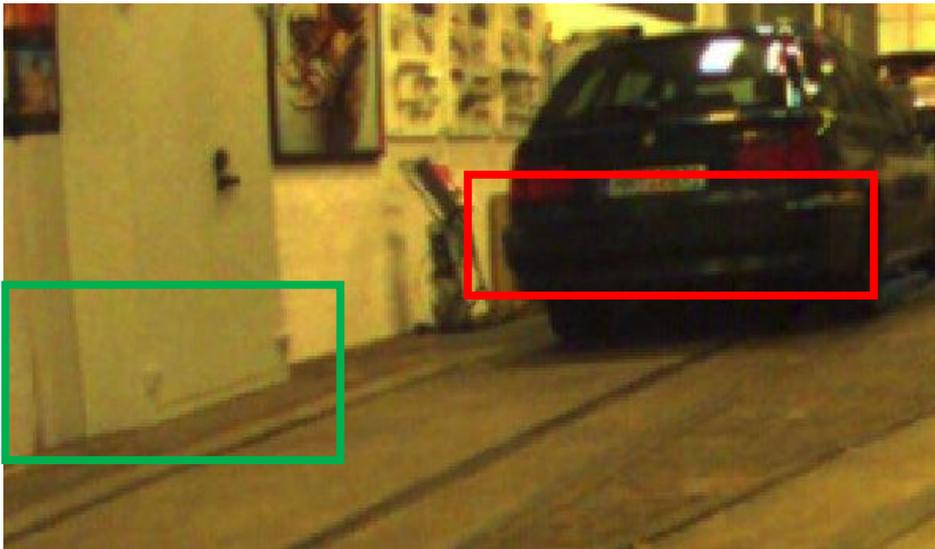


Abb. 9: Objektvergleich Kamera

Durch die Umwandlung der Scanner-Daten in grafische Umgebungsabbilder ist ein verwendbares Koordinatensystem entstanden, welches Objekte auf einer zweidimensionalen Ebene darstellen kann. Das Koordinatensystem besitzt seinen Ursprung in der unteren linken Ecke des Umgebungsabbildes. Die Position der Laserscanner befindet sich auf der X-Achse äquivalent zu ihrer Position im Fahrzeugkoordinatensystem. Die Koordinaten der Scanpunkte werden, um die halbierte Breite der Abbildung, nach rechts verschoben. Dies positioniert alle Scanpunkte im ersten Quadranten des Koordinatensystems.

Durch die Positionierung der Scanner auf der X-Achse lassen sich die Y-Werte als Entfernung zum Scanpunkt interpretieren. Ein Bildpixel entspricht hier einem Zentimeter. Dieser Mindestmaßstab wird durch die Genauigkeit der Laserscanner vorgegeben.

Es ist hier zu beachten, dass in Computergrafiken der Ursprungspunkt des Koordinatensystems in der oberen linken Ecke der Abbildung liegt. Auch OpenCV arbeitet nach diesem Adressierungsprinzip.

In der begleitenden Software bestimmen Höhe und Breite eines Bildes oft die Dimensionen eines Koordinatensystems und damit viele Variablen in den mathematischen Berechnungen. Für die Adressierung im Bild muss die Y-Koordinate entsprechend angepasst werden. Die X-Koordinate kann meist erhalten bleiben.

## 4.5 Konzepte verschiedener Messvorrichtungen

Um einen Bezug zwischen einem Scanpunkt und einer Höhe im Raum herzustellen ist ein Gegenstand oder eine Vorrichtung erforderlich, das durch ein deutliches Profil in den Umgebungsabbildern Rückschlüsse auf die Höhe des Scanpunktes erlaubt. Weiterhin ist es erforderlich, Scanpunkte einer Position im Kamerabild zuordnen zu können. Die eingesetzte Vorrichtung oder der verwendete Gegenstand sollte auf den Kamerabildern gut sichtbar sein und Charakteristika besitzen, welche eine Verbindung zwischen Umgebungsabbild und Kameraaufnahme schaffen. Es existieren hierfür verschiedenen Möglichkeiten. Diese sind letztlich vom Systemaufbau und der eingesetzten Sensortechnik abhängig.

### 4.5.1 Aufbau 1

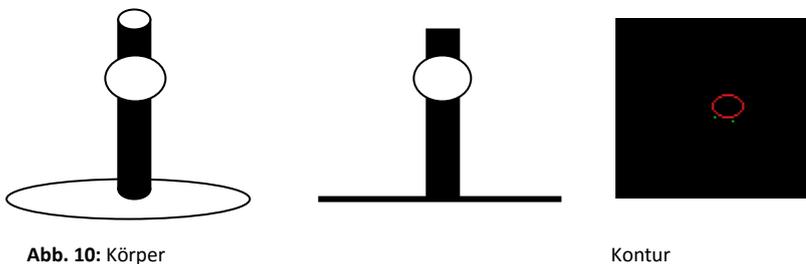


Abb. 10: Körper

Kontur

Bei diesem Aufbau wird ein Gegenstand, auf dem vertikal stehenden Stab, nach oben oder unten bewegt. Der Gegenstand erzeugt einen sichtbaren Scanpunkt, sobald er die Laserebene erreicht. Die Höhe kann abgemessen werden.

Vorteil:

- Einfaches Bestimmen der Höhe
- Einfache Zuordnung zu Bildpositionen
- Leicht durchzuführen

Nachteil:

- Erfordert Livebilder der Laserscanner
- Großer Winkel zwischen einzelnen Abtaststrahlen erschwert Positionierung (Aufbau kann zwischen zwei Abtaststrahlen stehen)
- Erfordert manuelle Einstellung und ist damit Zeitaufwendig

### 4.5.2 Aufbau 2

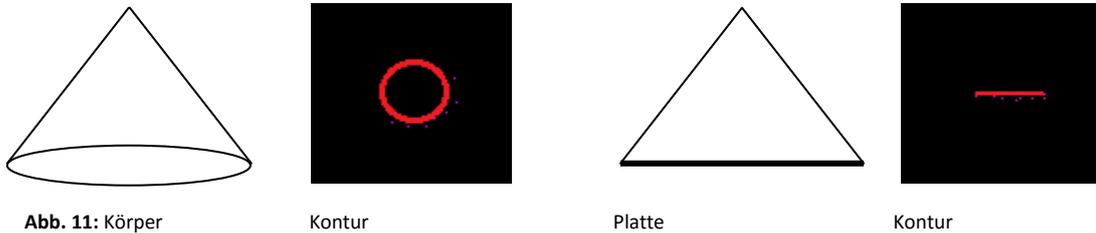


Abb. 11: Körper

Kontur

Platte

Kontur

Dieser Aufbau verwendet eine dreieckige Platte oder einen kegelförmigen Körper, um die Lage der Laserebenen zu bestimmen. Die Vorrichtung wird vor den Laserscannern und der Kamera platziert. Die Konturen der Vorrichtung im Umgebungsabbild erlauben es die Objektbreite bei der Platte oder den Durchmesser beim Körper, zu bestimmen. Über Breite oder Durchmesser lässt sich die Höhe vom Boden zur Laserebene bestimmen.

Vorteil:

- Einfaches positionieren der Vorrichtung
- Benötigt keine Livebilder der Laserscanner
- Schnelle Durchführung
- Auswertung und Höhenbestimmung kann automatisiert werden

Nachteil:

- Erfordert geringe Messtoleranz des Laserscanner für hinreichend genaue Höhenbestimmung
- Auswertung kann nur über die Laserscannerdaten erfolgen
- Schwierigkeit der Zuordnung zu Bildposition ist abhängig von Aufnahmeperspektive und Bildqualität

### 4.5.3 Aufbau 3

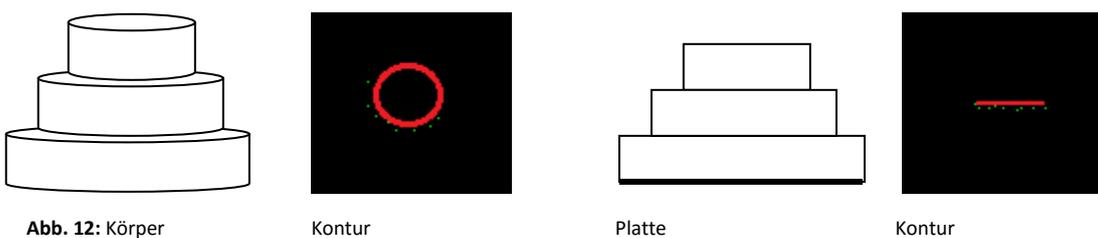


Abb. 12: Körper

Kontur

Platte

Kontur

Aufbau 3 ist mit Aufbau 2 verwandt. Anstelle glatter Oberflächen werden die Kanten der Vorrichtung mit Stufen versehen. Platte oder Körper werden wie bei Aufbau 2 vor dem Laserscannern und der Kamera platziert. Die Breite der Platte oder der Durchmesser des Körpers dient der Höhenbestimmung.

Vorteil:

- Einfaches positionieren der Vorrichtung
- Benötigt keine Livebilder der Laserscanner
- Schnelle Durchführung
- Auswertung und Höhenbestimmung kann automatisiert werden
- Stufenweite kann an Messtoleranz des Laserscanner angepasst werden

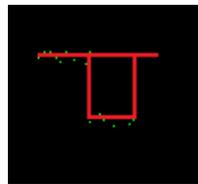
Nachteil:

- Auswertung kann nur über die Laserscanner-Daten erfolgen
- Zuordnung zu Bildposition ist abhängig von Aufnahme­perspektive und Bildqualität schwierig
- Genauigkeit bei Höhenbestimmung ist abhängig von der Stufenhöhe

#### 4.5.4 Aufbau 4



Abb. 13: Körper



Kontur

Bei diesem Aufbau wird das Stufenprinzip zur Höhenbestimmung aus Aufbau 3 aufgegriffen. Eine Treppe wird an eine breite Rückwand montiert. Die Vorrichtung wird vor dem Sensoraufbau positioniert. Die ausladende Treppe der Messvorrichtung zeigt in Richtung des Versuchsfahrzeugs. Die Höhe kann durch den Abstand zwischen Rückwand und Stufe ermittelt werden.

Vorteil:

- Einfaches positionierten der Vorrichtung
- Benötigt keine Livebilder der Laserscanner
- Schnelle Durchführung
- Auswertung und Höhenbestimmung kann automatisiert werden
- Stufenweite kann an Messtoleranz des Laserscanners angepasst werden
- Sichtbare Kanten ermöglichen eine bessere Zuordnung zu einzelnen Bildpositionen

Nachteil:

- Auswertung kann nur über die Laserscanner-Daten erfolgen
- Genauigkeit bei Höhenbestimmung ist abhängig von der Stufenhöhe

### 4.5.5 Auswertung

Die Abtaststrahlen des Laserscanners sind, für das menschliche Auge und die verwendete Kamera, nicht sichtbar. Der Einsatz einer Messvorrichtung ist notwendig zur Bestimmung der Lage der Laserebenen. Da die Laserscanner-Daten der Fahrzeugdatenbank entnommen werden, sind keine Livebilder verfügbar. Die Laserscanner im Versuchsfahrzeug erfassen Entfernungen in Zentimeter Schritten. Durch diese Umstände sind die Konzepte des ersten und des zweiten Aufbaus ungeeignet als Messvorrichtung. Es stehen damit der dritte und der vierte Aufbau zur Verfügung. Die verwendete Kamera besitzt eine maximale Auflösung von 1280 x 1024 Pixeln. Die geringe Auflösung erschwert die eindeutige Zuordnung zwischen Scanpunkten und Bildpositionen. Aufgrund der besseren Sichtbarkeit wird für dieses Versuchsfahrzeug der vierte Aufbau verwendet. Abhängig von der Aufnahmeperspektive ist es mit diesem Aufbau möglich, mehrere korrespondierende Punkte zwischen Umgebungsabbild und Kameraaufnahme zu bestimmen.

## 4.6 Umsetzung des Konzeptes

Für die Sensoranordnung im Versuchsfahrzeug wird das vierte Konzept der verschiedenen Messvorrichtungen umgesetzt, um die genaue Lage der Scanebenen zu bestimmen.

Es stehen mehrere Materialien für die Konstruktion zur Auswahl. Die Konstruktion wird aus Holz angefertigt. Holz ist anders als Metall oder Kunststoff leicht zu bearbeiten. Kunststoff und Metall erfordern unterschiedliche Werkzeuge für die Bearbeitung oder die Formung des Materials. Metall ist ein schwerer Werkstoff, wodurch die Konstruktion auch viel Gewicht erhält. Kunststoff ist, abhängig vom verwendeten Material, umständlich zu bearbeiten oder benötigt Spezialwerkzeuge.

Für die Messvorrichtungen wird eine Höhe von 40 cm gewählt. Damit liegt die obere Kante der Konstruktion über den Montagepositionen der Laserscanner.

Die Breite der Rückwand wird, wie die Höhe, auf 40 cm festgelegt. Die kleinste horizontale Schrittweite des Laserscanners beträgt  $1/32^\circ$ . Mit einer Konstruktionsbreite von 40 cm entstehen somit genug Punkte, um die Konturen der Messvorrichtung in den Umgebungsabbildern erkennen zu können.

Die im Versuchsfahrzeug verwendeten Laserscanner erfassen Entfernungen in Zentimeter Schritten. Der Abstand einer Stufe zur folgenden Stufe wird auf 3 cm festgelegt. Dies erlaubt es einzelne Stufen zuzuordnen und damit die Abstände zwischen Stufe und Rückwand zu bestimmen.

Für die einzelne Stufe wird eine Höhe von 2 cm und eine Breite von 14 cm festgelegt. Diese Höhe erlaubt eine leichtere Zuordnung der Scanpunkte zur Position im Kamerabild.

Mit steigender Entfernung zwischen Kamera und Messvorrichtung umfasst eine Höhenstufe nur noch wenige Pixel im Bild.

Bedingt durch das Konstruktionsmaterial und die Montage der Komponenten erreichen die einzelnen Stufen eine Höhe von ca. 2,13 bis 2,14 cm. Die Konstruktion erreicht so eine Gesamthöhe von 40,5 cm und umfasst 19 Stufen (Abb.14).



**Abb. 14:** Erstellte Messvorrichtung

Die Konstruktion wird vor den Laserscannern des Versuchsfahrzeuges platziert. Ein Kamerabild wird erstellt (Abb.15).



**Abb. 15:** Messvorrichtung vor Versuchsfahrzeug

Durch die Konstruktionsform ergeben sich zwei Konturprofile in den Umgebungsabbildern. Diese entstehen durch die unterschiedlichen Abtastwinkel, beim platzieren der Konstruktion auf der rechten (negative Y-Achse) oder der linken (positive Y-Achse) Seite des Fahrzeugs (Abb.16).



Abb. 16: Messvorrichtung Darstellung Scanner rechts und Scanner links

Anhand der Konturen lässt sich zunächst die Position der Rückwand bestimmen. Im Anschluss ermittelt man die Position der Stufen. Über den Abstand zwischen Stufe und Rückwand wird der Höhenlevel bestimmt (Abb.17). Dies wird für alle Scanebenen durchgeführt.



Abb. 17: Messvorrichtung Darstellung Scanebene 0 und Scanebene 3

Abhängig von der Aufnahmeperspektive bieten sich verschiedene Punkte für die Zuordnung an (Abb.18). Für die Zuordnung der Scanpunkte zu den jeweiligen Bildpunkten können die Kanten der Konstruktion verwendet werden. Die Kanten und Eckpunkte sind auf den Kameraaufnahmen gut zu erkennen. Generell ist die vollständig sichtbare Kante zu bevorzugen.

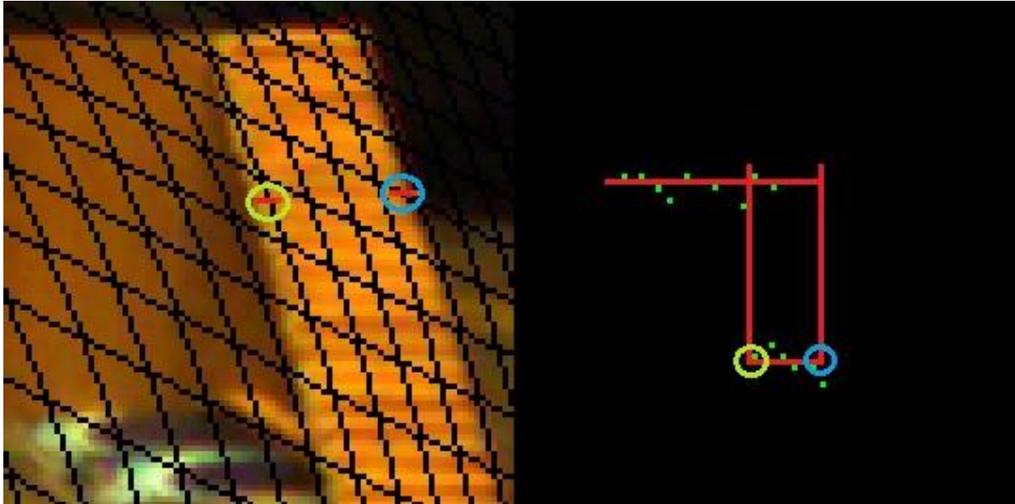


Abb. 18: Mögliche Zuordnungspunkte

## 4.7 Lagebestimmung der Scanebenen

Um die Sensorfusion mittels der perspektivischen Transformation durchzuführen, ist die Bestimmung der Transformationsparameter  $b_{11}$  bis  $b_{32}$  erforderlich. Hierfür werden äquivalente Punkte in beiden Bildern benötigt.

Mit Hilfe der Messvorrichtung werden äquivalente Punkte im Umgebungsabbild der Laserscanner und der Kameraaufnahme bestimmt. Damit sind ein Teil der Variablen für die Perspektivische Transformation bekannt. Es ist möglich, mit diesen Daten die Transformationsparameter der Scanebene zu berechnen.

Die Durchführung dieser Berechnung ist nur einmalig erforderlich. Die berechneten Parameter werden als Konstanten in der Begleitsoftware verwendet. Wird der Sensoraufbau des Versuchsfahrzeuges verändert, ist es notwendig die Parameter erneut zu bestimmen.

Um die Lage der Scanebenen im Raum, und damit auch die acht Parameter für die Transformationsgleichung, zu ermitteln werden mindestens vier unterschiedliche Scanpunkte für jede Ebene benötigt. Der verwendete Messaufbau erzeugt pro Messung in

jeder Scanebene verwendbare Punkte. Die Koordinaten der Scanpunkte werden ausgewählt und notiert. Im Kamerabild werden die Positionen der jeweiligen Scanpunkte ermittelt und deren Koordinaten erfasst.

Um die Transformationsparameter zu berechnen wird die Transformationsgleichung aufgelöst. Die unbekannt Parameter werden von den bekannten Größen getrennt.

$$x_q = \frac{b_{11} * x_z + b_{12} * y_z + b_{13}}{b_{31} * x_z + b_{32} * y_z + 1}$$

$$y_q = \frac{b_{21} * x_z + b_{22} * y_z + b_{23}}{b_{31} * x_z + b_{32} * y_z + 1}$$

$$\rightarrow x_q * (b_{31} * x_z + b_{32} * y_z + 1) = b_{11} * x_z + b_{12} * y_z + b_{13}$$

$$\rightarrow y_q * (b_{31} * x_z + b_{32} * y_z + 1) = b_{21} * x_z + b_{22} * y_z + b_{23}$$

$$\rightarrow x_q = b_{11} * x_z + b_{13} * y_q + b_{13} - b_{31} * x_z * y_q - b_{32} * y_q * x_z$$

$$\rightarrow y_q = b_{21} * x_z + b_{22} * y_q + b_{12} - b_{31} * x_z * y_q - b_{32} * y_q * x_z$$

Es ergeben sich zwei Gleichungen, für die Berechnung der x-Koordinate und der y-Koordinate.

$$x_{qPk} = b_{11} * x_{zPk} + b_{13} * y_{qPk} + b_{13} - b_{31} * x_{zPk} * y_{qPk} - b_{32} * y_{qPk} * x_{zPk}$$

$$y_{qPk} = b_{21} * x_{zPk} + b_{22} * y_{qPk} + b_{12} - b_{31} * x_{zPk} * y_{qPk} - b_{32} * y_{qPk} * x_{zPk}$$

$$k = 1 \dots 4$$

Diese Gleichungen erlauben, mit den Koordinatensätzen der vier korrespondierenden Punkte, die Aufstellung einer Rechenmatrix.

$$A_z * B = L_q$$

$A_z$ : Eingangsmatrix mit Parametern

$B$ : Transformationsparameter

$L_q$ : Ergebnismatrix mit Quellbildpunkten

$$\begin{bmatrix}
 x_{zP1} & y_{zP1} & 1 & 0 & 0 & 0 & -x_{zP1}x_{qP1} & -y_{zP1}x_{qP1} \\
 0 & 0 & 0 & x_{zP1} & y_{zP1} & 1 & -y_{zP1}x_{qP1} & -y_{zP1}y_{qP1} \\
 x_{zP2} & y_{zP2} & 1 & 0 & 0 & 0 & -x_{zP2}x_{qP2} & -y_{zP2}x_{qP2} \\
 0 & 0 & 0 & x_{zP2} & y_{zP2} & 1 & -y_{zP2}x_{qP2} & -y_{zP2}y_{qP2} \\
 x_{zP3} & y_{zP3} & 1 & 0 & 0 & 0 & -x_{zP3}x_{qP3} & -y_{zP3}x_{qP3} \\
 0 & 0 & 0 & x_{zP3} & y_{zP3} & 1 & -y_{zP3}x_{qP3} & -y_{zP3}y_{qP3} \\
 x_{zP4} & y_{zP4} & 1 & 0 & 0 & 0 & -x_{zP4}x_{qP4} & -y_{zP4}x_{qP4} \\
 0 & 0 & 0 & x_{zP4} & y_{zP4} & 1 & -y_{zP4}x_{qP4} & -y_{zP4}y_{qP4}
 \end{bmatrix}
 *
 \begin{bmatrix}
 b_{11} \\
 b_{12} \\
 b_{13} \\
 b_{21} \\
 b_{22} \\
 b_{23} \\
 b_{31} \\
 b_{32}
 \end{bmatrix}
 =
 \begin{bmatrix}
 x_{qP1} \\
 y_{qP1} \\
 x_{qP2} \\
 y_{qP2} \\
 x_{qP3} \\
 y_{qP3} \\
 x_{qP4} \\
 y_{qP4}
 \end{bmatrix}$$

$x_{zPk}, y_{zPk}$ : Zielbildpunkte zur Parameterbestimmung

$b_{mn}$ : Transformationsparameter

$x_{qPk}, y_{qPk}$ : Quellbildkoordinaten zur Parameterbestimmung

Das entstandene Lineare Gleichungssystem lässt sich mit unterschiedlichen Methoden auflösen. Ein Beispiel wäre die Anwendung des Gaußschen Eliminationsverfahrens.

Hier wurde für die Berechnung erneut das Programmpaket MATLAB eingesetzt. Die Eingangsmatrix der Zielbildpunkte und die Ergebnismatrix der Quellbildkoordinaten werden in das Programm eingegeben. Die MATLAB-Funktion `linsolve` ermöglicht die Lösung dieses Gleichungssystems. Die Ergebnismatrix der MATLAB-Funktion beinhaltet die jeweiligen Transformationsparameter einer Scanebene.

## 4.8 Durchführung der Sensorfusion

Die Linsenverzerrung der Kameraaufnahme ist korrigiert. Die Daten der zwei Laserscanner sind kombiniert und besitzen ein Format, das mit der Kameraaufnahme kompatibel ist. Die Lage der Scanebenen ist ermittelt und die jeweiligen Transformationsparameter bestimmt.

Die Fusion der Sensordaten wird mittels der Verwendung der perspektivischen Transformation durchgeführt. Es wird der Target to Source Ansatz verwendet. Die Transformationsparameter werden in die Gleichung eingesetzt. Die Koordinaten der Scanpunkte, in den Umgebungsabbildern des Laserscanners, werden bestimmt. Der Koordinatenursprung ist hierbei zu beachten. Die Koordinatensätze werden in die Gleichung, an die Stellen der Zielbildkoordinaten, eingesetzt. Mit der vollständigen Gleichung werden die korrespondierenden Koordinaten im Kamerabild berechnet.

Durch die Berechnung der Koordinaten im Kamerabild besteht ein Bezug zwischen den Umgebungsabbildern des Laserscanners und den Kameraaufnahmen. Eine Zuordnung der einzelnen Scanpunkte zu einzelnen Pixeln im Kamerabild ist jetzt möglich. Die Kameraaufnahme wurde um die Informationen des Laserscanners erweitert. Technisch ist die Kamera-Laserscanner-Sensorfusion für Kraftfahrzeuge damit abgeschlossen.

Zur Ansicht werden die einzelnen Scanpunkte im Kamerabild markiert. Jede Scanebene erhält hier, zur erleichterten Identifizierung, eine eigene farbliche Kenzeichnung (Abb.19).

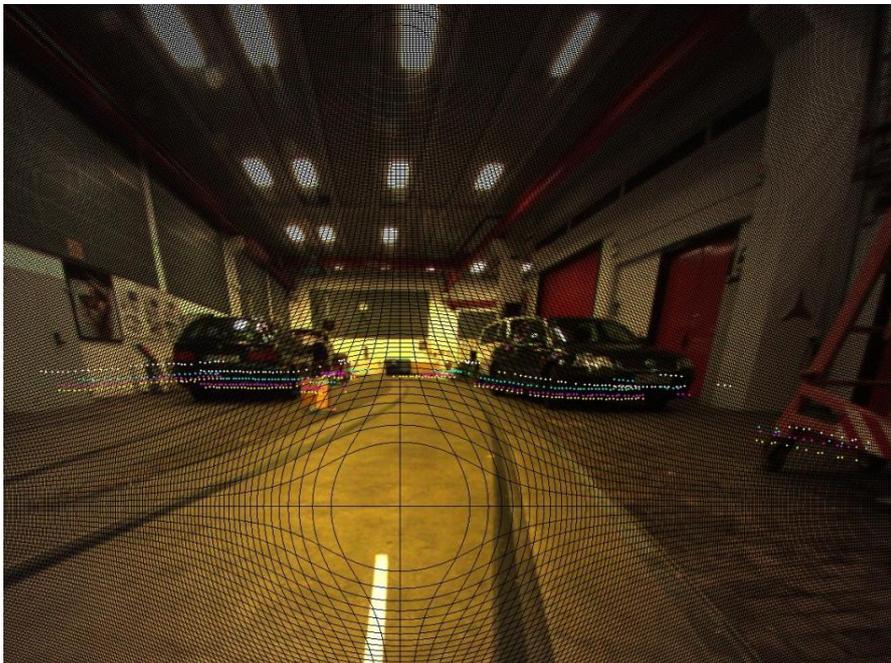


Abb. 19: Fusionsbild Ausschnitt

Beim Betrachten des Fusionsbilds fällt eine Besonderheit auf. Einige Markierungen, die den Verlauf der Hallenwände abbilden, wirken im Fusionsbild deplatziert (Abb.20).

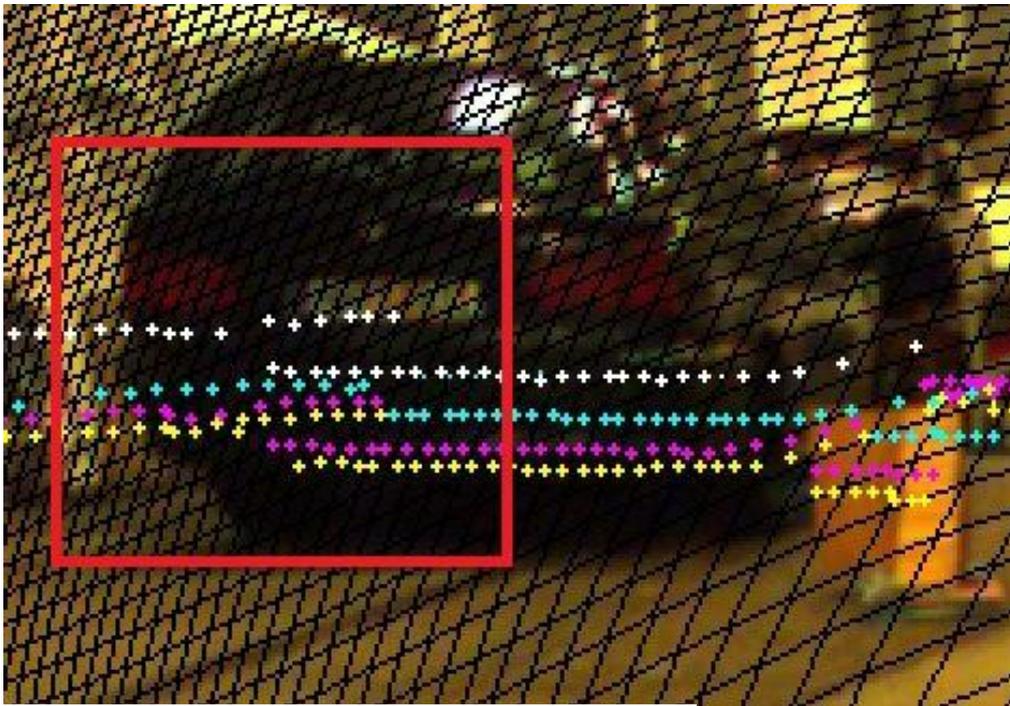
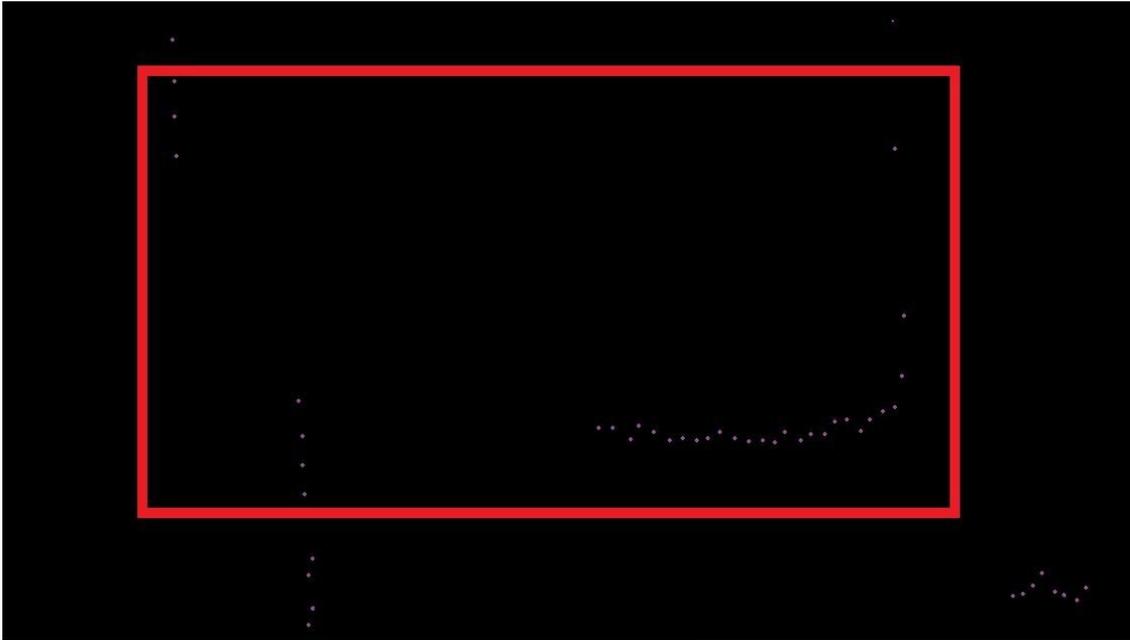


Abb. 20: Deplatzierte Markierungen

Anhängig von der Montageposition können die Laserscanner im Umgebungsabbild Konturen erfassen, die im Kamerabild durch andere Gegenstände verdeckt sind. Hier verdeckt das parkende Fahrzeug die dahinterliegende Wand.

Da die Wand im Umgebungsabbild erfasst wurde, wird auch diese durch die Perspektivische Transformation in das zweidimensionale Kamerabild übertragen (Abb.21).



**Abb. 21:** Blickfeld Laserscanner

Um diesen Effekt zu vermeiden kann man das Umgebungsabbild in der Breite einschränken, um die unerwünschten Informationen zu ignorieren. Besser ist eine Anpassung des Sensoraufbaus bei der ein Laserscanner unter der Kamera montiert wird. So besitzen Kamera und Laserscanner ein vergleichbares Blickfeld.

## 4.9 Mathematische Grundlagen zur Perspektivischen Transformation

Perspektivische Transformation Target to Source:

Transformationsgleichung für das Target to Source Mapping.

$$x_q = \frac{b_{11} * x_z + b_{12} * y_z + b_{13}}{b_{31} * x_z + b_{32} * y_z + 1}$$

$$y_q = \frac{b_{21} * x_z + b_{22} * y_z + b_{23}}{b_{31} * x_z + b_{32} * y_z + 1}$$

$x_q, y_q$ : Quellbildkoordinaten

$x_z, y_z$ : Zielkoordinaten

$b_{mn}$ : Transformationsparameter Target to Source

Perspektivische Transformation Source to Target:

Transformationsgleichung für das Source to Target Mapping.

$$x_z = \frac{b_{11}^* * x_q + b_{12}^* * y_q + b_{13}^*}{b_{31}^* * x_q + b_{32}^* * y_q + 1}$$

$$y_z = \frac{b_{21}^* * x_q + b_{22}^* * y_q + b_{23}^*}{b_{31}^* * x_q + b_{32}^* * y_q + 1}$$

$x_q, y_q$ : Quellbildkoordinaten

$x_z, y_z$ : Zielkoordinaten

$b_{mn}^*$ : Transformationsparameter Source to Target

# 5 Zusammenfassung und Ausblick

## 5.1 Zusammenfassung

Diese Arbeit wurde durch das Communication over Real-Time Ethernet (CoRE) Projekt der HAW Hamburg inspiriert. Das Versuchsfahrzeug der Projektgruppe ist umfassend mit verschiedenen Sensoren ausgestattet. Die Kamera und Laserscanner werden jedoch nur eingeschränkt genutzt. Unter dem Aspekt der Bildverarbeitung und der Robotvision wurden daraufhin mögliche Einsatzbereiche erörtert. Das Thema dieser Arbeit: „Kamera-Laserscanner-Sensorfusion für Kraftfahrzeuge“ ist ein Ergebnis dieses Vorgangs.

Das Ziel dieser Arbeit ist es, ein Konzept für eine Sensorfusion von Kameraaufnahmen und den Umgebungsdaten der Laserscanner zu erstellen. Für die Sensorfusion wurden Techniken der Bildbearbeitung als Arbeitsgrundlage gewählt.

Ein Teil dieser Arbeit beschäftigt sich mit den Sensordaten. Für die Sensorfusion war es zunächst erforderlich die Rohdaten aufzubereiten. Die Kameraaufnahmen wiesen eine starke Linsenverzeichnung auf. Für eine Weiterverarbeitung dieser Aufnahmen wurde die Verzeichnung im ersten Schritt korrigiert. Die Sensordaten der Laserscanner sind in ihrer rohen Form nicht zu interpretieren. Die Rohdaten wurden in grafische Umgebungsabbilder überführt. Hierzu wurden zunächst, separat für jeden Scanner, Umgebungsabbilder im Koordinatensystem des Scanners erstellt. Im Anschluss wurden diese Koordinatensysteme auf das Fahrzeugkoordinatensystem übertragen. Diese Vorgänge wurden für jeden Laserscanner und jede Scanebene durchgeführt. Die benötigten grafischen Umgebungsabbilder entstanden durch die Kombination der einzelnen Abbilder des linken und des rechten Laserscanners.

Die Schwierigkeit dieser Sensorfusion liegt in der unterschiedlichen Wahrnehmung der Sensoren. Die Laserscanner schneiden einen dreidimensionalen Raum. Diese Schnitte lassen zweidimensionale Umgebungsabbilder entstehen. Die Lage dieser Schnitte im Raum ist unbekannt. Die Kamera bildet einen dreidimensionalen Raum auf einer zweidimensionalen Fläche ab. Die Bildbearbeitungstechnik der perspektivischen Transformation wurde eingesetzt um die beiden zweidimensionalen Flächen miteinander zu verrechnen. Hierzu war es erforderlich die Lage der Scanebene im Raum zu ermitteln.

Zusätzlich mussten Kameraaufnahmen und Umgebungsabbilder untereinander in Bezug gesetzt werden.

Um einen Bezug zwischen Kameraaufnahme und Umgebungsabbild herzustellen, und damit auch die Lage der Scanebene im Raum zu bestimmen, wurde eine Messkonstruktion entworfen und gebaut. Die Konstruktion erlaubte es eine Beziehung zwischen Aufnahme und Umgebungsabbild herzustellen. Die benötigten Parameter für die perspektivische Transformation wurden im Anschluss ermittelt und die Sensorfusion durchgeführt.

Die erhaltenen Fusionsbilder zeigen, dass die verwendete Technik durchaus anwendbar ist. Das Konzept ist anpassbar und damit auf verschiedene Fahrzeugtypen und Sensorkonfigurationen anwendbar. Die Natur der verwendeten Technik setzt aber voraus, dass eine Sensorkonfiguration nach der Parameterbestimmung nicht mehr verändert wird. Die Parameter zur Korrektur der Linsenverzeichnung sind an die verwendete Kamerakonfiguration gebunden und führen bei einer Änderung des Kameramodels zu unbrauchbaren Ergebnissen. Gleiches gilt für die Parameter der perspektivischen Transformation. Diese Parameter sind zusätzlich abhängig von dem Model und der Einbauposition des Laserscanners. Zu beachten ist weiterhin, dass die verarbeiteten Kamerabilder und die Scanner-Daten zeitlich synchronisiert werden müssen, um fehlerhafte Darstellungen zu verhindern. Die praktische Umsetzung der Datensynchronisation ist abhängig vom internen Kommunikationssystem des Fahrzeugs und damit sehr unterschiedlich. Die, in dieser Arbeit, verwendeten Datensätze wurden dem stillstehenden Fahrzeug entnommen, um diese Problemstellung zu vermeiden.

## 5.2 Einsatzmöglichkeiten

Für den Einsatz der Kamera-Laserscanner-Sensorfusion in Kraftfahrzeugen gibt es eine Vielzahl von Möglichkeiten.

Eine Anwendungsmöglichkeit betrifft den Einsatz einer Kamera zur Parkassistentz. Eine in der Front oder im Heck installierte Kamera soll hier einen Fahrer beim Einparken des Fahrzeuges unterstützen. Das Kamerabild wird meist über ein Display im Wageninneren angezeigt. Kleine Displays oder niedrig auflösende Kameras schränken oft die Bildqualität ein. Es gehen, durch eine niedrige Bildqualität, gelegentlich Tiefeninformationen in der Aufnahme verloren. Durch die Sensorfusion ist es möglich Hindernisse, die dem Fahrzeug gefährlich nahe kommen, durch Einfärbung oder Markierung hervorzuheben. Eine verwandte Anwendungsmöglichkeit könnte darin bestehen Personen oder Objekte, deren Position ein Kollisionsrisiko darstellt, zu markieren. Das Ziel ist hier dem Fahrer erweiterte Informationen zu bieten und die Qualität des Assistenzsystems zu erhöhen.

Eine weitere Anwendungsmöglichkeit bietet sich beim Einsatz einer Kamera zur Straßenanalyse. Kameras können im Straßenverkehr zur Analyse von Streckenobjekten

eingesetzt werden. Ein Beispiel ist die Verkehrsschildanalyse. Eine Kamera überwacht hierbei den Streckenverlauf. Das System weist den Fahrer auf Tempolimits, STOP-Schilder oder Gefahrenbereiche wie Baustellen hin. Ein solches System muss hierfür jedes einzelne Bild im Videostream der Kamera nach einem Verkehrsschild durchsuchen. Es sind mehrere Analyseschritte notwendig, um ein Verkehrsschild im Bild zu finden und zu bestimmen. Das Finden eines Objektes im Bild, ist dabei eine der aufwändigsten Prozeduren. Es ist möglich entfernte oder auch schlecht sichtbare Objekte in diesem Analyseschritt falsch zu interpretieren und so auszuschließen. Durch die Fusion der Daten lassen sich Position und Maße der Region of Interest (ROI) durch die Laserscanner steuern. Dies hat mehrere Vorteile. Die Objektdetektion durch die Laserscanner ist wesentlich präziser. Wird ein Objekt am Straßenrand entdeckt, kann die Bildanalyse, durch Bestimmung dieser Koordinaten im Kamerabild, auf diesen Bereich konzentriert werden. Dies spart Zeit. Es ist auch nicht mehr nötig jedes einzelne Bild aufwändig nach Verkehrsschildern zu durchsuchen. Die Laserscanner übernehmen diese Aufgabe.

Die Fusion der Daten erlaubt es die Schwächen des einzelnen Systems auszugleichen. Im Kamerabild ist es aufwendig ein einzelnes Objekt zu finden aber möglich die Art der Objekte zu identifizieren. Die Hauptaufgabe der Laserscanner ist die Objektdetektion. Es ist aber schwer zu ermitteln was für ein Objekt der Scanner wahrnimmt. Die Sensorfusion kann hier einem System helfen präzisere Entscheidungen zu treffen oder auch schneller zu reagieren.

### **5.3 Bedeutung für die Zukunft**

In der Automobilindustrie werden ständig neue Ideen entwickelt und Konzepte getestet. Diese haben ein einfacheres, komfortableres oder sichereres Fahren zum Ziel. Die technische Ausstattung moderner Fahrzeuge wird immer umfangreicher und vielseitiger.

Der vermehrte Einsatz verschiedener Assistenzsysteme führt oft zur Installation verschiedener Sensorsysteme. Abstandswarner benötigen Sensoren zur Distanzerfassung. Spurhaltesysteme verwenden Kameras zur Straßenvermessung. Parkassistenten verwenden zum Teil Laserscanner zur Umgebungserfassung. Auch die Installation verschiedener Multimediasysteme und Servicegeräte haben Einfluss auf die Automobiltechnik. Die Fahrzeuge werden hierzu mit immer leistungsfähigeren Computern ausgestattet.

Betrachtet man diese Entwicklung ist es leicht vorstellbar, die vorgestellte Technik zur Sensorfusion praktisch umzusetzen. Die vorgestellte Technik kann bei einer Vielzahl von Assistenz- oder auch Sicherheitssystemen eingesetzt werden, um Leistungsvermögen oder Präzision zu erhöhen. Eine weitere Ausformung dieser Arbeit kann auch für verwandte Themengebiete, wie das autonome Fahren, profitabel sein.

## 6 Begleitende Software

Im Rahmen dieser Bachelorarbeit wurde begleitende Software erstellt, um das Sensorfusionskonzept zu Testzwecken praktisch umzusetzen. Verschiedene Grafiken und Darstellungen in dieser Bachelorarbeit entstammen dieser erstellten Software. Die Software wurde unter Verwendung der Sprache C/C++ erstellt. C/C++ wird im Versuchsfahrzeug als Standardsprache verwendet. Die freie Programmbibliothek Open Source Computer Vision Library (openCV) stellt verschiedene Funktionen und Strukturen zur Bildbearbeitung zur Verfügung. Es wurden zwei Versionen dieser Software für unterschiedliche Verwendungszwecke erstellt.

### 6.1 Program Sensorfusion

Die erste Softwarevariante benutzt das erstellte Konzept zur Sensorfusion, um ein Fusionsbild, die Umgebungsabbilder des Laserscanners sowie eine verzeichnungskorrigierte Kameraaufnahme zu erstellen. Die Software benötigt als Quelldaten einen zeitlich passenden Datensatz des linken und des rechten Laserscannens sowie eine zeitlich entsprechende Kameraaufnahme. Die Scandaten wurden hierfür der Datenbank entnommen und im .txt-Format gespeichert. Die Bilder wurden der Fahrzeugkamera mittels einer Software direkt entnommen und im .jpeg-Format gespeichert. Die Daten werden beim Programmstart geladen. Es muss für jede Datenquelle ein individueller Ladepfad beim Programmstart angegeben werden. Alternativ werden Default-Daten aus einem existierenden Standardverzeichnis geladen. Es ist weiterhin erforderlich ein Speicherverzeichnis zu bestimmen, in dem die Ergebnisse abgelegt werden. Alternativ wird auch hier ein vorhandenes Standardverzeichnis zur Speicherung verwendet.

#### 6.1.1 Funktionen

Public Methodes:

```
void info();
```

Gibt Autor- und Programminformationen aus.

```
int init(string ImageLName, string ImageSPfad, string Lida1Datei,  
string Lida2Datei);
```

Initialisierung mit Speicherpfad für erstellte Abbildungen und Ladepfaden mit Quelldaten.

`void run();`  
Führt nach erfolgreicher Initialisierung die Erstellung der Linsenverzeichnungskorrektur, der Umgebungsabbilder und des Fusionsbildes aus.

`void stop();`  
Setzt Initialisierung zurück und erlaubt eine erneutes ausführen der `init()` Funktion.

Private Methodes:

`Mat imageladen(string s);`  
Lädt eine Grafik im OpenCV Arbeitsformat. `String s` gibt Speicherort und Namen der Datei an.

`Mat imageladeninfo(string s);`  
Lädt eine Grafik im OpenCV Arbeitsformat. Gibt Grafikinformatoren Auflösung und Format aus. `String s` gibt Speicherort und Namen der Datei an.

`Mat verzeichnungskorrekturerstellen(Mat q);`  
Erstellt verzeichnungskorrigierte Grafik aus Kameraaufnahme. `Mat q` ist die zu bearbeitende Grafik im OpenCV Arbeitsformat.

`Mat fusionbilderstellenliste(Mat bild, Laserpunktliste liste);`  
Erstellt Fusionsbild aus verzeichnungskorrigierter Kameraaufnahme und den Scanpunktlisten. `Mat bild` ist die zu bearbeitende Grafik im OpenCV Arbeitsformat. `Laserpunktliste liste` enthält alle Scanpunkte jeder Scanebene.

`Mat fusionbilderstellenbild(Mat bild, Laserbilder lbilder);`  
Erstellt Fusionsbild aus verzeichnungskorrigierter Kameraaufnahme und den Umgebungsabbildern. `Mat bild` ist die zu bearbeitende Grafik im OpenCV Arbeitsformat. `Laserbilder lbilder` enthält die Umgebungsabbilder jeder Scanebene im OpenCV Arbeitsformat.

`string laserdateneinlesen(string dateipfad);`  
Lädt Laserdaten aus einer txt-Datei. `String dateipfad` gibt Speicherort und Namen der Datei an.

`Laserstruckt2202 laserstrucuterstellen(string i);`  
Zerlegt Rohdaten des Laserscanners und ordnet diese in ein `Struckt` zur Weiterverarbeitung ein. `String i` enthält die eingelesenen Rohdaten.

```
Laserbilder laserbildererstellen(Laserstruckt2202 l1,  
Laserstruckt2202 l2);
```

Erstellt Umgebungsabbilder aus zuvor strukturierten Daten des Laserscanners.  
Laserstruckt2202 l1, l2 enthält die Daten des linken bzw. rechten Laserscanners.

```
Laserpunktliste laserpunktlisteerstellen(Laserstruckt2202 l1,  
Laserstruckt2202 l2);
```

Erstellt Scanpunktlisten aus zuvor strukturierten Daten des Laserscanners.  
Laserstruckt2202 l1, l2 enthält die Daten des linken bzw. rechten Laserscanners.

```
void imageanzeigen(Mat i, string s);
```

Zeigt eine Grafik auf dem Monitor an. String s gibt Namen des Anzeigefensters an.  
Mat i ist die anzuzeigende Grafik im OpenCV Arbeitsformat

```
void imagespeichern(Mat i, string s);
```

Speichert eine Grafik im bmp-Format. String s gibt Speicherort und Namen der  
Datei an. Mat i ist die zu speichernde Grafik im OpenCV Arbeitsformat

```
int hilfhexstringindez(string i);
```

Hilfsfunktion zur Übertragung von hexadezimalen Werten in  
dezimale Werte. String s enthält die zu übertragenden Werte.

```
string hilfendian2bytetausch(string s);
```

Hilfsfunktion zur Konvertierung zwischen big und little endian Kodierung. String s  
enthält die zu konvertierenden Werte.

```
Koordinaten hilfscannerkoordinatenerstellen(int angeltprot, int  
angelpoint, int raddistance);
```

Hilfsfunktion zum erstellen der Scanpunkte im Koordinatensystem des  
Laserscanners. Int angeltprot, angelpoint, raddistance enthalten Werte um die  
Koordinaten zu erstellen.

```
Koordinaten hilffahrzeugkoordinatenerstellen(int angeltprot, int  
yawangel, Koordinaten scannerkoord, int montposy);
```

Hilfsfunktion zum übertragen der Scanpunkte vom Koordinatensystem des  
Laserscanners zum Koordinatensystem des Fahrzeugs. Int angeltprot, yawangel,  
montposy enthalten Werte um die Koordinaten zu übertragen. Koordinaten  
scannerkoord enthält zu übertragende Koordinaten.

Private Variables:

`bool start;`

Variable erfasst Initialisierungsstatus.

`Mat kamera;`

`Mat korrektur;`

`Mat fusion;`

Variablen für Grafikbearbeitung nach OpenCV.

`string lida1;`

`string lida2;`

Variablen nehmen Rohdaten der Laserscanner auf.

`string imageLaden;`

`string laser1Laden;`

`string laser2Laden;`

Variablen nehmen bei der Initialisierung die entsprechenden Ladepfade und Dateinamen auf.

`string cameraimageSpeichern;`

`string laserimage0Speichern;`

`string laserimage1Speichern;`

`string laserimage2Speichern;`

`string laserimage3Speichern;`

`string fusionimageSpeichern;`

Variablen nehmen bei der Initialisierung den entsprechenden Speicherort und die Dateinamen auf.

`Laserstruckt2202 srtuctlas1, srtuctlas2;`

Strukturen nehmen die aufbereiteten Daten der Laserscanner auf.

`Laserbilder laserbilder;`

Struktur besteht aus vier Variablen für Grafikbearbeitung nach OpenCV. Struktur nimmt bearbeitete Grafiken der Umgebungsabbilder auf.

### 6.1.2 Klassendiagramm

**Sensorfusion**  
Klasse

Felder

- cameraimageSpeichern
- fusion
- fusionimageSpeichern
- imageLaden
- kamera
- korrektur
- laser1Laden
- laser2Laden
- laserbilder
- laserimage0Speichern
- laserimage1Speichern
- laserimage2Speichern
- laserimage3Speichern
- lida1
- lida2
- srtuctlas1
- srtuctlas2
- start

Methoden

- fusionbilderstellenbild
- fusionbilderstellenliste
- hilfendian.zyretausch
- hilffahrzeugkoordinatenerstellen
- hilfhexstringindez
- hilfscannerkoordinatenerstellen
- imageanzeigen
- imageladen
- imageladeninfo
- imagespeichern
- info
- init
- laserbilderstellen
- laserdateneinlesen
- laserpunktlistenstellen
- laserstrukturstellen
- run
- Sensorfusion
- stop
- verzeichnungskorrekturerstellen

**Laserpunktliste**  
Struktur

Felder

- L0
- L1
- L2
- L3

**KoordinatenSc...**  
Struktur

Felder

- a
- b

**Laserbilder**  
Struktur

Felder

- LaserbildL0
- LaserbildL1
- LaserbildL2
- LaserbildL3

**Koordinaten**  
Struktur

Felder

- a
- b

**Laserstrucktheadr**  
Struktur

Felder

- Datatype
- DeviceID
- Magicword
- NTPtime
- Reserved
- Sizeofpreviousmessages
- Sizeofthismessage

**ScanPoint**  
Struktur

Felder

- Echo
- Echopulsewidth
- Flags
- Horizontalangle
- Layer
- Radialdistance
- Reserved

**Laserstrucktz202**  
Struktur

Felder

- Angleticksperrotation
- Datatype
- DeviceID
- Endangle
- Flags
- Magicword
- Mountingpositionpitchangle
- Mountingpositionrollangle
- Mountingpositionx
- Mountingpositiony
- Mountingpositionyawangle
- Mountingpositionz
- NTPtime
- Pointsl
- Reserved
- ScanendtimeNTP
- Scannerstatus
- Scannumber
- Scanpoints
- ScanstarttimeNTP
- Sizeofpreviousmessages
- Sizeofthismessage
- Startangle
- Syncphaseoffset

## 6.2 Modul Sensorfusion

Die zweite Softwarevariante bildet ein Funktionspaket. Das Paket verwendet das erstellte Konzept zur Sensorfusion um die Fusionsparameter und einige weitere Zwischenelemente, wie die Scanpunktlisten, zu generieren. Diese können im Anschluss anderen möglichen Softwarekomponenten zur Weiterverarbeitung zur Verfügung gestellt werden. Die Softwarekomponente besitzt keine Funktionen zum Laden oder Speichern von Daten. Diese Aufgaben müssen durch separate Komponenten erfüllt werden.

### 6.2.1 Funktionen

Public Methodes:

`void info();`

Gibt Autor- und Programminformationen aus.

`Mat createFusionimg(Mat img, Laserpunktliste list);`

Erstellt Fusionsbild aus verzeichnungskorrigierter Kameraaufnahme und den Scanpunktlisten. `Mat img` ist die zu bearbeitende Grafik im OpenCV Arbeitsformat. `Laserpunktliste list` enthält alle Scanpunkte jeder Scanebene. Funktion prüft die erhaltenden Eingangsdaten.

`Mat createLensecorrectionimg(Mat img);`

Erstellt verzeichnungskorrigierte Grafik aus Kameraaufnahme. `Mat img` ist die zu bearbeitende Grafik im OpenCV Arbeitsformat. Funktion prüft die erhaltenden Eingangsdaten.

`Laserbilder createLaserimg(string l1, string l2);`

Erstellt Umgebungsabbilder aus den Rohdaten des Laserscanners. `string l1, l2` enthält die Daten des linken bzw. rechten Laserscanners. Funktion prüft die erhaltenden Eingangsdaten.

`Laserpunktliste createLaserpointlst(string l1, string l2);`

Erstellt Scanpunktliste aus den Rohdaten des Laserscanners. `string l1, l2` enthält die Daten des linken bzw. rechten Laserscanners. Funktion prüft die erhaltenden Eingangsdaten.

```
Fusionsliste createFusionpointlst(Laserpunktliste list);
```

Erstellt Fusionspunktliste aus den Scanpunktlisten des Laserscanners.

`Laserpunktliste` list enthält die Daten des linken bzw. rechten Laserscanners.

Funktion prüft die erhaltenden Eingangsdaten.

#### Private Methodes:

```
Laserstruckt2202 laserstrucsterstellen(string i);
```

Zerlegt Rohdaten des Laserscanners und ordnet diese in ein Struckt zur

Weiterverarbeitung ein. `String` i enthält die eingelesenen Rohdaten.

```
Laserbilder laserbildererstellen(Laserstruckt2202 l1,  
Laserstruckt2202 l2);
```

Erstellt Umgebungsabbilder aus zuvor strukturierten Daten des Laserscanners.

`Laserstruckt2202` l1, l2 enthält die Daten des linken bzw. rechten Laserscanners.

```
Laserpunktliste laserpunktlisteerstellen(Laserstruckt2202 l1,  
Laserstruckt2202 l2);
```

Erstellt Scanpunktlisten aus zuvor strukturierten Daten des Laserscanners.

`Laserstruckt2202` l1, l2 enthält die Daten des linken bzw. rechten Laserscanners.

```
int hilfhexstringindez(string i);
```

Hilfsfunktion zur Übertragung von hexadezimalen Werten in

dezimale Werte. `String` s enthält die zu übertragenden Werte.

```
string hilfendian2bytetausch(string s);
```

Hilfsfunktion zur Konvertierung zwischen big und little endian Kodierung. `String` s

enthält die zu konvertierenden Werte.

```
Koordinaten hilfscannerkoordinatenerstellen(int angeltprot, int  
angelpoint, int raddistance);
```

Hilfsfunktion zum erstellen der Scanpunkte im Koordinatensystem des

Laserscanners. `int` angeltprot, angelpoint, raddistance enthalten Werte um die  
Koordinaten zu erstellen.

```
Koordinaten hilffahrzeugkoordinatenerstellen(int angeltprot, int  
yawangel, Koordinaten scannerkoord, int montposy);
```

Hilfsfunktion zum übertragen der Scanpunkte vom Koordinatensystem des

Laserscanners zum Koordinatensystem des Fahrzeugs. `int` angeltprot, yawangel,

montposy enthalten Werte um die Koordinaten zu übertragen. `Koordinaten`

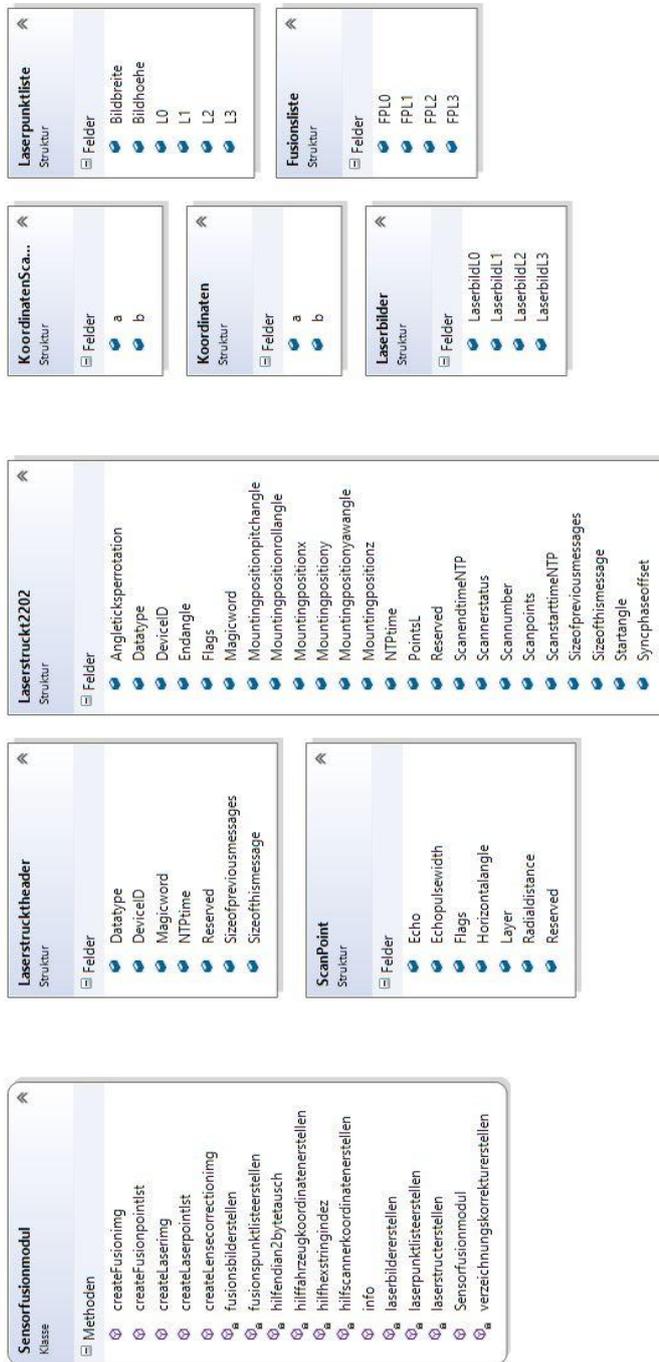
scannerkoord enthält zu übertragende Koordinaten.

**Mat** fusionsbilderstellen(**Mat** bild, **Laserpunktliste** liste);  
Erstellt Fusionsbild aus verzeichnungskorrigierter Kameraaufnahme und den Scanpunktlisten. **Mat** bild ist die zu bearbeitende Grafik im OpenCV Arbeitsformat. **Laserpunktliste** liste enthält alle Scanpunkte jeder Scanebene.

**Mat** verzeichnungskorrekturerstellen(**Mat** q);  
Erstellt verzeichnungskorrigierte Grafik aus Kameraaufnahme. **Mat** q ist die zu bearbeitende Grafik im OpenCV Arbeitsformat.

**Fusionsliste** fusionspunktlisteerstellen(**Laserpunktliste** liste);  
Erstellt Fusionspunktliste aus den Scanpunktlisten des Laserscanners.  
**Laserpunktliste** liste enthält die Daten des linken bzw. rechten Laserscanners.

## 6.2.2 Klassendiagramm



# 7 Quellen

## 7.1 Wissenschaftliche Quellen

- S.Bangadkar, P.Dhane, S. Nair, K. Kutty, Mapping matrix for perspective correction from fish eye distorted images, CREST, KPIT Cummins Infosystems Ltd. Pune, India Juli 2011, Seite: 1288-1292
- Zhengyou Zhang, A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume:22 , Issue: 11 ), August 2008, Seite: 1330-1334
- Christian Lundquist, Sensor Fusion for Automotive Applications, Linköping University Electronic Press, Oktober 2011

## 7.2 Internetquellen

- Abbildungsfehler/Verzeichnung. Webseite, Stand 20. März 2016  
<https://de.wikipedia.org/wiki/Abbildungsfehler>
- Fahrzeugkoordinatensystem. Webseite, Stand 20. März 2016  
<https://de.wikipedia.org/wiki/Fahrzeugkoordinatensystem>
- Entzerrung (Fotografie). Webseite, Stand 20. März 2016  
[https://de.wikipedia.org/wiki/Entzerrung\\_\(Fotografie\)](https://de.wikipedia.org/wiki/Entzerrung_(Fotografie))
- OpenCV Tutorials. Webseite, Stand 20. März 2016  
[http://docs.opencv.org/master/d9/df8/tutorial\\_root.html#gsc.tab=0](http://docs.opencv.org/master/d9/df8/tutorial_root.html#gsc.tab=0)
- OpenCV 2.4.12.0 dokumentation, OpenCV User Guide. Webseite, Stand 20. März 2016  
[http://docs.opencv.org/2.4/doc/user\\_guide/user\\_guide.html](http://docs.opencv.org/2.4/doc/user_guide/user_guide.html)
- PostgreSQL 9.4.6 Documentation. Webseite, Stand 20. März 2016  
<http://www.postgresql.org/docs/9.4/interactive/index.html>
- MathWorks Documentation MATLAB. Webseite, Stand 20. März 2016  
<http://de.mathworks.com/help/matlab/?requestedDomain=www.mathworks.com>

### 7.3 Buchquellen

- Papula Lothar, Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler, 9. Auflage, Friedr. Vieweg & Sohn Verlag | GWV Fachverlag GmbH, Wiesbaden 2006
- Papula Lothar, Mathematik für Ingenieure und Naturwissenschaftler Band 1, 11. Auflage, Vieweg + Teubner Verlag | GWV Fachverlag GmbH, Wiesbaden 2008
- Prof. Dr.-Ing. Andreas Meisel, Vorlesungsskript Robot Vision, 23.03.2012  
Foliensatz: RV03

### 7.4 Bildquellen

- Abb. 7: Perspektivische Transformation  
Prof. Dr.-Ing. Andreas Meisel, Vorlesungsskript Robot Vision, 23.03.2012  
Foliensatz: RV03 Seite: 84

# 8 Anhang

## 8.1 Tabellen

### 8.1.1 Punktliste

Liste von äquivalenten Punkten zwischen verzeichnungskorrigierter Kameraaufnahme und Umgebungsabbild des Laserscanners, für Transformationsparameterbestimmung, separiert nach Scanebenen.

#### Ebene 0

<u>Koordinaten[Pixel]</u>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	<b>P6</b>	<b>P7</b>	<b>P8</b>	<b>P9</b>	<b>P10</b>	<b>P11</b>
Kamera x	776	641	1401	1146	1441	1052	1279	519	868	861	1524
Kamera y	651	679	657	651	720	609	679	657	612	610	765
Laser x	629	632	940	871	865	904	868	560	646	684	861
Laser y	2189	2316	2264	2189	2381	1923	2316	2264	1909	2159	2400

#### Ebene 1

<u>Koordinaten[Pixel]</u>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	<b>P6</b>	<b>P7</b>	<b>P8</b>	<b>P9</b>	<b>P10</b>	<b>P11</b>
Kamera x	776	641	1395	1146	1441	1052	1279	525	868	861	1524
Kamera y	648	675	650	648	713	609	675	650	610	610	754
Laser x	629	632	939	871	865	904	868	561	646	684	861
Laser y	2189	2316	2262	2189	2381	1923	2316	2262	1909	2159	2400

#### Ebene 2

<u>Koordinaten[Pixel]</u>	<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>P5</b>	<b>P6</b>	<b>P7</b>	<b>P8</b>	<b>P9</b>	<b>P10</b>	<b>P11</b>
Kamera x	783	650	1387	1137	1435	1052	1270	533	868	861	1516
Kamera y	620	662	637	620	703	595	662	637	598	601	740
Laser x	629	632	940	871	865	904	868	650	646	684	861
Laser y	2175	2310	2255	2175	2378	1902	2310	2255	1891	2147	2397

Ebene 3

Koordinaten[Pixel]	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Kamera x	784	657	1383	1136	1425	1263	537	1143	861	1497
Kamera y	615	646	631	615	688	646	631	603	601	713
Laser x	629	632	940	871	865	868	560	916	684	861
Laser y	2172	2303	2252	2172	2376	2303	2252	2051	2144	2389

**8.1.2 Gleichungssysteme**

Aus den Punktlisten resultierende Gleichungssysteme, für die Bestimmung der Transformationsparameter, separiert nach Scanebenen.

Ebene 0

	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8	Quellkoord.
P1	629	2189	1	0	0	0	-488104	-1698664	776
	0	0	0	629	2189	1	-1698664	-1425039	651
P2	632	2316	1	0	0	0	-405112	-1484556	641
	0	0	0	632	2316	1	-1484556	-1572564	679
P3	940	2264	1	0	0	0	-1316940	-3171864	1401
	0	0	0	940	2264	1	-3171864	-1487448	657
P4	871	2189	1	0	0	0	-998166	-2508594	1146
	0	0	0	871	2189	1	-2508594	-1425039	651
P5	865	2381	1	0	0	0	-1246465	-3431021	1441
	0	0	0	865	2381	1	-3431021	-1714320	720
P6	904	1923	1	0	0	0	-951008	-2022996	1052
	0	0	0	904	1923	1	-2022996	-1171107	609
P7	868	2316	1	0	0	0	-1110172	-2962164	1279
	0	0	0	868	2316	1	-2962164	-1572564	679
P8	560	2264	1	0	0	0	-290640	-1175016	519
	0	0	0	560	2264	1	-1175016	-1487448	657
P9	646	1909	1	0	0	0	-560728	-1657012	868
	0	0	0	646	1909	1	-1657012	-1168308	612
P10	684	2159	1	0	0	0	-588924	-1858899	861
	0	0	0	684	2159	1	-1858899	-1316990	610
P11	861	2400	1	0	0	0	-1312164	-3657600	1524
	0	0	0	861	2400	1	-3657600	-1836000	765

Ebene 1

	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8	Quellkoord.
<b>P1</b>	629	2189	1	0	0	0	-488104	-1698664	776
	0	0	0	629	2189	1	-1698664	-1418472	648
<b>P2</b>	632	2316	1	0	0	0	-405112	-1484556	641
	0	0	0	632	2316	1	-1484556	-1563300	675
<b>P3</b>	939	2262	1	0	0	0	-1309905	-3155490	1395
	0	0	0	939	2262	1	-3155490	-1470300	650
<b>P4</b>	871	2189	1	0	0	0	-998166	-2508594	1146
	0	0	0	871	2189	1	-2508594	-1418472	648
<b>P5</b>	865	2381	1	0	0	0	-1246465	-3431021	1441
	0	0	0	865	2381	1	-3431021	-1697653	713
<b>P6</b>	904	1923	1	0	0	0	-951008	-2022996	1052
	0	0	0	904	1923	1	-2022996	-1171107	609
<b>P7</b>	868	2316	1	0	0	0	-1110172	-2962164	1279
	0	0	0	868	2316	1	-2962164	-1563300	675
<b>P8</b>	561	2262	1	0	0	0	-294525	-1187550	525
	0	0	0	561	2262	1	-1187550	-1470300	650
<b>P9</b>	646	1909	1	0	0	0	-560728	-1657012	868
	0	0	0	646	1909	1	-1657012	-1164490	610
<b>P10</b>	684	2159	1	0	0	0	-588924	-1858899	861
	0	0	0	684	2159	1	-1858899	-1316990	610
<b>P11</b>	861	2400	1	0	0	0	-1312164	-3657600	1524
	0	0	0	861	2400	1	-3657600	-1809600	754

## Ebene 2

	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8	Quellkoord.
P1	629	2175	1	0	0	0	-492507	-1703025	783
	0	0	0	629	2175	1	-1703025	-1348500	620
P2	632	2310	1	0	0	0	-410800	-1501500	650
	0	0	0	632	2310	1	-1501500	-1529220	662
P3	940	2255	1	0	0	0	-1303780	-3127685	1387
	0	0	0	940	2255	1	-3127685	-1436435	637
P4	871	2175	1	0	0	0	-990327	-2472975	1137
	0	0	0	871	2175	1	-2472975	-1348500	620
P5	865	2378	1	0	0	0	-1241275	-3412430	1435
	0	0	0	865	2378	1	-3412430	-1671734	703
P6	904	1902	1	0	0	0	-951008	-2000904	1052
	0	0	0	904	1902	1	-2000904	-1131690	595
P7	868	2310	1	0	0	0	-1102360	-2933700	1270
	0	0	0	868	2310	1	-2933700	-1529220	662
P8	650	2255	1	0	0	0	-346450	-1201915	533
	0	0	0	650	2255	1	-1201915	-1436435	637
P9	646	1891	1	0	0	0	-560728	-1641388	868
	0	0	0	646	1891	1	-1641388	-1130818	598
P10	684	2147	1	0	0	0	-588924	-1848567	861
	0	0	0	684	2147	1	-1848567	-1290347	601
P11	861	2397	1	0	0	0	-1305276	-3633852	1516
	0	0	0	861	2397	1	-3633852	-1773780	740

Ebene 3

	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8	Quellkoord.
P1	629	2172	1	0	0	0	-493136	-1702848	784
	0	0	0	629	2172	1	-1702848	-1335780	615
P2	632	2303	1	0	0	0	-415224	-1513071	657
	0	0	0	632	2303	1	-1513071	-1487738	646
P3	940	2252	1	0	0	0	-1300020	-3114516	1383
	0	0	0	940	2252	1	-3114516	-1421012	631
P4	871	2172	1	0	0	0	-989456	-2467392	1136
	0	0	0	871	2172	1	-2467392	-1335780	615
P5	865	2376	1	0	0	0	-1232625	-3385800	1425
	0	0	0	865	2376	1	-3385800	-1634688	688
P6	868	2303	1	0	0	0	-1096284	-2908689	1263
	0	0	0	868	2303	1	-2908689	-1487738	646
P7	560	2252	1	0	0	0	-300720	-1209324	537
	0	0	0	560	2252	1	-1209324	-1421012	631
P8	916	2051	1	0	0	0	-1046988	-2344293	1143
	0	0	0	916	2051	1	-2344293	-1236753	603
P9	684	2144	1	0	0	0	-588924	-1845984	861
	0	0	0	684	2144	1	-1845984	-1288544	601
P10	861	2389	1	0	0	0	-1288917	-3576333	1497
	0	0	0	861	2389	1	-3576333	-1703357	713

### 8.1.3 Transformationsparameter

Aus den Gleichungssystemen berechnete Transformationsparameter, separiert nach Scanebenen.

	<b>b11</b>	<b>b12</b>	<b>b13</b>
<b>Ebene0</b>	0.180480068124	-0.380343852372	807.389.568.204.428
<b>Ebene1</b>	0.180939744862	-0.380208417099	806.928.055.129.361
<b>Ebene2</b>	0.159621420744	-0.400049065836	848.299.976.984.453
<b>Ebene3</b>	0.198583033968	-0.388264178555	815.601.786.557.780

	<b>b21</b>	<b>b22</b>	<b>b23</b>
<b>Ebene0</b>	0.002718734328	-0.234902796567	587.741.794.082.016
<b>Ebene1</b>	0.003465895653	-0.234930912065	586.220.979.747.004
<b>Ebene2</b>	0.003063233515	-0.237705330607	581.154.953.036.253
<b>Ebene3</b>	0.005019022913	-0.226165016238	566.457.998.805.325

	<b>b31</b>	<b>b32</b>
<b>Ebene0</b>	0.000000639743	-0.000403924845
<b>Ebene1</b>	0.000000739102	-0.000403879738
<b>Ebene2</b>	0.000000881958	-0.000411466557
<b>Ebene3</b>	0.000000904218	-0.000402452965

## Versicherung über Selbstständigkeit

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, den \_\_\_\_\_