

**Untersuchung und Entwicklung von Algorithmen zur  
Stereobildauswertung für die Erfassung von Objekten im Umfeld von  
Fahrzeugen und Realisierung einer Hindernisdetektion in Echtzeit  
mittels einer Hardwareimplementierung auf einem FPGA**

**Dissertation**

zur Erlangung des akademischen Grades

**Doktoringenieur  
(Dr.-Ing.)**

von Dipl.-Ing. Michael Tornow

geb. am 02.05.1977 in Magdeburg

genehmigt durch die Fakultät für Elektrotechnik und Informationstechnik  
der Otto-von-Guericke-Universität Magdeburg

Gutachter:

Prof. Dr.-Ing. habil. Bernd Michaelis

Prof. Dr. rer. nat. Jörg Kaiser

Promotionskolloquium am 03.03.2009



## Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Elektronik, Signalverarbeitung und Kommunikationstechnik der Otto-von-Guericke-Universität Magdeburg.

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. habil. Bernd Michaelis, der die Anregung für diese Arbeit gab und mir jederzeit mit stetem Interesse, wertvollen Hinweisen und freundlicher Unterstützung zur Seite stand.

Bei Prof. Dr. rer. nat. Jörg Kaiser bedanke ich mich recht herzlich für die Begutachtung dieser Arbeit.

Den Kollegen des Lehrstuhls für Technische Informatik danke ich für die hilfreichen Hinweise und das angenehme Arbeitsklima.

Mein besonderer Dank gilt meinem Zimmerkollegen Jens Kaszubiak für die gute Zusammenarbeit in den gemeinsamen Projekten. Den Mitarbeitern der Institutswerkstatt danke ich für die schnelle Umsetzung der verschiedenen Versuchsaufbauten und die Hilfe bei der Bestückung der Entwurfshardware. Weiterhin danke ich Herrn Dr. Thomas Schindler, Herrn Dr. Gerald Krell und Herrn Helmut Bresch für die vielen wertvollen Tipps und Hinweise bei der praktischen Umsetzung der Arbeit.

Meinen Eltern danke ich für die wertvolle Unterstützung während der Bearbeitungszeit der Dissertation.

Magdeburg, 9. Dezember 2008

Michael Tornow



# Kurzreferat

Die vorliegende Arbeit behandelt die Implementierung von Algorithmen zur Stereobildauswertung in Echtzeit. Dazu wurden verschiedene Algorithmen zur Lösung des Korrespondenzproblems bezüglich einer Implementierung in DSPs und FPGAs entwickelt und untersucht. Nach Betrachtung der Randbedingungen von mehreren Szenarien der Umfelderkennung wurde aufgrund der Echtzeitanforderungen eine FPGA-Implementierung favorisiert. Es wurden zwei Verfahren für eine Implementierung in Echtzeit untersucht und implementiert.

Für die Vermessung der Position von Objekten in einem Messbereich, bei dem eine Dimension deutlich größer als die anderen ist (10 - 150 m; Öffnungswinkel  $30^\circ$ ), wurde ein hierarchisches Verfahren entwickelt. Dabei wird eine Bildpyramide, bei der ebenenweise die Auflösung reduziert wird, aufgebaut und zunächst in jeder Ebene unabhängig eine blockbasierte Korrelation durchgeführt. Im Anschluss werden die Ergebnisse aller Ebenen gegeneinander verifiziert und zusammengeführt, sodass eine gemeinsame Tiefenkarte entsteht. Das Verfahren wurde für die Onlineauswertung der Bilder eines Stereokamerasystems mit  $1024 \times 1024$  Pixel a 10 Bit und 25 Bildern/s implementiert. Als Korrelationskriterien können die quadrierte normierte Kreuzkorrelationsfunktion und die Summe der Differenzenquadrate (SAD) verwendet werden. Mit Hilfe dieses Verfahrens kann eine Reduktion des Rechenaufwandes bei der Stereobildauswertung mit einem großen Disparitätsbereich auf vielen Implementierungsplattformen erreicht werden.

Weiterhin wurde eine Struktur zur Implementierung einer linearen blockbasierten Korrelation in einem FPGA entwickelt. Durch den Aufbau einer Korrelatorkaskade kann eine starke Parallelisierung der linearen Suche realisiert werden. Diese ermöglicht die Anwendung der FPGA-Implementierung der SAD-Funktion in der Stereophotogrammetrie, aber auch in anderen Bildverarbeitungsdisziplinen, wie zum Beispiel der Bewegungsanalyse in einem großen Suchbereich und mit einem hohen Datendurchsatz.

Auf Basis der Korrelatorkaskade wurde eine Hindernisdetektion für ein fahrerloses Fahrzeug realisiert und erfolgreich getestet. Dabei wurden Objekte in einem Messbereich von 1 - 20 m detektiert. Je nach Entfernung zum Messobjekt wurde das Fahrzeug allmählich abgebremst oder sofort gestoppt.



# Abstract

This work covers the implementation of algorithms for stereo image processing in real time. Therefore several different algorithms for solving the correspondence problem are analysed with respect to implementation in DSPs or FPGAs. Based on the boundary conditions of two scenarios of sensing in vehicle or robotic environment real time conditions are determined. Due to these real time conditions an implementation in an FPGA is chosen. Two different procedures for stereo image processing in real time were discussed and implemented.

A hierarchical measurement method for determination of the position of objects in a measurement range with one dimension (10-150 m; angle of aperture  $30^\circ$ ) being explicit bigger than the others was developed. By using an image pyramid, in which the resolution is reduced layer by layer, the depth map is determined. Processing every layer with a block based correlation method and combining the local correlations results over all layers, the disparity for every block is calculated. This hierarchical method was implemented for continuous processing stereo images using a stereo camera system with 1024x1024 Pixel at 10 Bit at 25 images per second in real time. The sum of absolute differences (SAD) and the squared normalised zero mean cross correlation function can be used as similarity criterions. Using this method the calculation effort of the stereo image processing with a big disparity range can be highly reduced on different implementation platforms.

Furthermore a structure for a linear block based correlation in an FPGA was developed. By using a correlator cascade a strong parallelisation of the linear disparity search can be realized. Using an FPGA-implementation of the SAD-function with this structure enables high data throughput and big search areas for the stereo image processing and applications in other image processing fields, e.g. motion analysis.

Based on the correlator cascade an obstacle detection for a driver-less vehicle was realised and tested. In a range from 1 m to 20 m obstacles were detected. Depending on the distance to the obstacle the vehicle was gradually slowed down or immediately stopped.





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Überblick über die Umfelderkennung für Fahrerassistenzsysteme, fahrerlose Transportfahrzeuge und Robotik . . . . .	3
1.2	Bildverarbeitung in eingebetteten Systemen . . . . .	5
1.3	Eingrenzung der Aufgabenstellung . . . . .	6
1.4	Aufbau der Arbeit . . . . .	7
<b>2</b>	<b>Umfelderfassung mittels Stereophotogrammetrie</b>	<b>9</b>
2.1	Optoelektronische Bildaufnahme . . . . .	9
2.2	Methoden zur Bestimmung von 3D-Informationen mit monokularen Aufnahmesystemen . . . . .	12
2.3	Grundlagen der Stereobildauswertung . . . . .	13
2.3.1	Der Normalfall der Stereophotogrammetrie . . . . .	14
2.3.2	Rektifizierung . . . . .	15
2.3.3	Epipolargeometrie . . . . .	16
2.3.4	Fehlerbetrachtung von stereophotogrammetrischen Systemen . . . . .	17
2.4	Korrespondenzproblem . . . . .	18
2.4.1	Ermittlung des Suchbereiches . . . . .	19
2.4.2	Korrespondenzanalyse . . . . .	19
2.4.3	Korrelationskriterien . . . . .	23
2.4.4	Ermittlung dichter kontinuierlicher Tiefenkarten . . . . .	26
2.4.5	Korrespondenzproblem in echtzeitfähigen Stereoauswertesystemen . . . . .	31

2.4.6	Vor- und Nachteile beim Einsatz von Verfahren zur Erzeugung dichter Tiefenkarten . . . . .	32
2.4.7	Bildvorverarbeitung . . . . .	33
2.5	Diskussion der Verfahren zur multikularen 3D-Vermessung . . . . .	34
<b>3</b>	<b>Echtzeitfähige Plattformen für die Stereobildauswertung in eingebetteten Systemen</b>	<b>37</b>
3.1	Datenfluss- und kontrollflussorientierte Plattformen . . . . .	38
3.1.1	Kontrollfluss . . . . .	38
3.1.2	Datenfluss . . . . .	39
3.2	Prozessorbasierte Plattformen . . . . .	39
3.2.1	Universalprozessoren . . . . .	40
3.2.2	Digitale Signalprozessoren . . . . .	42
3.2.3	Alternative Prozessorsysteme . . . . .	44
3.3	Hardwarebasierte Plattformen . . . . .	46
3.3.1	Anwendungsspezifische Schaltkreise . . . . .	46
3.3.2	Programmierbare Logikbausteine . . . . .	48
3.3.3	Vergleich von Hardware- und Softwarekonzepten zur Parallelisierung . . . . .	55
3.4	Arithmetik in der Digitaltechnik . . . . .	58
3.5	Diskussion einer echtzeitfähigen Plattform für die Bildverarbeitung . . . . .	60
<b>4</b>	<b>Konzept zur Echtzeit-Stereobildauswertung</b>	<b>63</b>
4.1	Umfelderfassung mit photogrammetrischer Stereobildauswertung in Echtzeit . . . . .	63
4.2	Einsatzszenarien der Umfelderfassung . . . . .	65
4.2.1	Objektdetektion im Fernbereich . . . . .	65
4.2.2	Objektdetektion im Nahbereich . . . . .	68
4.2.3	Untersuchung der erforderlichen Auflösung in x- und z-Richtung . . . . .	72
4.3	Rechenaufwand von Algorithmen zur Stereobildauswertung . . . . .	77
4.3.1	Korrelationskriterien in Softwareimplementierungen . . . . .	77

4.3.2	Korrelationskriterien in Hardwareimplementierungen . . . . .	79
4.3.3	Vergleich des Rechenaufwandes auf verschiedenen Plattformen . . . . .	84
4.4	Echtzeitfähige Implementierung durch datenflussorientierte Stereobildauswertung	85
<b>5</b>	<b>Umsetzung von zwei Verfahren zur Stereobildauswertung in Echtzeit</b>	<b>89</b>
5.1	Hierarchisches Suchverfahren . . . . .	90
5.1.1	Prinzip des hierarchischen Verfahrens . . . . .	90
5.1.2	Generierung der Ebenen und der Messbereiche . . . . .	91
5.1.3	Position der lokalen Maxima . . . . .	93
5.1.4	Tiefenkarte . . . . .	94
5.1.5	Hardwareimplementierung des hierarchischen Algorithmus . . . . .	95
5.2	Datenflussorientierung durch Anpassung der Implementierung eines linearen blockbasierten Korrelationsverfahrens . . . . .	104
5.2.1	Diskussion der Implementierungsprobleme eines einfachen Blockmatchingalgorithmus . . . . .	104
5.2.2	Datenflussorientierte Implementierung durch eine Korrelatorkaskade . . .	105
5.2.3	Beschreibung eines Kaskadenelementes . . . . .	106
5.2.4	Laufzeitproblematik für die Korrelatorkaskade bei verschiedenen Korrelationskriterien . . . . .	107
5.3	Vergleich der linearen und der hierarchischen blockbasierten Korrelation bei einer Softwareimplementierung . . . . .	108
5.4	Diskussion der Eigenschaften der Implementierungsvarianten für die Stereobild- auswertung . . . . .	109
5.5	Fazit . . . . .	112
<b>6</b>	<b>Praktische Erprobung und Vergleich der Algorithmen</b>	<b>115</b>
6.1	Vermessung im Fernbereich . . . . .	116
6.1.1	Vermessung der Entfernung des Fahrzeugs . . . . .	117
6.1.2	Vermessung der Entfernung der Messtafel . . . . .	118
6.1.3	Auswirkung der vereinfachten Kalibrierung . . . . .	122
6.2	Vermessung im Nahbereich . . . . .	126

6.2.1	Auswertung von Aufnahmen verschiedener Kamerasysteme mit den Verfahren 1 - 4 . . . . .	128
6.2.2	Aufnahme und Auswertung mit dem kommerziellen Sensorsystem "Bumblebee 2" zur Stereovermessung im Nahbereich . . . . .	130
6.3	Diskussion der Eigenschaften der in dieser Arbeit entwickelten Verfahren zur Stereobildauswertung . . . . .	132
<b>7</b>	<b>Umsetzung einer Hindernisdetektion für fahrerlose Transportfahrzeuge</b>	<b>135</b>
7.1	Anforderungen an das System zur Hindernisdetektion . . . . .	136
7.2	Hindernisdetektion mit Hilfe eines statistischen Verfahrens zur Auswertung der Tiefenkarte . . . . .	137
7.2.1	Statistische Clusterverfahren im Vergleich zu anderen Clusterverfahren .	138
7.2.2	Beschreibung des räumlichen statistischen Clusterungsverfahrens . . . . .	138
7.3	Hardwareplattform für die Implementierung der Hindernisdetektion . . . . .	142
7.3.1	Eingliederung in das Gesamtsystem . . . . .	143
7.3.2	Hardware-Software-Co-Design zur Realisierung einer Echtzeitverarbeitung	144
7.3.3	Bildaufnahme und Bildausgabe mittels Kameralink . . . . .	145
7.4	Testfahrt mit Hindernisdetektion auf einem automatisierten Fahrzeug . . . . .	148
7.5	Ergebnisse der Entwicklung einer Hindernisdetektion für fahrerlose Transportfahrzeuge . . . . .	151
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>153</b>
<b>A</b>	<b>Algorithmen der Stereophotogrammetrie und Bildverarbeitung</b>	<b>159</b>
A.1	Innere und äußere Orientierung einer Kamera . . . . .	160
A.2	Der allgemeine Fall der Stereophotogrammetrie . . . . .	162
A.3	Der Normalfall der Stereophotogrammetrie . . . . .	164
A.4	Rektifizierung . . . . .	166
A.5	Epipolargeometrie . . . . .	167
A.6	Kalibrierung und Fehlerbetrachtung von stereophotogrammetrischen Systemen .	168
A.7	Fehlerbetrachtung von stereophotogrammetrischen Systemen . . . . .	170

<b>B Vergleich der Eigenschaften verschiedener Korrelationsverfahren</b>	<b>173</b>
B.1 Vorstellung verschiedener Korrelationskriterien . . . . .	173
B.2 Rechenaufwand von Korrelationskriterien . . . . .	176
B.3 Eigenschaften verschiedener Korrelationskriterien und Untersuchung der Subpixelgenauigkeit . . . . .	186
B.4 Untersuchung der Korrelationskriterien mit realen Bildern . . . . .	192
B.5 Diskussion . . . . .	195
<b>C Messergebnisse</b>	<b>199</b>
<b>D Korrektur systematischer Fehler im Stereonormalfall</b>	<b>211</b>
<b>Literaturverzeichnis</b>	<b>217</b>



# Abbildungsverzeichnis

2.1	Ermittlung der 3D-Position im Normalfall der Stereophotogrammetrie . . . . .	15
2.2	Rektifizierung eines Stereobildpaares . . . . .	16
2.3	Epipolargeometrie im Normalfall der Stereophotogrammetrie . . . . .	17
2.4	Beispiel einer Tiefenkarte . . . . .	19
2.5	Korrespondenzanalyse mittels neuronaler Netze [31] . . . . .	21
2.6	Organisation des Blockmatching . . . . .	21
2.7	Klassische Bildpyramide . . . . .	22
3.1	Blockschaltbild der ALU des Festkomma-DSPs BF533 [8] . . . . .	43
3.2	Logikelement eines FPGA; APEX20K-Reihe von Altera [22] . . . . .	49
3.3	LAB-Struktur der APEX20K-Reihe von Altera [22] . . . . .	50
4.1	Optischer Vergleich der Abbildung des Standardmessobjektes mit verschiedenen Brennweiten . . . . .	71
4.2	Variation der Disparität aufgrund des optischen Einflusses der Brennweite . . . . .	73
4.3	Sequentielle Implementierung der SAD . . . . .	79
4.4	Parallele Implementierung der SAD . . . . .	81
4.5	Parallele Implementierung der QNKKFMF . . . . .	82
4.6	Rechenaufwand der parallelen Implementierung von Korrelationskriterien bei der Variation der Blockgröße (8 - 64 Pixel) . . . . .	83
4.7	Rechenaufwand der parallelen Implementierung von Korrelationskriterien in Abhängigkeit von der Bitbreite im Bereich (8 - 24 Bit) der Eingangswerte (qualitativ) . . . . .	84
5.1	Generierung der Ebenen durch Mittelwertbildung . . . . .	92
5.2	Typisches Beispiel einer Korrelationsfunktion (KKFMF) . . . . .	93

## ABBILDUNGSVERZEICHNIS

5.3	Auswahl der Blöcke zum Vergleich der lokalen Korrelationsmaxima . . . . .	94
5.4	Beispiele für die Zusammenführung der Ebenen . . . . .	96
5.5	Modularer Aufbau . . . . .	97
5.6	Struktur zur Generierung der Ebenen in Echtzeit . . . . .	99
5.7	Zuführung der mittelwertfreien Daten mittels des gleitenden Mittelwertes . . . . .	100
5.8	Erstellung eines Datensatzes . . . . .	101
5.9	Dispsearch und Subpixelinterpolation in Hardware . . . . .	102
5.10	Ermittlung der dominierenden Maxima in den überlappenden Blöcken der Ebene mit Hilfe der Korrelationswerte . . . . .	103
5.11	Schematische Darstellung der Korrelatorkaskade . . . . .	105
5.12	Blockschaltbild der kaskadierbaren SAD-Funktion . . . . .	106
5.13	Blockschaltbild der kaskadierbare VHDL-Funktion "Dispsearch" . . . . .	107
6.1	Linkes Kamerabild bei der Vermessung des Fahrzeugs in zwei Entfernungen . . . . .	116
6.2	Verlauf der Entfernungbestimmung eines Fahrzeugs von 10 - 150 m . . . . .	117
6.3	Abs. Fehler der Position eines Fahrzeugs im Fernbereich . . . . .	119
6.4	Rel. Fehler der Position eines Fahrzeugs im Fernbereich . . . . .	120
6.5	Theoretischer Verlauf des rel. Fehler für das hierarchische Verfahren . . . . .	121
6.6	Messtafel mit hohem Kontrast . . . . .	122
6.7	Vergleich der Entfernungvermessung der Messtafel im kalibrierten und nicht kalibrierten Fall . . . . .	123
6.8	Abs. Fehler der Position der Messtafel im Fernbereich . . . . .	124
6.9	Abs. Fehler der Position der Messtafel im Fernbereich kalibriert . . . . .	125
6.10	verwendete Kamerasysteme . . . . .	126
6.11	Vergleich der Auswirkungen von Basisbreite und Brennweite auf die Messung . . . . .	127
6.12	Mittelwert der Entfernungsmessung im Nahbereich mit dem Kamerasystem 1 . . . . .	129
6.13	Mittelwert der Entfernung bei der Auswertung mit den Algorithmen des Bumblebee2-Systems . . . . .	131
7.1	Anpassung des Detektionsbereiches an den Fahrkorridor . . . . .	137
7.2	Clustering im räumlichen Histogramm . . . . .	139



7.3	Reduktion auf ein 2D-Histogramm zur Clusterung . . . . .	140
7.4	Ablauf des Clusterverfahrens . . . . .	141
7.5	Ausgabe der Ergebnisse der Clusterung durch das Messsystem . . . . .	142
7.6	Hardwareplattform . . . . .	143
7.7	Aufteilung des Algorithmus auf Hard- und Software . . . . .	144
7.8	Programm FPGA-Docking-Station zur Anpassung von Parametern zur Laufzeit	149
7.9	Testfahrt ohne Hindernis in der Fahrstrecke . . . . .	150
7.10	Testfahrt mit Hindernis in der Fahrstrecke . . . . .	151
A.1	Elemente der inneren Orientierung am Kameramodell . . . . .	160
A.2	Elemente der inneren und äußeren Orientierung . . . . .	161
A.3	Ermittlung der 3D-Position im allgemeinen Fall der Stereophotogrammetrie . . . . .	163
A.4	Ermittlung der 3D-Position im Normalfall der Stereophotogrammetrie . . . . .	165
A.5	Rektifizierung eines Stereobildpaares . . . . .	166
A.6	Epipolargeometrie im allgemeinen Fall der Stereophotogrammetrie . . . . .	167
A.7	Epipolargeometrie im Normalfall der Stereophotogrammetrie . . . . .	167
B.1	Struktur einer SAD-Funktion mit sequentieller Abarbeitung . . . . .	177
B.2	Struktur einer SSD-Funktion mit sequentieller Abarbeitung . . . . .	177
B.3	Struktur einer SAD-Funktion mit paralleler Abarbeitung . . . . .	178
B.4	Struktur einer SSD-Funktion mit paralleler Abarbeitung . . . . .	179
B.5	Realisierung des gleitenden Mittelwertes in Hardware . . . . .	180
B.6	Struktur der sequentiellen Implementierung der QNKKFMF . . . . .	182
B.7	Struktur der parallelen Implementierung der QNKKFMF . . . . .	184
B.8	Synthetisches Bildpaar (IM26) mit einer schmalen Sinushalbwellen und steigender Disparität . . . . .	186
B.9	Korrelationsfunktionen angewendet auf Zeile 600 des Bildes IM26 . . . . .	187
B.10	Anwendung der $SAD_{LoG}$ - und der $SSD_{LoG}$ -Funktion auf das Bildpaar IM26 (Zeile 600) . . . . .	188
B.11	Vergleich der Fehler der Subpixelgenauen Disparität des Bildpaares IM26 . . . . .	189

## ABBILDUNGSVERZEICHNIS

B.12 Synthetisches Bildpaar (IM26) mit einer breiten Sinushalbwellen- und steigender Disparität . . . . .	189
B.13 Korrelationsfunktionen angewendet auf Zeile 600 des Bildes IM23 . . . . .	190
B.14 Anwendung der $SAD_{LoG}$ - und der $SSD_{LoG}$ -Funktion auf das Bildpaar IM23 (Zeile 600) . . . . .	191
B.15 Vergleich der Fehler der Subpixelgenauen Disparität des Bildpaares IM23 . . . . .	191
B.16 Reales Szene mit Betrachtung mehrerer Ziele . . . . .	192
B.17 Fehler der Subpixeldisparität für das erste Objekt . . . . .	193
B.18 Kurvenverläufe der Korrelationskriterien für das zweite Objekt . . . . .	194
B.19 Fehler der Subpixeldisparität für den schwarzen PKW . . . . .	195
B.20 Kurvenverläufe der Korrelationskriterien für das dritte Objekt . . . . .	196
B.21 Fehler der Subpixeldisparität für den grauen PKW im Vordergrund (drittes Objekt)(links QNKKFMF; rechts SAD . . . . .	197
C.1 Messobjektes: Messtafel . . . . .	199
C.2 Verlauf der Entfernung bei der Vermessung der Messtafel . . . . .	200
C.3 Abs. Fehler der Position eines Messobjektes im Fernbereich . . . . .	201
C.4 Rel. Fehler der Position eines Messobjektes im Fernbereich . . . . .	202
C.5 (kalibriert) Verlauf der Entfernung . . . . .	203
C.6 Abs. Fehler der Position eines Messobjektes im Fernbereich . . . . .	204
C.7 Rel. Fehler der Position eines Messobjektes im Fernbereich . . . . .	205
C.8 Vergleich der Auswirkungen von Basisbreite und Brennweite auf die Messung . . . . .	207
C.9 Mittelwert der Entfernungsmessung im Nahbereich mit dem Kamerasystem 1 . . . . .	208
C.10 Absoluter Fehler der Entfernung im Nahbereich mit Photonfokus-Kameras . . . . .	209
C.11 Mittelwert der Entfernungsmessung im Nahbereich mit dem Kamerasystem 2 . . . . .	210
C.12 Absoluter Fehler der Entfernung im Nahbereich mit Bumblebee 2 . . . . .	210
D.1 Absolutfehler von Z nach der Korrektur . . . . .	214
D.2 Disparität bei lateralem Versatz des Messobjektes . . . . .	215
D.3 Absolutfehler von X vor der Korrektur . . . . .	216
D.4 Absolutfehler von X nach der Korrektur . . . . .	216

# Tabellenverzeichnis

4.1	Echtzeitanforderung: Vergleich des Einflusses verschiedener Geschwindigkeiten . . . . .	67
4.2	Größe verschiedener Messobjekte in unterschiedlicher Entfernung . . . . .	74
4.3	Vergleich der Echtzeitanforderungen und Messbereiche der Szenarien . . . . .	76
4.4	Übersicht des Rechenaufwandes von Korrelationskriterien in Software . . . . .	78
4.5	Vergleich des Hardwareaufwands verschiedener Korrelationskriterien . . . . .	80
5.1	Messbereichsübersicht für ein Beispielsystem . . . . .	93
5.2	Vergleich des Rechenaufwandes von Korrelationskriterien beim linearen und beim hierarchischen Algorithmus in Software . . . . .	109
5.3	Vergleich der hardwareimplementierten Algorithmen zur Stereobildauswertung . . . . .	111
B.1	Vergleich der Ressourcen von SAD und SSD . . . . .	179



# Abkürzungs-, Symbol- und Formelzeichenverzeichnis

## Abkürzungsverzeichnis

3D	dreidimensional
ABEL	Advanced Boolean Express Language
ABS	Anti-Blockier-System
ACC	Adaptive-Cruise-Control
AHDL	Altera Hardware Description Language
ADD	Addierer/Addition
ALU	Arithmetik-Logic-Unit
APS	Actived-Pixel-Sensor
ASIC	Application Specific Integrated Circuit
B	Betrag (einer Zahl)
CAN	Controller Area Network
CCD	Charged Coupled Devices
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex PLD
CISC	Complex Instruction Set Computing
DIV	Division
DMA	Direct Memory Access
DPR	Dual Port RAM
DSP	Digitaler Signalprozessor
EEPROM	Electrical EPROM
ESP	Elektronisches Stabilitätsprogramm
EPROM	Eraseable PROM
FF	FlipFlop
FFT	Fast-Fourier-Transformation
FIR	Far Infrared Range
FPGA	Field Programable Gate Array
FTF	fahrerloses Transportfahrzeug
GAL	Gate-Array-Logic

## ABKÜRZUNGS-, SYMBOL- UND FORMELZEICHENVERZEICHNIS

GHz	Giga-Herz
GPU	Grafic-Processing-Unit
HDL	Hardware Description Language
HDR	High-Dynamic-Range
hier.	hierarchisch (-es Verfahren)
IEEE	Institute of Electrical and Electronical Engineers
IESK	Institut für Elektronik, Signalverarbeitung und Kommunikationstechnik an der Universität Magdeburg
IO	Input/Output
IP	Interlectual Property
KKF	Kreuzkorrelationsfunktion
LAB	Logic Array Block
LE	Logikelement
LKW	Lastkraftwagen
LoG	Laplacian of Gaussian
LSB	Least Significant Bit
LUT	Look-Up-Table
LVDS	Low-Voltage-Differential-Signal
MAC	Multiply and Accumulate
max.	maximal
MByte	MegaByte (1024 x 1024 Byte)
MHz	Mega-Herz
MIMD	Multiple Instruction Multiple Data
min.	minimal
MIR	Medium Infrared Range
MISD	Multiple Instruction Single Data
MM	Mannmonate
MOR	Moravec-Kriterium
MSB	Most Significant Bit
MUX	Multiplexer
NIR	Near Infrared Range
NKKFMF	Normierte mittelwertfreie Kreuzkorrelationsfunktion
NSSDMF	Normierte mittelwertfreie Summe der Differenzenquadrate
par.	parallel
PAL	Programmable Array Logic
PC	Personal Computer
PCI	Peripheral Component Interconnect
Perf.	Performanze
Pipel.	Pipelining
PLA	Programmable Logic Array
PLD	Programmable Logic Device
PLL	Phased Locked Loop
PROM	Programmable Read Only Memory

PMD	Photonic Mixer Device
PPE	Power Processing Element
PKW	Personenkraftwagen
Q	Quadrierung
QNKKFMF	quadrierte NKKFMF
RAM	Random Access Memory
ROM	Read Only Memory
REG	Register
RISC	Reduced Instruction Set Computing
SAD	Summe der absoluten Differenzen
SADMF	mittelwertfreie SAD
seq.	sequentiell
SIMD	Single Instruction Multiple Data
SISD	Single Instruction Single Data
SoC	System on a Chip
SoPC	System on a Programmable Chip
SPE	Synergetic Processing Element
SRAM	static RAM
SSD	Summe der Differenzenquadrate
SSDMF	mittelwertfreie SSD
SSE	Streaming SIMD Extensions
USB	Universal Serial Bus
UML	Unified Markup Language
UV	Ultraviolett
Verilog	Verifying Logic (Hardware Description Language)
VHDL	V(ery High Speed Integrated Circuit) Hardware Description Language
VHDL-AMS	VHDL-Erweiterung zur Beschreibung von analogen Schaltungen

## Symbol- und Formelzeichenverzeichnis

$\lambda$	Wellenlänge der elektromagnetischen Strahlung
$\Delta u$	Disparität
$\Delta u_r$	subpixelgenaue Disparität
$x, x', x''$	Bildkoordinaten in horizontaler Richtung
$y, y', y''$	Bildkoordinaten in vertikaler Richtung
$X$	Weltkoordinaten laterale Koordinate
$Y$	Weltkoordinaten Höhe
$Z$	Weltkoordinaten Entfernung
$P$	Objektpunkt definiert durch $P(X, Y, Z)$
$C_{(0)}$	aktueller Korrelationswert
$C_{(1)}$	Nachfolger des Korrelationswertes
$C_{(-1)}$	Vorgänger des Korrelationswertes

## ABKÜRZUNGS-, SYMBOL- UND FORMELZEICHENVERZEICHNIS

$b$	Basisbreite
$f$	Brennweite
$c$	Kamerakonstante $c = f/Pixelbreite$
$x_n, y_n$	normierte Bildkoordinaten rektifizierter Stereobilder
$O', O''$	Projektionszentren der Kameras
$\sigma_z$	Standardabweichung der Entfernung
$\sigma_{\Delta u}$	Standardabweichung der Disparität
$i, j$	Pixelposition innerhalb des Blockes
$m, n$	Blockgröße
$\xi, \eta$	Verschiebung des Suchblock gegenüber dem Referenzblock in x- und y-Richtung
$P_r(x, y)$	Referenzblockes für die Pixelposition $x, y$
$F(x + \xi, y + \eta)$	Suchblock für die Pixelposition $x + \xi, y + \eta$
$P_r(i, j)$	Pixel des Referenzblockes
$F(\xi + i, \eta + j)$	Pixel des Suchblockes
$\overline{P_r(i, j)}$	mittelwertfreier Pixel des Referenzblockes
$\overline{F(\xi + i, \eta + j)}$	mittelwertfreier Pixel des Suchblockes
$\Delta g$	Laplacien of Gaussian
$n_t$	Anzahl der erforderlichen Testvektoren
$m_e$	Anzahl der Eingänge einer digitalen Schaltung
$m_s$	Anzahl der Speicher einer digitalen Schaltung
$x_{2gl}$	binäre Zahl im Gleitkommaformat
$m_z$	Mantisse der Gleitkommazahl
$n_z$	Exponent der Gleitkommazahl
$\alpha$	Formatwinkel des Objektivs
$n$	Anzahl der zusätzlichen Bildebene im hierarchischen Verfahren
$E_l(x_{max}, y_{max})$	Ebene $l$ beim hierarchischen Verfahren mit der Auflösung $x_{max}, y_{max}$
$l$	Nummer der Ebene
$x_l$	x-Koordinate eines Blockes der Ebene
$\nu_l$	positionsbezogene Nummer des Referenzblockes
$x'$	x-Koordinate in der Originalauflösung
$u'$	Disparität in der Originalauflösung
$u_l$	Disparität eines Referenzblockes in der Ebene <i>ebene</i>
$u_{min}$	minimale Disparität für Fälle mit reduzierter Auflösung der Disparität (im Regelfall Null)







# Kapitel 1

## Einleitung

Der Mensch wird heute in vielen Bereichen seines Lebens durch verschiedenartige Technik bei z.B. ermüdenden oder gefährlichen Tätigkeiten unterstützt. Für die Zukunft werden viele Innovationen auf dem Gebiet der autonomen Systeme, wie z.B. PKWs, die ihre Insassen selbstständig und sicher zu ihrem Ziel bringen oder Haushaltsroboter, die uns bei den Arbeiten des täglichen Lebens unterstützen, propagiert. Dies sind Systeme, die ohne menschliche Eingriffe spezielle Aufgaben lösen können.

Seit 1970 werden in Japan, Deutschland, USA u.a. selbstständig fahrende PKWs untersucht. Diese Untersuchungen mündeten in Deutschland im EUREKA Prometheus Projekt [93], bei dem im Zeitraum von 1987 bis 1995 automatische Fahrzeuge von Daimler-Benz und von der Bundeswehrhochschule München entwickelt wurden, die auf Autobahnen mehrere hundert Kilometer automatisch gesteuert zurücklegten [94].

Die Ergebnisse dieser Forschungen sind in die Weiterentwicklung von fahrerlosen Transportfahrzeugen (FTF) eingeflossen. Sie werden gegenwärtig überwiegend durch Spurführungssysteme geleitet und sind je nach Geschwindigkeit mit Laserscannern und optomechanischen Sicherheitssystemen ausgestattet. Andererseits haben diese Forschungen Einfluss auf die aktiven Sicherheitssysteme in PKWs und LKWs gehabt. Während die erste Generation aktiver Sicherheitssysteme mit dem Anti-Blockier-System (ABS) und dem Elektronischen Stabilitäts-Programm (ESP) sich auf das Fahrzeug und die sichere Straßenführung bezog, nimmt die zweite Generation das direkte Umfeld des Fahrzeuges wahr und reagiert darauf. Beispiele hierfür sind die Adaptive Cruise Control (ACC) bei der auf Basis von Radar- oder Kamerasensoren der Straßenabschnitt vor einem Fahrzeug nach anderen Fahrzeugen untersucht und die Geschwindigkeit angepasst wird, oder der Spurhalteassistent, bei dem kamerabasiert die Spurmarkierungen detektiert werden und der Fahrer davor gewarnt wird, wenn er die Fahrspur verlässt.

Um diese Aufgaben zu bewältigen, müssen die sensorischen Fähigkeiten für technische Systeme steigen. Ein Großteil unserer Umwelt ist auf die visuelle Wahrnehmung ausgerichtet. Daher ist das maschinelle Sehen ein sehr wichtiges sensorisches System für aktuelle und zukünftige technische Geräte.

Um ein System für maschinelles Sehen für die Anwendungsfelder Fahrerassistenzsysteme und fahrerlose Transportfahrzeuge realisieren zu können, müssen die Daten zur Berechnung der 3D-Punkte für das Umfeld mit erforderlicher Genauigkeit innerhalb einer kurzen Reaktionszeit erfasst werden. Um die Messgenauigkeit zu gewährleisten, wurde auf das Verfahren der Stereophotogrammetrie zurückgegriffen, da über die Aufnahme und Auswertung eines Stereobildes die exakte Bestimmung der Position eines Objektes im 3D-Raum möglich ist.

Die digitale Stereobildauswertung hat sich seit 1970 durch die schnellen Fortschritte in der Technologie der Mikroelektronik deutlich weiterentwickelt. Zunächst wurden digitalisierte Fotografien mit Hilfe der Rechentechnik ausgewertet. Mit der Entwicklung der CCD-Technologie war es möglich, Bilder elektronisch aufzunehmen und anschließend direkt digital auszuwerten. Aufgrund des hohen Datenaufkommens bei Stereobildern dauerte die Auswertung eines Bildes in den Anfängen mitunter mehrere Stunden.

Die Fortschritte der letzten 15 Jahre auf dem Gebiet der Mikroelektronik ermöglichen die zeitnahe und kontinuierliche Aufnahme und Auswertung der Bilder von Kameras in einem Automatisierungsumfeld. Dies führte im vergangenen Jahrzehnt zu einer stärkeren Nutzung kamerabasierter Systeme zur Steuerung und zur Qualitätskontrolle. In den letzten 5 Jahren sind die technischen Möglichkeiten von kamerabasierten Systemen weiter stark gestiegen. Dabei sind eine hohe Auflösung von Kameras durch die Miniaturisierung, kürzere Rechenzeiten durch leistungsfähige Rechnersysteme und ein niedriger Energieverbrauch durch Einsatz der CMOS-Technologie entscheidend.

Bedingt durch das Szenario, in dem ein System eingesetzt wird, ergibt sich eine notwendige Reaktionszeit auf Situationen im Umfeld des Fahrzeuges und damit eine erforderliche Rechengeschwindigkeit. Die maximale zulässige Reaktionszeit wird mit dem Begriff Echtzeitforderung bezeichnet. Für die Szenarien Spurwechselassistent und fahrerlose Transportfahrzeuge ergeben sich Echtzeitforderungen mit sehr kurzen Reaktionszeiten (max. 40 - 100 ms).

Die Kombination von Echtzeitverarbeitung bei einem hohem Rechenaufwand der Stereobildauswertung bedingt ein leistungsfähiges Rechensystem, welches für den Einsatz in PKWs oder fahrerlosen Fahrzeugen mittels eines eingebetteten Systems umgesetzt werden muss.

In dieser Arbeit werden Algorithmen und Verfahren zur Stereobildauswertung anhand der Szenarien Spurwechselassistent im PKW und Hinderniserkennung bei fahrerlosen Transportfahrzeugen untersucht und eine FPGA<sup>1</sup>-basierte Implementierung der Bildverarbeitungsalgorithmen davon abgeleitet und praktisch umgesetzt.

---

<sup>1</sup>Field Programmable Gate Array beschrieben in Abschnitt 3.3.2

## 1.1 Überblick über die Umfelderfassung für Fahrerassistenzsysteme, fahrerlose Transportfahrzeuge und Robotik

Mit Komfortfunktionen wie z.B. DISTRONIC bzw. ACC und Einparkhilfen sind erste Umfelderfassungssysteme für Fahrerassistenzsysteme bereits in der Serienproduktion. Dafür kommen vorwiegend Radar- und Ultraschallsensoren zum Einsatz. Für die Automatisierung von Transportaufgaben in Fabriken oder in Krankenhäusern werden fahrerlose Transportfahrzeuge seit einigen Jahren eingesetzt. Zur Gewährleistung der Sicherheit wurden Umfelderfassungssysteme auf der Basis von Laserscannern und Ultraschallsensoren verwendet, um Fahrzeuge bei Auftreten eines Hindernisses im Fahrweg automatisch zu stoppen. Fahrerlose Transportfahrzeuge und ihre Umfelderfassungssysteme sind heute vorwiegend für den Einsatz in geschlossenen Gebäuden konzipiert. Für die Fahrerassistenzsysteme, aber auch für FTF außerhalb von Gebäuden werden Sensorprinzipien für den Außenbereich bzw. für den gemischten Betrieb erforderlich. Häufig sind die Situationen im Außenbereich durch eine Vielzahl von Objekten und deutlich größere Messbereiche erheblich komplexer als im Innenbereich.

Je nach Anforderung sind die äußeren Einflüsse, aber auch Randbedingungen wie der Öffnungswinkel des Messsystems sehr unterschiedlich. Fahrerlose Transportfahrzeuge sind häufig mit Sensoren (z.B. 180° Laserscanner) ausgestattet, die den gesamten Bereich vor dem Fahrzeug abdecken. Ebenso wird durch eine gute Verteilung der Ultraschallsensoren für Einparkhilfen bei PKWs angestrebt, eine lückenlose Abdeckung des Heckraumes zu erreichen. Systeme für hohe Geschwindigkeiten, wie z.B. DISTRONIC oder auch ACC, aber auch Spurwechselassistenzsysteme, erfordern einen Messbereich mit einer großen Ausdehnung in der Entfernung.

Durch die Vielzahl von Sensorprinzipien, die für eine Umfelderfassung in Frage kommen, können angemessene Lösungen für jeden Anwendungsfall realisiert werden. Interessant für die Umfeld- und Situationserfassung bei Fahrzeugen sind Radar, Ultraschall, Laser und aktive wie passive optische Messverfahren, zu denen auch die kamerabasierten Messverfahren zählen. Jedes Messprinzip hat hierbei spezifische Vor- und Nachteile. Daher ist ein Trend zur Kombination mehrerer Messprinzipien erkennbar, um die Schwächen des einzelnen Systems auszugleichen.

Die meisten Messprinzipien arbeiten aktiv auf Basis der Laufzeitmessung eines ausgesendeten Signals. Ultraschallsensoren sind für relativ kurze Entfernungen gut geeignet. Bei einem Messbereich von wenigen Metern können in Innenräumen hiermit sehr genaue Entfernungsmessungen durchgeführt werden [92]. Im Außenbereich werden die Messergebnisse der Ultraschallsensoren von Umwelteinflüssen z.B. Wind gestört [52] und entsprechend verschlechtert. Eine vollständige Umfelderfassung ist hier jedoch nur mit vielen Sensoren möglich, da der Öffnungswinkel eines Sensors relativ klein ist.

Radarsensoren werden für den Nahbereich (24 GHz) und Fernbereich (76 GHz) eingesetzt. Sie bieten ebenfalls nur einen schmalen Öffnungswinkel. Eine vollständige Umfelderfassung ist nur durch das Schwenken der Radarantenne, wie in der Luftfahrt üblich, oder durch den Einsatz mehrerer Radarsensoren möglich. Vorteil der Radarsensoren gegenüber anderen Messverfahren ist die Witterungsunabhängigkeit. Die Sicht eines Radarsensors wird nur durch physische

Hindernisse eingeschränkt. In Abhängigkeit der Reflexions- und Absorptionseigenschaften des Hindernisses für elektromagnetische Strahlung wird es vom Radarsensor erkannt. Für Bodenfahrzeuge werden derzeit Systeme mit Messentfernungen von wenigen Metern bis zu mehreren hundert Metern angeboten. Radarsensoren besitzen im Vergleich zu Kamerasensoren eine verhältnismäßig geringe laterale Auflösung und stellen durch ihre starke elektromagnetische Strahlung eine Störquelle für technische Systeme dar. Sie haben negative Wirkungen auf Lebewesen [99].

Laserscanner werden bereits seit einigen Jahren zur Umfelderkennung für fahrerlose Transportfahrzeuge verwendet. Ihr Einsatz für Fahrerassistenzsysteme wird ebenfalls untersucht. Auch sie basieren auf der Laufzeitmessung und messen jeweils immer einen 3D-Punkt zu einem Zeitpunkt mit relativ hoher Genauigkeit. Um eine Umfelderkennung durchführen zu können, ist ein mechanisch bewegter Ablenkspiegel oder ein anderweitig bewegtes System erforderlich [29]. Die Messentfernungen liegen im Bereich weniger Meter bis einige hundert Meter. Begrenzt wird die Messentfernung im öffentlichen Raum durch die Laserleistung, welche bei hohen Werten eine schädliche Wirkung auf biologische Sehorgane hat.

Seit einigen Jahren werden aktive optische Laufzeitverfahren [51], [91] untersucht, bei denen ein flächiges Tiefenbild mit einem optischen Sensor ermittelt wird. Ein Beispiel hierfür sind die PMD-Sensoren, bei denen mit Hilfe einer zeitlichen Intensitätsvariation über die Reflexion die Entfernung zu einem Objekt ermittelt werden kann. Auch hier tritt das Problem der Leistungsbegrenzung zum Schutz des menschlichen Auges auf, wodurch die Messentfernung auf wenige Meter begrenzt wird.

Im Mittelpunkt dieser Arbeit steht die Vermessung durch passive bildgebende Sensoren. Aufgrund der vorwiegend auf visuelle Wahrnehmung ausgerichteten Umwelt des Menschen haben kamerabasierte Systeme, insbesondere Systeme auf Basis der Stereophotogrammetrie den Vorteil, dass sie mit entsprechendem Aufwand auf die gleichen Informationen zurückgreifen können.

Bei den kamerabasierten Messverfahren unterscheidet man monokulare und multikulare Verfahren. Die monokularen Verfahren ermitteln die Entfernung im Wesentlichen relativ. Sie betrachten entweder die zeitliche Veränderung der Abbildung eines Objektes bei Variation eines äußeren Parameters (z.B. optischer Fluss) [65] oder beruhen auf der Abschätzung der Entfernung durch a priori Wissen über die Dimensionen des zu vermessenden Objektes, z.B. eine standardisierte Messlatte. Sind die Messobjekte jedoch nicht bekannt oder veränderlich, wie dies oft bei der Umfelderkennung der Fall ist, dann kommt es zu Fehlmessungen. In kritischen Situationen sind diese Verfahren zur Situationserfassung und -vermessung nur bedingt geeignet.

Bei der Verwendung von mehreren Kameras können genaue Messungen der Entfernung durchgeführt werden [69]. Ein Stereokamerasystem ist das kleinste Multikamerasystem zur photogrammetrischen Entfernungsvermessung. Automatische Auswerteverfahren für Mehrkamerasysteme sind sehr rechenaufwändig und werden im Regelfall in Software auf Workstations realisiert. Die Auswertung der Informationen aus Bildern erfolgt mit Algorithmen der digitalen Bildverarbeitung. Die kontinuierliche Verarbeitung von Videodaten in Echtzeit ist eine sehr rechenaufwändige Aufgabe. Dies resultiert zum einen aus der großen Menge an Daten, aber

auch aus den Algorithmen, die teilweise während der Auswertung ein Mehrfaches des originalen Datenaufkommens produzieren. Dieses Datenaufkommen kann vorwiegend aus Gründen der Datenerfassung und der Speicherhierarchie nur bedingt oder nicht in Echtzeit auf aktuellen PC-Workstations abgearbeitet werden. In autonomen Systemen werden jedoch üblicherweise eingebettete Systeme verwendet, die eine geringere Leistungsaufnahme, jedoch auch eine geringere Rechenleistung haben.

## 1.2 Bildverarbeitung in eingebetteten Systemen

Um eine Stereobildauswertung als Sensorsystem für die Spurwechselassistenten oder fahrerlose Transportfahrzeuge realisieren zu können, muss sie in eingebetteten Systemen implementiert werden. Durch das hohe Datenaufkommen, welches durch Bilder entsteht, ist die Bildverarbeitung sehr rechenaufwändig.

Da Prozessoren für eingebettete Systeme im Allgemeinen sehr energieeffizient arbeiten müssen und auch in ihrer Abwärme sehr eingeschränkt sind, ist die Rechenleistung deutlich geringer als die von PC-Prozessoren. Einfache Bildverarbeitungsaufgaben werden direkt in eingebetteten Prozessoren wie z.B. dem Intel Xscale durchgeführt. Für komplexere Bildverarbeitungsaufgaben und die Verarbeitung von höher aufgelösten Bildern werden häufig DSPs<sup>2</sup> oder FPGAs eingesetzt.

DSPs sind vorwiegend für die Verarbeitung von eindimensionalen Signalen ausgelegt. Aktuelle Versionen sind leistungsstark genug um Bildverarbeitungsaufgaben wie Bildfilterung, Berechnung der FFT<sup>3</sup> eines Bildes oder aber verschiedene Kompressions- und Kodierverfahren durchführen zu können. So arbeiten in einer Reihe von "Hardware-Bildverarbeitungsschaltkreisen" wie z.B. MPEG<sup>4</sup>-Encoder und Decoder DSPs mit einem festen Programm.

In vielen kommerziellen Produkten werden FPGAs als Steuersystem oder als Unterstützung für einen Prozessor bei rechenaufwändigen Aufgaben eingesetzt. Der Entwurf eines Designs für ein FPGA-Design ist jedoch mit einem größeren Aufwand verbunden als der eines DSPs oder Mikroprozessors.

FPGAs sind aufgrund ihrer Möglichkeiten zur starken Parallelisierung sehr gut für die Bildverarbeitungsaufgaben geeignet. Ein Beispiel hierfür ist zum Beispiel die Korrektur der Linsenverzerrung von Rückprojektionsfernsehgeräten mittels einer ortsvarianten Filterung [19].

In Industriekameras werden üblicherweise kleinere bis mittlere FPGAs eingesetzt, um die Nachverarbeitung der Bilder durchzuführen. Diese können unter anderem Algorithmen enthalten, wie sie in [19] zur Korrektur der Linsenverzerrung einer Projektion verwendet werden, für die Korrektur der Verzerrung einer Linse realisiert werden. Weiterhin werden in einigen Framegrabbern durch den Anwender programmierbare FPGAs eingesetzt. Dies ermöglicht eine anwendungs-

---

<sup>2</sup>Digital Signal Prozessor beschrieben in Abschnitt 3.2.2

<sup>3</sup>Fast Fourier Transformation - erzeugt eine Frequenzdarstellung eines Signales oder Bildes

<sup>4</sup>Moving Pictures Expert Group - ein Standard zur Videokompression

spezifische Vorverarbeitung während der Bildaufnahme. Der Einsatz dieser FPGAs wird häufig durch die vom Hersteller mitgelieferten Bibliotheken zur einfacheren Programmierung unterstützt.

Verschiedene kommerzielle Lösungen für die Auswertung von Stereobildern werden durch eine Kombination von FPGA, DSP oder Mikroprozessor realisiert. Das System Bumblebee 2 von Point-Grey-Research berechnet rektifizierte Kamerabilder mit Hilfe eines FPGAs. Die Firma TYXZ erstellt in ihrem Stereoauswertesystem Deep Sea rektifizierte und Census-Transformierte Bilder für die weitere Verarbeitung.

Die Firma Bosch entwickelt ein Steuergerät für die Stereobildauswertung und Clusterung von 3D-Punkten, welches voraussichtlich ebenfalls auf FPGA-Technologie basiert. Der genaue Aufbau ist nicht bekannt.

Im Abschnitt 2.4.5 werden die oben genannten echtzeitfähige Lösungen zur Stereobildverarbeitung detaillierter diskutiert.

### 1.3 Eingrenzung der Aufgabenstellung

Diese Arbeit untersucht die Möglichkeiten eines Umfelderkassungssystems auf Basis einer Stereobildauswertung für ein Spurwechselassistenzsystem und die Hinderniserkennung für fahrerlose Transportfahrzeuge. Für beide Anwendungsszenarien werden jeweils die Randbedingungen, Messbereich, Genauigkeit und Echtzeitforderung abgeleitet. Eine Implementierung einer echtzeitfähigen Auswertung von Stereobildern auf einem eingebetteten System für Spurwechselassistenzsysteme und fahrerlose Transportfahrzeuge ist das Ziel dieser Arbeit.

Beide Szenarien unterscheiden sich deutlich hinsichtlich des Messbereiches und der Echtzeitbedingungen. Die Erstellung und Auswertung der Tiefenkarte aus einem Stereobildpaar, aufgenommen im Normalfall der Stereophotogrammetrie, erfolgt in beiden Fällen unter sehr ähnlichen Randbedingungen.

Für diese Arbeit ist vorwiegend die Positions- und Größenbestimmung der Objekte relevant. Für fahrerlose Transportfahrzeuge wird eine Hinderniserkennung entwickelt. Die Klassifizierung und Bestimmung von Objekten wird in dieser Arbeit nicht näher betrachtet, da dieser Aspekt für die Funktionen beider Anwendungsfälle nicht relevant ist.

Zur Realisierung eines optischen Sensorsystems zur Umfelderkassung werden Algorithmen und Verfahren zur Stereobildauswertung bezüglich ihrer Bildverarbeitungseigenschaften und ihres Rechenaufwandes bewertet und auf Genauigkeit und Robustheit getestet. Im Mittelpunkt stehen hierbei blockbasierte Korrelationsverfahren. Die Genauigkeit der Messung und der Rechenaufwand hängen sowohl von der Blockgröße als auch von der räumlichen Auflösung der Tiefenkarte ab.

Anhand der Echtzeitforderung und des Rechenaufwandes der verschiedenen Kriterien wird die Implementierungsplattform für die Realisierung diskutiert. Dazu werden die Algorithmen hin-



sichtlich ihrer Leistungsfähigkeit in eingebetteten Systemen auf DSP- und FPGA-Basis untersucht.

Anhand von Untersuchungen mit synthetischen und realen Bildern wurde ein hierarchisches und ein lineares blockbasiertes Verfahren zur Stereobildauswertung entwickelt und auf einer FPGA-basierten Plattform echtzeitfähig umgesetzt. Dieses System wurde anschließend zu einem System zur Hindernisdetektion für fahrerlose Transportfahrzeuge weiterentwickelt.

## 1.4 Aufbau der Arbeit

Die Arbeit inklusive Einleitung und Zusammenfassung aus 8 Kapiteln. Die Einleitung führt auf das Thema hin, begründet die Auswahl des Sensorsystems und grenzt die Aufgabenstellung ein.

Im Kapitel 2 werden nach den Kameratechnologien mit unterschiedlichen Spektral- und Dynamikbereichen zunächst grob Verfahren zur monokularen Gewinnung von Tiefenwerten beschrieben. Anschließend wird näher auf die Grundlagen der Mehrbildaufnahme eingegangen. Hierbei werden die innere Orientierung, die Kalibrierung und die wesentlichen Auswerteverfahren beschrieben. Abschließend wird der aktuelle Stand der Technik von 3D-Vermessungssystemen für Echtzeitanwendungen beschrieben.

Kapitel 3 konzentriert sich auf die Implementierung und die hierfür zur Verfügung stehenden Plattformen. Nachdem zunächst die für die Implementierung wesentlichen Begriffe Datenfluss und Kontrollfluss beschrieben wurden, werden die Eigenschaften und Möglichkeiten von prozessorbasierten und hardwarebasierten Plattformen diskutiert.

Anschließend erfolgt in Kapitel 4 der Entwurf eines Konzeptes für eine Stereobildauswertung in Echtzeit. Zunächst werden die konkreten Randbedingungen anhand der zwei Szenarien Spurwechselassistenzsystem und fahrerlosen Transportfahrzeuge ermittelt. Anhand der Randbedingungen werden Algorithmen und Plattformen einander gegenübergestellt und diskutiert. Basierend auf den Untersuchungen dieses Kapitels wird ein Konzept bezüglich einer datenflussorientierten Stereobildauswertung auf einer FPGA-Plattform vorgeschlagen.

Im Kapitel 5 werden zwei Lösungsvorschläge vorgestellt. Es wird einerseits ein für eine Hardwareimplementierung gut geeignetes hierarchisches Verfahren beschrieben, welches während der Bearbeitung des Themas entwickelt und patentiert wurde. Zum anderen wird ein lineares Korrelationsverfahren durch Mehrfachimplementierung der Korrelation in Form einer Kaskade an die Implementierung in Hardware angepasst. Diese beiden Ansätze werden bezüglich ihrer Implementierung, der erreichten Geschwindigkeit sowie ihrer Erweiterungsmöglichkeiten verglichen.

Kapitel 6 enthält den messtechnischen Vergleich der beiden in Kapitel 5 implementierten Verfahren für verschiedene Szenarien. Die experimentelle Überprüfung der beiden Verfahren wurden sowohl im Bereich von 1 - 20 m mit zwei unterschiedlichen Stereokamerasystemen mit 5,5 cm und 12 cm Basisbreite und in einem Bereich von 10 - 120 m mit einem Stereokamerasystem

## KAPITEL 1. EINLEITUNG

von 70 cm Basisbreite Messungen durchgeführt. Es wurden in beiden Fällen synthetische und reale Objekte für die Messungen verwendet.

Im Kapitel 7 wird, ausgehend von den Ergebnissen der echtzeitfähigen Implementierung eine Hinderniserkennung für fahrerlose Transportfahrzeuge entwickelt. Die zur Realisierung verwendete Hardwareplattform wird näher vorgestellt. Hierbei wird auch die Anbindung der Kameras und die Kommunikation mit dem Fahrzeug betrachtet. Abschließend werden die Ergebnisse einer Testfahrt des Hindernissensors auf einem fahrerlosen Fahrzeug bei der Götting KG vorgestellt und diskutiert.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick.

Im Anhang sind ergänzende Informationen zur Thematik der Arbeit zusammengestellt.

# Kapitel 2

## Umfelderfassung mittels Stereophotogrammetrie

Die Wahrnehmung von Tiefeninformationen aus einem Stereobild wird im menschlichen Gehirn während des Kleinkindalters erlernt und dann kontinuierlich in “Echtzeit” ausgeführt. Um ähnliche Vorgänge mit technischen Systemen zu realisieren, ist ein großer technischer und algorithmischer Aufwand erforderlich. Das folgende Kapitel beschreibt die wesentlichen Grundlagen der Aufnahme und Auswertung von Stereobildern mit den Methoden der technischen Bildverarbeitung. Hierbei wird kurz auf die zur Verfügung stehenden Kameratechnologien und die Optik der Stereophotogrammetrie eingegangen. Im weiteren Verlauf werden dann die Auswertungs- und Verarbeitungsmöglichkeiten digitaler Systeme detaillierter erläutert.

### 2.1 Optoelektronische Bildaufnahme

In diesem Abschnitt wird kurz auf die technischen Möglichkeiten zur Bildaufnahme eingegangen, da für eine Umfelderfassung im Außenbereich die Wahl des optischen Sensors entscheidend ist.

Die wesentlichen Sensortechnologien für eine Bildaufnahme sind CCD-Sensoren und Active-Pixel-Sensoren. Das Funktionsprinzip von beiden beruht auf dem inneren photoelektrischen Effekt, der bei halbleitenden Materialien beobachtet wird. Photonen, die ausreichend Energie besitzen, setzen Elektronen in dem halbleitenden Material frei. Die gespeicherte resultierende Ladung dient als Maß für die Helligkeit. Verschiedene Halbleitermaterialien unterscheiden sich in der zur Freisetzung erforderlichen Energie und somit auch im Bereich des aufnehmbaren Spektrums. Für die Umfelderfassung sind vor allem das sichtbare Spektrum und das Infrarot-Spektrum interessant.

Bei High-Dynamic-Range-Aufnahmen handelt es sich um indirekte Aufnahmen, bei denen mehrere Bilder unterschiedlicher Belichtung zusammengefasst werden, um einen sehr hohen Dynamikbereich zu erzeugen.

**CCD-Kameras** Die Grundlagen der Charged-Coupled-Devices (CCD) wurden Ende der 1960er Jahre in den Bell Laboratories bei AT&T gelegt. 1969 erfanden Willard Bolye und George E. Smith eine Technologie zur Bildaufnahme, bei der, die durch den inneren photoelektrischen Effekt ([13]) erzeugte Ladung in einer MOS-Kapazität gespeichert wird und nach der Bildaufnahme zu einem für jede Bildzeile vorhandenen Analogsignalverstärker geschoben wird. Hierbei wird die Ladung von der Kapazität eines Sensorelementes zum nächsten geschoben, wodurch der Name geprägt wurde ([55] S. 161).

Wesentlicher Vorteil der CCD-Technologie ist die Maximierung der zur Bildaufnahme verfügbaren Fläche durch den Verzicht auf aktive Sensorelemente. Daher wird CCD auch als passives Sensorprinzip bezeichnet. Dies war insbesondere zu Beginn der optoelektronischen Bildaufnahme ein wichtiger Schritt, da die Strukturen auf integrierten Schaltkreisen im Vergleich zum gegenwärtigen Stand der Technik verhältnismäßig groß waren und mittels dieses Sensorprinzips Bilder mit guter Lichtausbeute bei akzeptablem Preis geliefert werden konnten.

Problematisch bei CCD-Sensoren ist die Überbelichtung. Selbst kleine überbelichtete Bereiche können durch den Kapazitätenverbund zum so genannten Blooming-Effekt führen. Dieser Effekt ist insbesondere für einen Einsatz außerhalb von geregelten Beleuchtungsbedingungen problematisch z.B. im Außenbereich. Scheinwerfer in der Nacht und die Sonne am Tage können sehr schnell zu einer Überbelichtung führen.

**Active-Pixel-Sensor (CMOS-Sensor)** Im Gegensatz zu den CCD-Sensoren werden bei den Active-Pixel-Sensoren (APS) die Ladungen direkt verstärkt und weitergeleitet. Dabei hat jedes Sensorelement einen eigenen Analogsignalverstärker. Dieses Sensorprinzip wurde zuerst von Noble [68] im Jahr 1968 beschrieben. Durch den erheblichen Aufwand an Elektronik reduzierte sich die optisch nutzbare Fläche des Sensors zu Beginn auf 30%. Dies führte zu einem sehr geringen Wirkungsgrad.

Durch die Miniaturisierung der Strukturen in der Halbleitertechnologie, insbesondere mit der Einführung der CMOS<sup>1</sup>-Technologie, konnte die aktive Fläche der Sensoren deutlich vergrößert werden. Zusätzlich werden Mikrolinsen eingesetzt, die ein Maximum des verfügbaren Lichtes auf die aktive Sensorfläche lenken. Da dieses Sensorprinzip erst durch den Einsatz der CMOS-Technologie sinnvoll genutzt werden konnte, werden diese Sensoren häufig auch als CMOS-Sensoren bezeichnet. Der Begriff Active Pixel Sensor wurde vor allem von Eric Fossum [27] geprägt.

Der Vorteil dieses Sensorprinzips ist eine sehr robuste Bildaufnahme auch in optisch problematischen Situationen. Durch den Einsatz der energieeffizienten CMOS-Technologie haben die Sensoren trotz ihres höheren Elektronik-Anteils einen deutlich geringeren Leistungsbedarf als vergleichbare CCD-Sensoren, die auf MOS-Technologie basieren. Weiterhin können sie direkt auf CMOS-Fertigungsstraßen für die Massenproduktion hergestellt werden.

---

<sup>1</sup>Complementary Metal Oxide Semiconductor

**High-Dynamic-Range-Imaging** Beim High-Dynamic-Range-Imaging handelt es sich nicht wie bei CCD und APS um ein elektronisches Sensorprinzip. Vielmehr steht hinter diesem Prinzip eine Kombination mehrerer Aufnahmen mit einer der oben genannten Sensortechnologien. Die zur Kombination verwendeten Aufnahmen unterscheiden sich in ihrer Belichtung. Durch die Kombination der jeweils nicht von den anderen Bildern erfassten Dynamikbereiche erhält man so ein Bild mit deutlich höherem Dynamikumfang.

Das Verfahren wurde 1997 von Paul Debevec [20] entwickelt, um Aufnahmen mit einem hohen Dynamikumfang zu erzeugen. Hierbei werden mit herkömmlicher Fotografie verschiedene Bilder mit unterschiedlicher Belichtungszeit aufgenommen. Die Bilder werden digitalisiert und zur Ermittlung der Antwortfunktion des fotografischen Prozesses verwendet. Anschließend werden die Bilder anhand der gewonnenen Informationen zusammengefügt. Ziel ist es, Aufnahmen zu erhalten, die den großen Dynamikbereich des menschlichen Auges haben.

In den letzten 10 Jahren konnte durch die Beschleunigung der Bildaufnahme die Implementierung dieses Verfahrens in spezielle APS-Kameras erreicht werden. Hierbei werden 3 Bilder in Folge mit unterschiedlicher Belichtung mit sehr hoher Bildfrequenz aufgenommen. Anschließend werden die Bilder online kombiniert [34]. Hierbei werden zum Beispiel Sequenzen mit 180 Bildern/s aufgenommen und daraus HDR-Bilder mit einer resultierenden Bildfrequenz von 60 Hz erzeugt. Die Bilder stehen in einer Auflösung von 16 Bit Graustufen bzw. in 16 Bit je Farbe zur Verfügung.

Die in ihrem Dynamikumfang begrenzten Ausgabemedien machen eine Überarbeitung der Bilder und eine Kompression des Dynamikbereiches für eine Darstellung erforderlich. Für die automatische Auswertung besteht kein prinzipielles technisches Problem bezüglich der höheren Auflösung des Dynamikbereiches. Die Verarbeitung in Echtzeit könnte durch den höheren rechnerischen Aufwand erschwert werden. Es können weiterhin Probleme bei einer stereophotogrammetrischen Auswertung der Bilder entstehen, da die Aufnahmen durch einen adaptiven Algorithmus erzeugt werden. Somit könnte die Ähnlichkeit der Objekte in beiden Bildern gestört sein. In dieser Arbeit wurden keine Untersuchungen zur Verwendung von HDR-Bildern zur Stereophotogrammetrie vorgenommen.

**Bedeutung verschiedener elektromagnetischer Spektren** Für die Umfelderfassung sind unterschiedliche Bereiche des elektromagnetischen Spektrums relevant. Der Bereich des sichtbaren Lichtes ( $\lambda = 380 - 780nm$ ) ermöglicht es einem Messsystem, theoretisch die gleichen Informationen zu erhalten, die auch ein Mensch wahrnehmen kann. Dies ist insbesondere zur Orientierung, Navigation und auch der Hinderniserkennung wichtig. Im Infrarot-Bereich werden Hindernisse und Objekte anhand ihrer Wärmesignatur dargestellt. Hierbei muss zwischen dem nahen (NIR;  $\lambda = 780 - 1400nm$ ), dem mittleren (MIR;  $\lambda = 3 - 50\mu m$ ) und dem fernen (FIR;  $\lambda = 50\mu m - 1mm$ ) Infrarotbereich unterschieden werden.

Während der nahe Infrarotbereich am Tag eine unseren Sehgewohnheiten sehr ähnliche Darstellung bietet und nachts Objekte anhand ihrer Wärmesignatur deutlich besser wahrnehmbar als im Spektrum des sichtbaren Lichtes darstellt, werden im mittleren und im fernen Infrarot-

Bereich die Objekte immer gleich angezeigt und die Darstellung geringer von der Sonneneinstrahlung beeinflusst. Sie weisen eine gewisse Unempfindlichkeit bezüglich Nebel und ähnlichen Sichtbehinderungen auf und können so auch am Tag bei widrigen Witterungsbedingungen eine Zusatzinformation liefern. Aufgrund der Beeinflussung durch die Sonneneinstrahlung erfolgt die Auswertung im Allgemeinen nicht über die Absolutwerte, sondern vorwiegend durch die Ermittlung des Grauwert-Gradienten.

Das ultraviolette Licht ( $\lambda = 1nm - 380nm$ ) spielt in der Umfelderfassung gegenwärtig keine Rolle. Dieses hochenergetische Licht benötigt Spezialkameras und Objektive, da die UV-Strahlung durch Glas gedämpft wird und die herkömmlichen Sensoren nicht ausreichend empfindlich sind.

## 2.2 Methoden zur Bestimmung von 3D-Informationen mit monokularen Aufnahmesystemen

Die Gewinnung von Tiefeninformationen aus Bildern wird seit 1851 untersucht. Da der Aufwand für eine Stereobildverarbeitung sehr hoch ist, wie die nachfolgenden Abschnitte zeigen, soll an dieser Stelle kurz auf die Möglichkeiten der monokularen Bildauswertung eingegangen werden.

Eine allgemeine Ermittlung von Tiefendaten aus einem Bild ohne a priori Informationen ist zunächst nicht möglich. Über die Beziehungen der Optik können Vermutungen über die Verhältnisse der Größe verschiedener Objekte im Bild angenommen werden. Ist die Größe oder die Entfernung eines Objektes bekannt, kann die jeweils andere Größe mittels einfacher Triangulation gewonnen werden. Weiterhin können auch die Größe und Entfernung anderer Objekte geschätzt werden. In der Fahrzeugtechnik kann z.B. die Größe des Nummernschildes verwendet werden, da diese in den meisten Fällen (bis auf Sondergrößen) nur in engen Grenzen variiert [62].

**Shape from Shading** Durch die Auswertung weiterer Informationen aus dem Bild kann eine deutlich präzisere und zuverlässigere Ermittlung der Tiefendaten erreicht werden. Eine Variante sind “Shape from Shading”-Verfahren [14]. Hierbei wird durch Beobachtung von Helligkeit, Reflexionspunkten und Schatten einer bekannten Lichtquelle über eine Bildsequenz eine deutlich feinere Oberflächenbestimmung möglich. Die Bestimmung der absoluten Abmaße ist nicht möglich [98]. Eine Bestimmung von Tiefeninformationen außerhalb von Gebäuden ist nur bedingt möglich.

**Shape from Focus/Defocus** Die Veränderung der Aufnahmeparameter durch das Verändern der Fokussierung der Kamera wird als “Shape from Focus/Defocus” bezeichnet [66]. Es entstehen durch die unterschiedliche Fokussierung in jeweils unterschiedlichen Entfernungsbereichen scharfe Kanten. Durch diese Auswertung kann eine vollständige Tiefenkarte gewonnen werden. Dieses Verfahren ist jedoch sehr zeitaufwändig und für bewegte Szenen nicht geeignet.

**Optischer Fluss** Ein weiteres Verfahren ist die Bestimmung der relativen Tiefeninformationen auf Basis des optischen Flusses. Hierbei wird in einer Bildfolge die Bewegung jedes Objektpunktes aus dem Versatz zwischen den Bildern bestimmt [65]. Aus dieser Bewegungsinformation erfolgt dann eine Annahme über die Position der Pixel relativ zur Kamera. Dieses Verfahren ist vorwiegend zur Bestimmung der Entfernung bei statischer Kamera und bewegtem Objekt oder statischem Objekt und bewegter Kamera geeignet.

**3D-Vermessung durch Zusatzinformationen** Durch Hinzufügen von Informationen zum Bild, z.B. dem Überlagern mit einem sinusförmigen Streifenmuster [63], können durch Auswertung der Frequenz- bzw. Phasenveränderungen auf einem Objekt die Ausmaße und die Position von Objekten im Kamerabild relativ zueinander ermittelt werden. Da die Verzerrung und Vergrößerung der Streifen sowohl durch die Entfernung als auch durch die Form des Objektes bestimmt wird, muss die absolute Entfernung durch zusätzliche Informationen in mindestens einem Punkt ermittelt werden. Die Verwendung von aktiven Sensorsystemen wurde aufgrund der bereits in der Einleitung genannten Probleme für diese Arbeit ausgeschlossen.

Die Verfahren "Shape from Shading" und 3D-Vermessung durch Überlagerung von Zusatzinformationen sind aktive Messverfahren und können somit von vornherein ausgeschlossen werden.

Die in diesem Abschnitt kurz vorgestellten Verfahren zur Gewinnung der Tiefeninformation mittels monokularer Aufnahmesysteme sind für die in dieser Arbeit behandelte Aufgabenstellung nicht oder nur bedingt geeignet, da mit ihnen in unbekanntem Umgebungen im Außenbereich keine exakten Messungen durchgeführt werden können. Sie stellen jedoch in ihren jeweiligen Anwendungsbereichen gute Lösungen dar, vor allem bei Berücksichtigung des technischen und algorithmischen Aufwandes. Im Folgenden werden nun die wesentlichen Grundlagen zur Stereobildaufnahme und ihrer Verarbeitung beschrieben.

## 2.3 Grundlagen der Stereobildauswertung

Stereokamerasysteme stellen die minimale Konfiguration eines Mehrkamerasystems dar. Die Aufnahme und Auswertung aus Mehrkamerakonfigurationen wird unter dem Begriff Photogrammetrie zusammengefasst. Durch die Verwendung von mehreren Kameras bei der Bildaufnahme werden mehr unabhängige Daten gewonnen. Durch die daraus gewonnenen zusätzlichen Informationen können konkrete 3D-Informationen für alle Punkte, die durch mindestens 2 Kameras abgebildet werden, bestimmt werden. Je nach Anwendung können durch die Verwendung von mehreren Kameras z.B. Verdeckungsprobleme aufgelöst oder Räume vollständig vermessen werden. Erforderlich ist hierfür insbesondere die Kenntnis der Position und der Winkel der Aufnahmen zueinander. Diese Informationen werden im Kalibrierprozess gewonnen.

Entwickelt wurde die Photogrammetrie kurz nach der Erfindung der Fotografie. Sie wurde durch den französischen Offizier Laussedat(1851) und den deutschen Architekten Meydenbauer (1858) unabhängig von einander entwickelt und zunächst vorwiegend zur Vermessung von

Gebäuden und zur Landvermessung eingesetzt. Im Laufe der Zeit wurden verschiedene Apparaturen zur effektiveren Bildaufnahme und Auswertung entwickelt. Die Auswertung erfolgte im folgenden Jahrhundert vorwiegend manuell unter Zuhilfenahme spezieller Geräte, zum Beispiel des Stereokomparators ([55] S.20). Nähere Informationen zu der geschichtlichen Entwicklung der Stereophotogrammetrie werden zum Beispiel bei Luhmann ([55] S. 18ff) und Kraus ([46] S. 2ff) ausgeführt.

Mit der kontinuierlichen Entwicklung optoelektronischer Kameras gewann die digitale Bildaufnahme und -auswertung auch in anderen Anwendungsbereichen stetig an Bedeutung. Zum Beispiel wurden beim EUREKA Prometheus Projekt [94] Stereokameras auch für die Umfelderfassung verwendet. Die Auswertung erfolgte auf Spezialrechnern im Laderaum des Fahrzeugs. Im nächsten Abschnitt werden die Möglichkeiten zur digitalen Auswertung von Stereobildpaaren dargelegt.

Im Anhang A wird die Photogrammetrie ausführlicher beschrieben. Weiterhin wird auf die Standardwerke Kraus [46], Luhmann [55] und Schenk [79] verwiesen. Im Folgenden werden die Grundlagen des Normalfalles der Stereophotogrammetrie kurz zusammengefasst.

### 2.3.1 Der Normalfall der Stereophotogrammetrie

Durch geeignete Ausrichtung der Kameras zueinander kann der allgemeine Fall der Stereophotogrammetrie zum Spezialfall des Normalfalles vereinfacht werden.

Dabei werden zwei möglichst identische Kameras so ausgerichtet, dass die Projektionsebene beider Kameras in der gleichen Ebene liegt und alle Drehwinkel zwischen den beiden Kamerakoordinatensystemen Null sind. In Abbildung 2.1 ist ein Schnitt durch die XZ-Ebene eines Kamerasystems im Stereonormalfall abgebildet. Hierbei stellen  $X, Y, Z$  das Weltkoordinatensystem und  $x, y$  das Bildkoordinatensystem dar. Die Projektionszentren  $O'(X_{01}, Y_{01}, Z_{01})$  und  $O''(X_{02}, Y_{02}, Z_{02})$  beider Kameras liegen in der XY-Ebene, während die Projektionsebene eine in Z-Richtung verschobene parallele Ebenen zur XY-Ebene ist. Damit wird die Basisbreite  $b$  durch  $b = X_{02} - X_{01}$ , dem einzigen Versatz der beiden Kameras zueinander, beschrieben. Der Ursprung des Weltkoordinatensystems liegt im Projektionszentrum  $O'$  der ersten Kamera somit können die 3D-Koordinaten des Objektpunktes durch Zusammenhänge in den Gleichungen 2.1, 2.2, 2.3 beschrieben werden.

Durch diese Voraussetzungen reduziert sich die Suche auf den Versatz in x-Richtung. Wird der Versatz in x-Richtung als Disparität  $\Delta u = x' - x''$  definiert, ergeben sich folgende Zusammenhänge für die Koordinaten des Punktes P:

$$X = \frac{b \cdot x'}{\Delta u} \tag{2.1}$$

$$Y = \frac{b \cdot y'}{\Delta u} \tag{2.2}$$

$$Z = \frac{b \cdot c}{\Delta u} \tag{2.3}$$



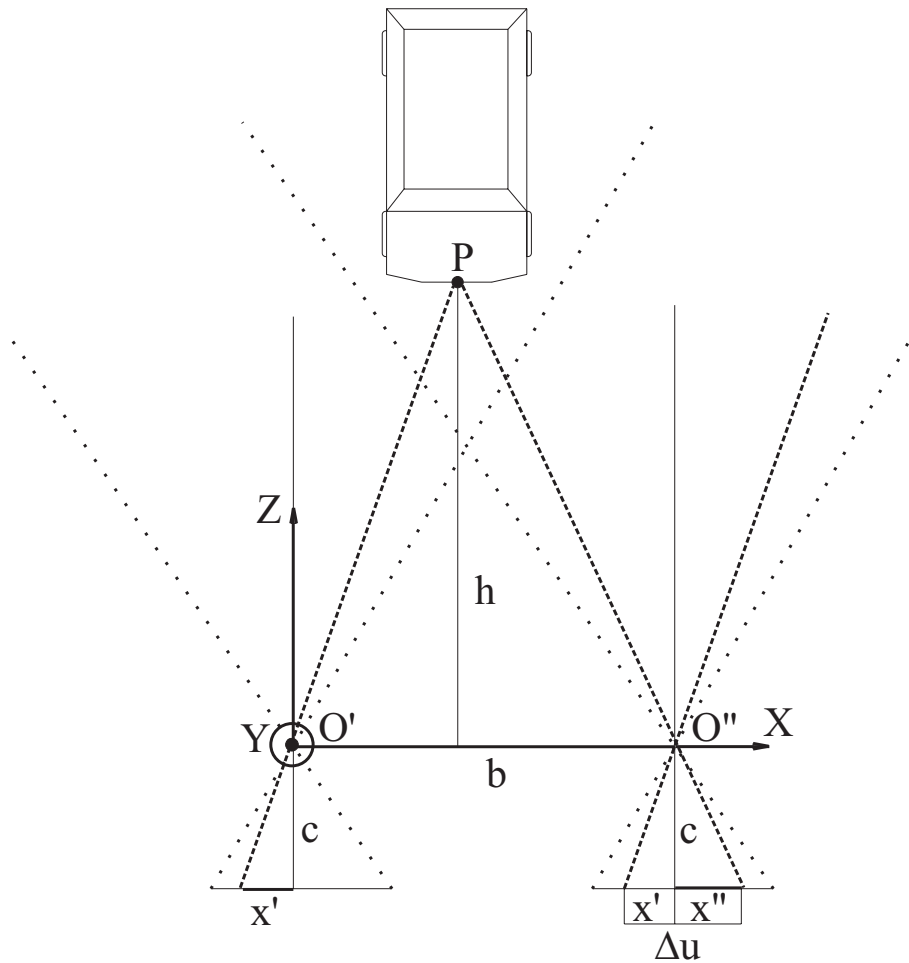


Abbildung 2.1: Ermittlung der 3D-Position im Normalfall der Stereophotogrammetrie

Dieser Zusammenhang ergibt sich für die obigen Annahmen auch aus den Kollinearitätsgleichungen [55].

Durch diese Anordnung wird die Auswertung wesentlich vereinfacht. Der Aufwand für die Justage der Kameras steigt dadurch jedoch wesentlich. Es müssen alle Drehwinkel beider Kameras und der Versatz in  $X$ - und in  $Z$ -Richtung frei variiert werden können. Während der laufenden Messung hätte eine Veränderung dieser Parameter einen Messfehler zur Folge. Für eingebettete Sensorsysteme kommt bei Voraussetzung des Normalfalles der Stereophotogrammetrie somit nur eine sehr exakte Fertigung und Fixierung der Kameras in einem festen Rahmen in Frage.

### 2.3.2 Rektifizierung

Die Rektifizierung [11] beschreibt ein Verfahren, um Bilder, die im allgemeinen Fall der Stereophotogrammetrie aufgenommen wurden, mit dem Normalfall der Stereophotogrammetrie zu verarbeiten. Dazu wird mindestens eines der beiden Bilder so verzerrt, dass der Normalfall erfüllt ist.

Zur Rektifizierung werden zwei Bildebenen ( $x'_n, y'_n$  und  $x''_n, y''_n$ ) angenommen, deren Projektionszentren in denen der wahren Kameras  $O'$  und  $O''$  liegen, die jedoch den Normalfall der Stereophotogrammetrie erfüllen, wie in Abbildung 2.2 dargestellt. Hierbei ist  $el$  die Epipolarlinie der rektifizierten Bilder und  $el', el''$  sind die jeweiligen Epipolarlinien im allgemeinen Fall der Stereophotogrammetrie. Die Grauwerte werden durch eine Koordinatentransformation in die Bildkoordinaten der Normalbilder umgerechnet.

Die Transformation wird auf Basis der Kalibrierung durchgeführt. Für Systeme, die sich dem Normalfall der Stereophotogrammetrie sehr gut annähern, kann eine Rektifizierung hardwareunterstützt in Echtzeit realisiert werden [74].

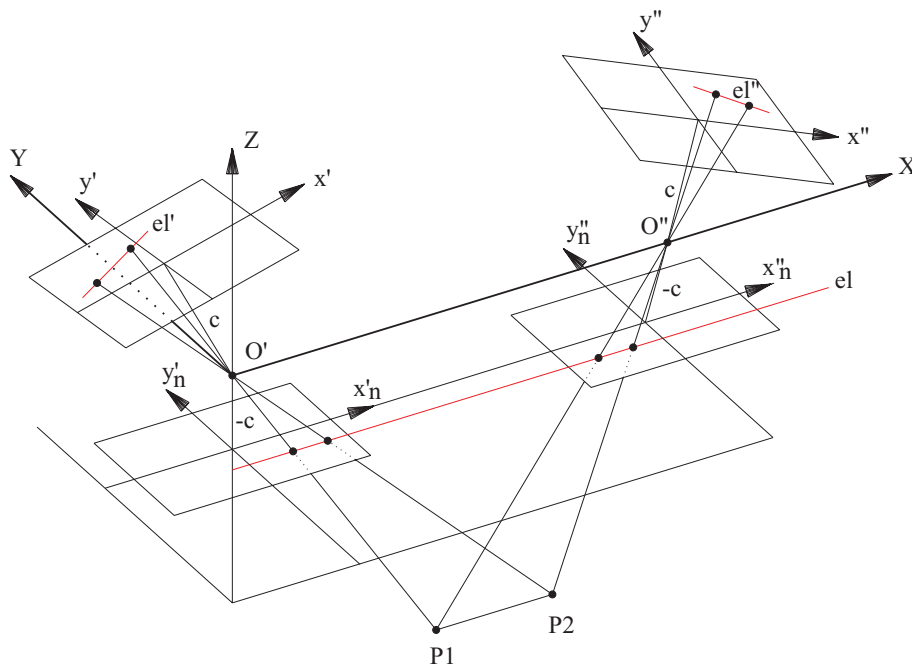


Abbildung 2.2: Rektifizierung eines Stereobildpaares

### 2.3.3 Epipolargeometrie

Die Epipolargeometrie [16] beschreibt die Lage der homologen Punkte durch eine Ebene durch beide Projektionszentren und einen Bildpunkt bzw. den Objektpunkt. Dadurch ergibt sich für die zweite Kamera ein Bereich im Bild, in dem sich der korrespondierende Punkt befinden wird. Bei Verwendung des Normalfalles der Stereophotogrammetrie verlaufen aufgrund der speziellen Annahmen über die Koordinatensysteme (siehe Abschnitt 2.3.1) die Epipolarlinien in x-Richtung, wie in Abbildung 2.3 dargestellt. Dies vereinfacht die Suche nach den korrespondierenden Punkten. Im Normalfall kann die Suche auf die Richtung der Zeilen begrenzt werden.

Durch die Vereinfachungen bei der Ermittlung der 3D-Koordinaten und der Suche nach den korrespondierenden Punkten eignet sich der Stereonormalfall sehr gut für die Auswertung mit

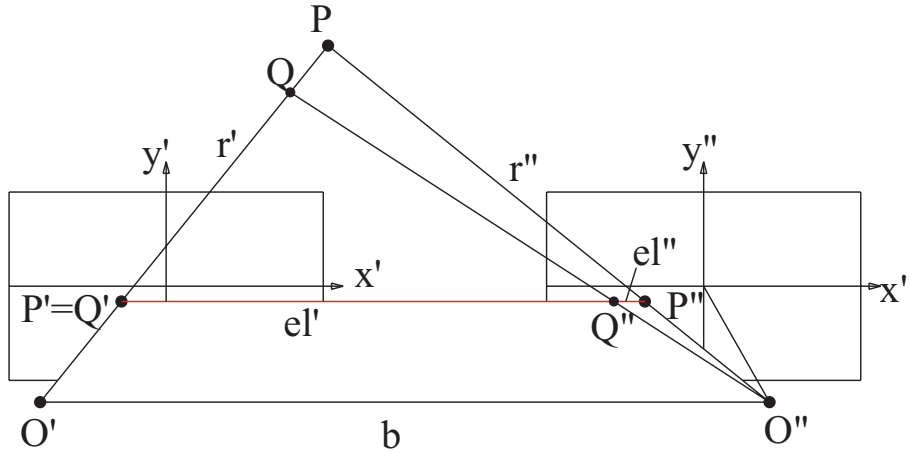


Abbildung 2.3: Epipolargeometrie im Normalfall der Stereophotogrammetrie

eingebetteten Systemen und wird daher auch in dieser Arbeit angewendet. Hierzu werden die Kameras so justiert, dass sie im Normalfall der Stereophotogrammetrie angeordnet sind.

### 2.3.4 Fehlerbetrachtung von stereophotogrammetrischen Systemen

Die Genauigkeit eines Stereokamerasystems ist vor allem von der Geometrie der Aufnahmekonfiguration und der erreichbaren Bildmessgenauigkeit abhängig [25]. Oft erhält man eine sehr gute Abschätzung, indem man die Gesetze der Fehlerfortpflanzung auf die Formeln des Normalfalles der Stereophotogrammetrie (Gleichung 2.1ff) anwendet. Der mittlere Fehler  $\sigma_z$  in der meist dominanten Z-Richtung ist dann

$$\sigma_z = \frac{Z^2}{c \cdot b} \cdot \sigma_u \quad (2.4)$$

Die Messung wird umso genauer, je geringer der Objektabstand, je größer die Kamerakonstante, je breiter der Basisabstand der Kameras ist und je genauer die Bildkoordinaten bestimmt werden können [56].

Durch die Auflösung der Kameras ergibt sich eine Quantelung der messbaren Entfernung. Mit Hilfe einer Subpixelinterpolation kann eine weitere Steigerung der Auflösung der Disparität erreicht werden. Die Präzision der Messung wird jedoch auch durch optische Einflüsse der Luft bestimmt, wodurch dieser Effekt bei großen Entfernungen relativiert wird. Der Öffnungswinkel der Kameras legt den Sichtbereich des Systems fest. Bei großen Öffnungswinkeln entstehen jedoch auch meist unerwünschte Verzerrungen, die durch nichtlineare Korrekturterme siehe (Gl. A.3 und Gl. A.4) korrigiert werden müssen. Dies ist bei Präzisionsmessungen [83] grundsätzlich erforderlich.

$$\Delta u_r = \Delta u + \frac{\frac{1}{2}(C_{(1)} - C_{(-1)})}{2 \cdot C_{(0)} - C_{(1)} - C_{(-1)}} \quad (2.5)$$

Die Genauigkeit der Bestimmung der Disparität kann sinnvoll auf  $1/10$  Pixel ausgeweitet werden. Hierzu werden die Korrelationswerte vor und nach dem Extremum gespeichert und mit Hilfe der Gleichung 2.5 eine Bestimmung der Position des Maximums der so entstehenden quadratischen Funktion durchgeführt. Der Extremwert  $C_{(0)}$ , der vorherige  $C_{(-1)}$  sowie der nachfolgende Wert  $C_{(1)}$  des Korrelationskriteriums werden zur Berechnung der subpixelinterpolierten Disparität  $\Delta u_r$  zur Disparität  $\Delta u$  addiert.

## 2.4 Korrespondenzproblem

Das Korrespondenzproblem beschreibt die Suche nach dem nächstmöglichen Äquivalent eines Objektes oder eines Bereiches eines Bildes in einem anderen Bild. Da die Abbildung im zweiten Bild durch Bewegung des Objektes oder Variation des Blickwinkels vom ersten differieren kann, muss ein gewisser Fehler bei der Bestimmung des äquivalenten Bereiches im zweiten Bild berücksichtigt werden. Dies bedeutet, dass die Verfahren fehlertolerant sein müssen. Dies ist jedoch insbesondere in Situationen mit vielen ähnlichen Abbildungen problematisch.

Es gibt viele unterschiedliche Verfahren, um das Korrespondenzproblem zu lösen, d.h. in zwei Bildern dasselbe Objekt wiederzufinden. Technische Lösungen für das Korrespondenzproblem wurden zu Beginn vor allem für eindimensionale Signale, wie sie in Audio- oder Funksignalverarbeitung vorkommen, entwickelt. Hierfür wurden auch frühzeitig schon Korrelationsverfahren eingesetzt. Mit dem Aufkommen der digitalen Bildverarbeitung wurden diese Verfahren zum Lösen des Korrespondenzproblems für Bilder erweitert und um weitere Verfahren ergänzt. Im Anschluss soll ein kurzer Überblick über die verschiedenen Verfahren gegeben werden.

Die Stereobildauswertung basiert auf der Zuordnung von homologen Punkten in zwei oder mehreren Bildern. Es müssen die Abbildungen eines Objektes aus unterschiedlichen Blickwinkeln als zusammengehörig identifiziert werden. Aus der Position der Abbildungen in den Bildern ist dann entsprechend dem räumlichen Vorwärtsschnitt im allgemeinen Fall oder über die Gleichungen 2.1ff im Normalfall der Stereophotogrammetrie die Bestimmung der 3D-Koordinaten eines Objektpunktes möglich. Das Zuordnen der Abbildungen zueinander wird im Allgemeinen als Korrespondenzproblem bezeichnet. Wird eine dichte Suche über das gesamte Bild durchgeführt, entsteht eine Tiefenkarte (siehe Abbildung 2.4).

Problematisch ist die Korrespondenzanalyse bei Objekten, die nur eine geringe bzw. keine Eigentextur besitzen (wie z.B. weiße Wände). Hier bietet sich eine vorherige Suche nach Kanten bzw. Bildbereichen mit ausreichender Textur an. Nur an diesen Stellen wird dann die Korrespondenzsuche vorgenommen. Andere Bereiche werden ausgeblendet bzw. interpoliert. Dies ist neben entstehenden Problemen auch teilweise vorteilhaft, weil so die 3D-Punktewolke von vornherein ausgedünnt wird.

Im Abschnitt 2.3.3 wird die Eingrenzung des Suchbereiches mittels der Epipolarometrie beschrieben. Dadurch kann ein wesentlicher Teil des Rechenaufwandes eingespart werden.

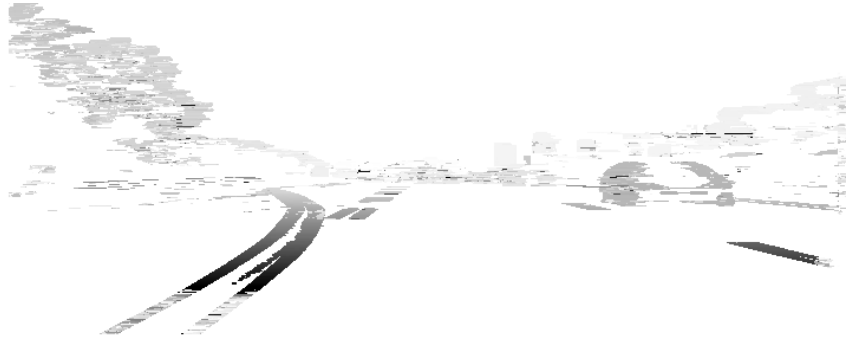


Abbildung 2.4: Beispiel einer Tiefenkarte (dunklere  $\hat{=}$  nahe Punkte; hell  $\hat{=}$  ferne Punkte)

### 2.4.1 Ermittlung des Suchbereiches

Es gibt verschiedene rechnerische Verfahren, die Zusammengehörigkeit bzw. die Ähnlichkeit zweier Bildbereiche zu identifizieren. Diese Verfahren sind, wie im Anschluss gezeigt wird, sehr rechenaufwändig. Ein Schritt zum Einschränken des Rechenaufwandes ist die Begrenzung des Suchbereiches. Hierbei hilft die Epipolargeometrie.

Ist die innere und äußere Orientierung eines Kamerasystems aus der Kalibrierung bekannt, kann durch den Objektpunkt oder durch eine der Abbildungen zusammen mit den Projektionszentren eine Ebene aufgespannt werden. Diese Ebene schneidet die zweite Bildebene in einer Geraden, welche Epipolarlinie genannt wird. Der entsprechende korrespondierende Punkt im zweiten Bild kann sich nur auf der Epipolarlinie befinden. Dadurch wird der Suchbereich von einem zweidimensionalen auf einen eindimensionalen eingeschränkt.

Im allgemeinen Fall der Stereophotogrammetrie verläuft die Epipolarlinie schräg zum Bildkoordinatensystem. Somit ist es je nach Kamerakonfiguration erforderlich, für die Suche einen großen Teil bzw. das gesamte Bild im Speicher zu halten. Bei Verwendung des Normalfalles der Stereophotogrammetrie reduziert sich der Suchbereich in idealer Weise auf eine Pixelzeile und verläuft auch in Richtung der Zeile [46]. Entspricht ein Kamerasystem einem gut angenäherten Normalfall, muss nur auf einen sehr kleinen Suchbereich zurückgegriffen werden. Jedoch können dann zur exakten Bestimmung der 3D-Koordinaten die Gleichungen 2.1ff nicht mehr verwendet werden. Es ist eine Korrektur erforderlich. Eine Variante der Korrektur [48] wird im Anhang D beschrieben.

### 2.4.2 Korrespondenzanalyse

Zum Identifizieren homologer Punkte in verschiedenen Bildern gibt es Ansätze auf unterschiedlichen Ebenen der Bildverarbeitung. Es gibt objektbasierte, texturbasierte, frequenzbasierte und flächenbasierte Ansätze. Zunächst werden Verfahren betrachtet, die auf das gesamte Bild angewendet werden.

### 2.4.2.1 Verfahren für das gesamte Bild

**Objektbasierte Verfahren** Bei den objektbasierten Ansätzen erfolgt zunächst eine Klassifizierung der Objekte in beiden Bildern. Objekte, die dabei sehr ähnlich sind, werden in ihren Bildpositionen verglichen. Dies gilt insbesondere auch für signalisierte Punkte [49]. Hierbei werden eindeutig identifizierbare Marken an speziellen Punkten angebracht. Bei der Verwendung von Passpunktfeldern sind die Passpunkte in aller Regel signalisierte Punkte.

**Interest-Operatoren** In der klassischen Photogrammetrie [12] [55] werden häufig Interest-Operatoren auf beide Bilder angewendet. Interest-Operatoren sind zum Beispiel die Autokorrelationsfunktion, der Moravec-Operator und der Förstner-Operator. Die Interest-Operatoren identifizieren Regionen mit signifikanten Mustern gegenüber ihrer Umgebung. Dabei wird ein kleiner Teil des Bildes mit seiner Umgebung verglichen. Bei der Autokorrelationsfunktion ergibt sich z.B. dann ein stark ausgeprägtes Maximum, wenn sich das Zentrum des untersuchten Bereiches sehr stark von seiner Umgebung unterscheidet.

Anschließend werden die so identifizierten Bildbereiche aufgrund ihrer Lage, der Epipolargeometrie [46] und ihrer Ähnlichkeit einander zugeordnet und so das Korrespondenzproblem gelöst. Das Verfahren ist rechenaufwändig, da zunächst die relevanten Punkte identifiziert werden müssen, indem beide Bilder sehr feingliedrig abgesucht werden.

Bei diesen beiden Verfahren wird das gesamte Bild zu Beginn der Verarbeitung und für die Suche nach dem korrespondierenden Punkt herangezogen. Im Weiteren werden verschiedene Verfahren vorgestellt, bei denen jeweils nur ein bestimmter Teil des Bildes für eine Operation ausgewählt wird.

**Neuronale Netze** Verwendet man neuronale Netze, um die Ähnlichkeit in verschiedenen Bildbereichen zu untersuchen, kann man, wie in Henkel [31] beschrieben, eine Struktur aufbauen, bei der beide Bilder als Eingangsdaten (siehe Abbildung 2.5) verwendet werden. Durch eine Verschränkung der Datenpfade ergibt sich ein Maximum bei korrespondierenden Bildbereichen. Dieses Verfahren ist gut zur Erzeugung dichter Tiefenkarten geeignet. Je nach Tiefe und Breite des Netzes berücksichtigt es größere Bereiche des Bildes und größere Tiefenbereiche.

Das Verfahren ist für große Tiefenbereiche sehr aufwändig, lässt sich jedoch gut auf das Problem skalieren und in Hardware implementieren [32].

### 2.4.2.2 Blockbasierte Verfahren

Bei den blockbasierten Verfahren [79] werden beide Bilder in Blöcke eingeteilt und die Blöcke beider Bilder zumeist mit Hilfe von Korrelationskriterien miteinander verglichen. Je nach Verfahren wird eine regelmäßige Struktur, orientiert am Bildkoordinatensystem, oder eine Struktur, orientiert an Merkmalen, im Bild erzeugt. Im Allgemeinen wird bei blockbasierten Verfahren zwischen dem Referenz- und dem Suchbild unterschieden. Dementsprechend ergeben sich auch

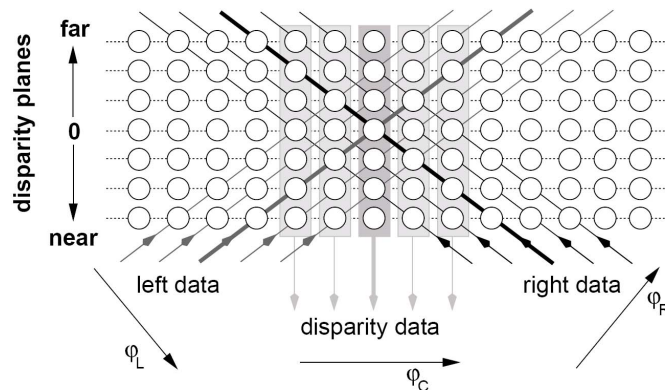


Abbildung 2.5: Korrespondenzanalyse mittels neuronaler Netze [31]

Referenz- und Suchblöcke. Der Suchbereich ist ein definierter Bereich im Suchbild, der sich durch die Anwendung der Epipolargeometrie ergibt.

**Blockbasierte Korrelation** Die blockbasierte Korrelation [12] wird auch als Blockmatching bezeichnet und ist eines der bedeutenden Verfahren zur Bewegungsschätzung. Bei der Bewegungsschätzung wird meist nur eine Kamera verwendet, von der dann mehrere aufeinanderfolgende Bilder ausgewertet werden, um die Bewegungsvektoren bestimmter Objekte zu gewinnen.

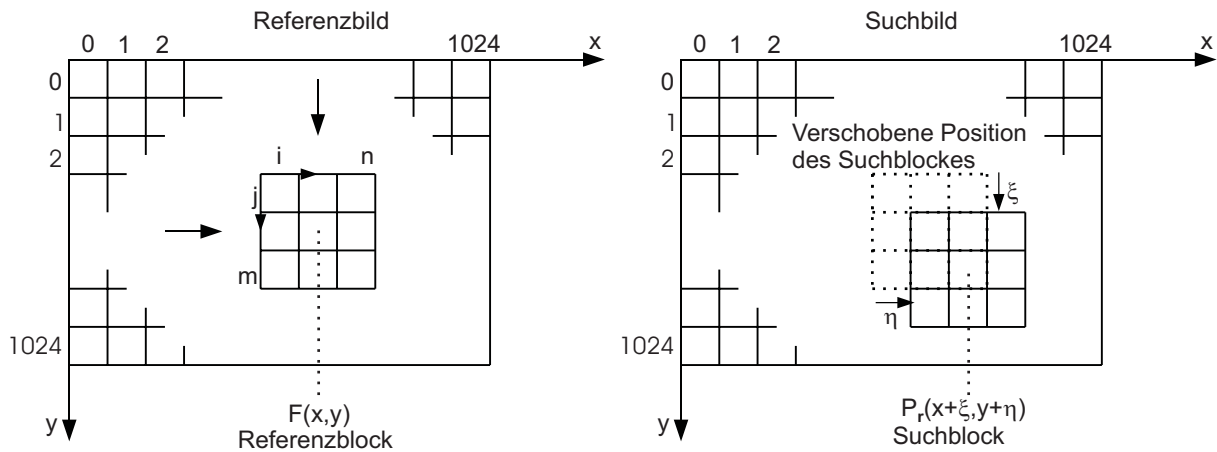


Abbildung 2.6: Organisation des Blockmatching

Die blockbasierte Korrelation ist gut für die Stereobildauswertung geeignet. Bei diesem Verfahren werden regelmäßige Blöcke direkt am Bildkoordinatensystem orientiert. Ein Block  $F(x, y)$  aus dem Referenzbild wird mit allen im Suchbereich liegenden Suchblöcken  $P_r(x + \xi, y + \eta)$  mit Hilfe eines Korrelationskriteriums (siehe Abschnitt 2.4.3) verglichen. Hierbei wird die Verschiebung der Position des Suchblockes durch  $\xi$  und  $\eta$  angegeben. Wird eine gute Übereinstimmung gefunden, so wird dies als Wert der Disparität dem Referenzblock zugeordnet. Das Verfahren ist in Abbildung 2.6 dargestellt.

**Ereignisorientierte Verfahren** Bei ereignisorientierten Verfahren werden im allgemeinen Fall Regionen mit signifikanter Information identifiziert und von ihnen ausgehend eine Suche durchgeführt. Die Verfahren auf Basis der Interest-Operatoren gehören prinzipiell zu den ereignisorientierten Verfahren. Bei den blockbasierten ereignisorientierten Verfahren wird mittels eines Operators eine Region mit signifikanten Informationen in einem Bild identifiziert [43]. Im Folgenden wird diese Region mit einem Verfahren ähnlich der einfachen blockbasierten Korrelation nach dem korrespondierenden Punkt durchsucht.

In Abhängigkeit des Operators und des Bildinhaltes ergeben sich unterschiedlich dichte Tiefenkarten. Mögliche Operatoren sind neben den Interest-Operatoren aus Abschnitt 2.4.2.1 zum Beispiel statistische Momente (2. Ordnung  $\hat{=}$  der Varianz) eines Blockes und verschiedene Kanten-Filter wie zum Beispiel der Canny-Operator. Ein Problem der statistischen Momente gegenüber den Interest-Operatoren ist ihre Eigenenschaft, nur eine hohe Entropie [36] innerhalb des Blockes anzuzeigen, ohne jedoch eine Aussage über die Eigenschaften bezüglich der Umgebung zu treffen. Wenn ein Block mit hoher Entropie mehrfach auftritt, zum Beispiel bei einem Zaun, kann es zu einer fehlerhaften Zuordnung kommen.

**Klassische Bildpyramiden** Die Bildpyramiden wurden zunächst für adaptive Komprimierungszwecke [97] in die Bildverarbeitung eingeführt. Hierbei wird zwischen hierarchischen Verfahren durch Verminderung der Anzahl der Pixel des Bildes [54] und der Grauwertaufösung der Pixel unterschieden. Für die Korrespondenzanalyse stellt die Reduktion der Grauwertaufösung eher einen Nachteil dar, da sich die Anzahl der Mehrdeutigkeiten dadurch deutlich erhöht. Somit kommen für die Korrespondenzanalyse nur Bildpyramiden infrage, die durch Variation der Bildauflösung [79] erstellt(siehe Abbildung 2.7) werden.

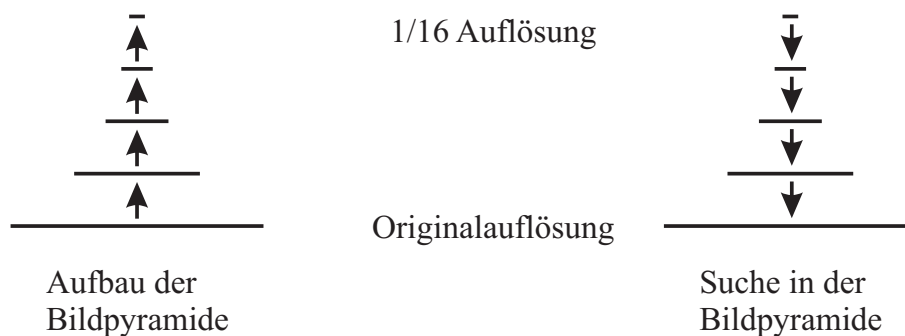


Abbildung 2.7: Klassische Bildpyramide

Durch die Verwendung von Bildpyramiden können auch bei Mehrdeutigkeiten oder schwacher Entropie der Informationen zuverlässige Zuordnungen der Blöcke durchgeführt werden. Dazu werden aus beiden Bildern entsprechend Bildebenen mit niedrigerer optischer Auflösung generiert. Ausgehend von der am niedrigsten aufgelösten Bildebene wird nun nach korrespondierenden Punkten gesucht. In dieser wird eine normale blockbasierte Korrelation durchgeführt, wobei der Suchbereich relativ klein ist. Das Ergebnis wird anschließend als Startwert für die Suche in der nächst höher aufgelösten Ebene verwendet. In dieser wird dann ausgehend von



dem Startwert ebenfalls eine Suche durchgeführt und so wird dies bis zur höchst aufgelösten Ebene fortgesetzt. Der Suchbereich in den höher aufgelösten Ebenen wird durch das Ergebnis der niedriger aufgelösten stark eingeschränkt, wodurch sich für die Softwarelösung eine gewisse Rechenzeitoptimierung ergibt. Weiterhin hat das Verfahren den Vorteil, dass bei sehr fein abgestimmten Ebenen eine sehr gute Sicherheit für die Korrespondenzen erreicht wird.

Eine weitere Variante ist eine Korrelation mit Blöcken unterschiedlicher Abmaße und der bei ihnen angewendeten Schrittweiten bei der blockbasierten Korrelation [37].

**Gabor-Filter** Mit Gabor-Filtern wird die Frequenzinformation innerhalb von Bildern berücksichtigt. Bei der Gabor-Transformation handelt es sich um eine gefensterter Form der Fouriertransformation [53]. Mit ihr können sehr gut Texturen untersucht und miteinander verglichen werden. Dies kann auch für die Auswertung von Stereobildern genutzt werden.

Zum Beispiel kann mittels Gabor-Filtern die Bedeutung unterschiedlicher Frequenzen für die Stereosuche bei großen Bildbereichen [57] untersucht werden.

Durch Anwendung eines Satzes aus unabhängigen Gabor-Filtern [23] kann eine sehr präzise Stereobildauswertung erfolgen. Es werden die Tiefenkarten beider Bilder ermittelt und anschließend erfolgt eine iterative Optimierung der 3D-Information jedes Punktes.

### 2.4.3 Korrelationskriterien

Um rechnerisch eine Aussage über die Ähnlichkeit bzw. die statistische Abhängigkeit zweier Signalfolgen treffen zu können, werden Korrelationskriterien verwendet. Der Einsatzbereich der Korrelationskriterien umfasst die gesamte digitale Signalverarbeitung, von der digitalen Korrelation von Signalen im Mobilfunk bis hin zur Bildverarbeitung. Aber auch Messreihen im Allgemeinen werden mit Hilfe von Korrelationskriterien auf Ähnlichkeit hin untersucht.

Korrelationskriterien können in Anlehnung an [9] in Ähnlichkeits- (KKF, NKKFMF) und Fehlerkriterien (SAD, SSD, NSSDMF) unterteilt werden. Im Folgenden werden die für diese Arbeit relevanten Kriterien vorgestellt. In Kapitel 4 werden einige Kriterien bezüglich ihres Rechenaufwandes und ihrer Eignung zur Echtzeitimplementierung für verschiedene Plattformen untersucht.

Bei den folgenden Kriterien werden jeweils alle durch  $\xi, \eta$  definierten Blockkombinationen für eine Pixelposition  $x, y$  des Referenzblockes berechnet, wobei  $P_r(i, j)$  und  $F(\xi + i, \eta + j)$  die Grauwerte der Pixel des Referenzblockes  $P_r(x, y)$  und des Suchblockes  $F(x + \xi, y + \eta)$  sind.  $m, n$  beschreiben die Ausdehnung des Referenz- bzw. Suchfensters,  $i, j$  die konkrete Pixelposition innerhalb der Blöcke und  $\xi, \eta$  die Verschiebung des Suchfensters gegenüber der Position des Referenzblockes. Ein Beispiel hierfür ist in Abbildung 2.6 dargestellt.

Eine große Familie von Fehler- bzw. Unähnlichkeitskriterien stellen die Summe der Differenzenquadrate (SSD) bzw. Differenzenbeträge (SAD) dar. Bei ihnen werden, wie die Namen schon

sagen, die Differenzen der Signalfolgen aufsummiert. Der enorme Vorteil dieser Verfahren ist ihre einfache Realisierung. Jedoch sind sie gegenüber Helligkeitsdifferenzen der Bilder empfindlich. Daher sollte entweder eine gute und für beide Stereokameras synchron realisierte Helligkeitsregelung eingesetzt werden, oder statt der Pixel  $P_r(i, j)$  und  $F(\xi + i, \eta + j)$  die mittelwertfreien Pixel  $\overline{P_r(i, j)}$  und  $\overline{F(\xi + i, \eta + j)}$  verwendet werden.

$$SAD(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} |P_r(i, j) - F(\xi + i, \eta + j)| \quad (2.6)$$

$$SSD(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (P_r(i, j) - F(\xi + i, \eta + j))^2 \quad (2.7)$$

$i, j$	-	Pixelposition innerhalb des Blockes
$m, n$	-	Blockgröße
$\xi, \eta$	-	Verschiebung des Suchblockes gegenüber dem Referenzblock in x- und y-Richtung

Eine weitere große Familie von Korrelationskriterien stellen die Kreuzkorrelationsfunktionen dar. Sie gehören zu den Ähnlichkeitsmaßen.

Bei der Kreuzkorrelationsfunktion werden Blöcke unterschiedlicher Bilder miteinander verglichen. Für ein breites Einsatzspektrum wurde eine große Anzahl an Varianten geschaffen.

$$KKF(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (P_r(i, j) \cdot F(\xi + i, \eta + j)) \quad (2.8)$$

Ebenso wie bei der SSD und der SAD-Funktion ist der Wertebereich unbegrenzt. Sie funktioniert auch bei verrauschten Signalfolgen noch sehr gut [9].

Durch Verwendung der mittelwertfreien Grauwerte der Pixel kann eine Unabhängigkeit von einem additiven Unterschied zwischen beiden Folgen erreicht werden.

$$KKFMF(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right) \quad (2.9)$$

Diese Funktion ist damit lokal mittelwertfrei. Dies ist bereits ein entscheidender Vorteil für die Auswertung, da positive Werte eine Äquivalenz und negative Werte eine Antivalenz ausdrücken. Der Betrag der Werte ist jedoch unbegrenzt. Somit kann kein globaler Schwellwert definiert werden. Es fällt auf, dass das Produkt der beiden mittelwertfreien Signalfolgen die Kovarianz der Signalfolgen darstellt. Durch Normierung mit der Wurzel des Produktes der beiden Varianzen der Signalfolgen entsteht die normierte mittelwertfreie Kreuzkorrelationsfunktion. Bei

identischen Signalfolgen ergibt sich unabhängig vom Wertebereich der Wert +1 und für entgegengesetzte Signalfolgen der Wert -1. Nimmt das Ergebnis den Wert Null an, sind die zwei Signalfolgen nicht miteinander korreliert.

$$NKKFMF(\xi, \eta) = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right)}{\sqrt{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2}} \quad (2.10)$$

$\overline{P_r(i, j)}$	-	mittelwertfreie Grauwerte der Pixel des Referenzblockes
$\overline{F(\xi + i, \eta + j)}$	-	mittelwertfreie Grauwerte der Pixel des Suchblockes

Die NKKFMF ist lokal mittelwertfrei und wertebegrenzt. Dadurch sind eindeutige Aussagen und Vergleiche über eine Signalfolge hinaus möglich. Dies kann bei der Stereobildverarbeitung einen deutlichen Vorteil darstellen.

Um ein robustes Unähnlichkeitsmaß zu realisieren, wurde an der INRIA<sup>2</sup> das Normierungssystem der NKKFMF mit der SSDMF verknüpft. Ein Nachteil der Unähnlichkeitsmaße ist der nach oben unbeschränkte Wertebereich. Nach unten stellt die Null das absolute Minimum dar und drückt vollständige Übereinstimmung aus.

$$NSSDMF(\xi, \eta) = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} - \overline{F(\xi + i, \eta + j)} \right)^2}{\sqrt{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2}} \quad (2.11)$$

Durch Vereinfachung der normierten mittelwertfreien KKF entsteht das Moravec-Kriterium Gleichung. 2.12. Es wird auch als pseudonormierte KKF bezeichnet. Das Moravec-Kriterium hat eine leichte Sensitivität gegenüber multiplikativen Störungen. Nach Aschwanden [9] ist es sehr stark von den Aufnahmebedingungen abhängig, insbesondere ob multiplikative Helligkeitsschwankungen auftreten.

$$MOR(\xi, \eta) = \frac{2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right)}{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 + \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2} \quad (2.12)$$

<sup>2</sup>Institut National de Recherche en Informatique et en Automatique - das Französische Institut für Informatik und Automatisierungstechnik

Die Funktion ist zwar laut Aschwanden [9] nicht ganz so robust gegenüber Störungen wie die NKKFMF, jedoch bietet sie einige Vorteile hinsichtlich des Rechenaufwandes. Durch den Wegfall der Wurzel und die geringere Anzahl von Multiplikationen kann die Funktion in Hardware und in Software effektiver implementiert werden.

Aus der Literatur [9] [79] ist bekannt, dass die vorgenannten Korrelationskriterien bei gleichen Eingangsdaten, insbesondere an ausgeprägten Kanten im Bild, nahezu identische Werte für die Disparität liefern. Daher können sie zur Nutzung ihrer spezifischen Vorteile weitgehend gegeneinander ausgetauscht werden. Ein Beispiel hierfür wird im Anhang B.4 gegeben.

Aschwanden [9] untersucht in seiner Arbeit die Eigenschaften verschiedener Korrelationskriterien. Im Vordergrund stehen dabei multiplikative und additive Störungen sowie verschiedene Arten von Rauschen. Er beschreibt die NKKFMF als sehr robust gegenüber allen untersuchten Störungen. Die SAD und die SSD sind dagegen deutlich weniger robust gegenüber additiven und multiplikativen Störungen. Die additiven Störungen können jedoch durch die Verwendung der lokal mittelwertfreien Funktionen weitestgehend ausgeglichen werden. Multiplikative Störungen entstehen zum Beispiel durch die Anpassung der Blende bei einem Einkamerasystem oder durch unterschiedliche Blenden bei einem Stereokamerasystem. Wenn die Stereokameras exakt aufeinander eingeregelt werden, sollten bei Stereokamerasystemen multiplikative Störungen vermieden werden können.

Im Anhang B werden die Korrelationskriterien SAD, SADMF, SSD, SSDMF, NKKFMF und QNKKFMF bezüglich ihrer Eigenschaften und ihrer Wirkung auf synthetische und reale Bilder untersucht. Es wurden Untersuchungen zur Wirkung auf Objekte unterschiedlicher Breite im Vergleich zur Blockgröße durchgeführt. Für Bilder mit schmalen Objekten konnten alle Funktionen gute Ergebnisse erzielen. Bei breiten Objekten wurden die besten Ergebnisse durch die SAD und die NKKFMF bzw. QNKKFMF erzeugt. Für reale Bilder und Bilder mit leicht sichtbaren Helligkeitsabweichungen wurden die deutlichsten Maxima durch die SADMF und die QNKKFMF erzielt. Da diese Bilder in den in Kapitel 4 beschriebenen Anwendungsfällen dominieren, werden diese beiden Funktionen weiter untersucht und in der folgenden Arbeit verwendet.

#### 2.4.4 Ermittlung dichter kontinuierlicher Tiefenkarten

Bei den bisher betrachteten Verfahren zur Stereobildauswertung entstehen Punktwolken mit verhältnismäßig geringer Auflösung gegenüber der Bildauflösung. Dies ist durch die vorherige Auswahl relevanter Punkte über Kantenfilter, statistische Momente oder die Autokorrelationsfunktion begründet. Zur detaillierten Auswertung der Tiefeninformation besteht für einige Anwendungsfälle das Interesse an sehr dichten und kontinuierlichen Tiefenkarten.

Die Berechnung dichter kontinuierlicher Tiefenkarten erfordert zusätzliche Arbeitsschritte gegenüber den bisher beschriebenen Verfahren. Diese Arbeitsschritte sind teilweise in der Bildvorverarbeitung angesiedelt, greifen im Wesentlichen jedoch direkt in den Korrelationsprozess ein oder werden als Nachverarbeitung durchgeführt. Es werden im Anschluss die Verfahren von der

Carnegie Mellon University (Verfahren 1) und das Verfahren von van der Mark und Garvillia (Verfahren 2) vorgestellt und diskutiert.

**Verfahren 1** An der Carnegie Mellon University wurde von Zitnik und Kanade [102], [103] ein Verfahren zur Ermittlung dichter und kontinuierlicher Tiefenkarten entwickelt. Es basiert auf der iterativen Verbesserung einer durch blockbasierte Korrelation ermittelten Tiefenkarte. Diese Tiefenkarte wird durch einem Algorithmus mit fünf Schritten verbessert.

- Erstellen einer Disparitätskarte mit einer Auflösung äquivalent zum Referenzbild
- Initialisieren des Übereinstimmungskriteriums für jeden Punkt der Disparitätskarte auf Basis eines korrelationsbasierten Verfahrens (KKFMF oder SAD)
- Iterative Verbesserung der Übereinstimmungswerte mittels Gleichung 2.13 bis zur Konvergenz
- für jedes Pixel wird die Disparität mit der maximalen Übereinstimmung gesucht.
- Wenn das Maximum über dem Schwellwert liegt, wird die Disparität ausgegeben, andernfalls wird der Wert von der Weiterverarbeitung ausgeschlossen.

Dabei wird davon ausgegangen, dass die Disparität eine Kontinuität bezüglich der Umgebung aufweist, sich nicht sprunghaft ändert und eine eindeutige Antwort für das Korrespondenzproblem existiert. Über die Funktion

$$L_{n+1}(r, c, d) = L_0(r, c, d) * \left( \frac{S_n(r, c, d)}{\sum_{(r'', c'', d'') \in \Psi(r, c, d)} S_n(r'', c'', d'')} \right)^\alpha \quad (2.13)$$

$L_{n+1}(r, c, d)$	-	(n+1)-te Iteration der Ähnlichkeit
$L_0(r, c, d)$	-	Ähnlichkeitskriterium mit dem der 3D-Punkt initialisiert wird
$r$ und $c$	-	Bildkoordinaten
$d$	-	Disparität
$S_n(r, c, d)$	-	räumlich lokaler Bezug der einzelnen Punkte zueinander
$n$	-	Nummer der aktuellen Iteration
$\alpha$	-	eine Konstante, die die Stärke der Veränderung je Iteration angibt

wird das Optimum bezüglich des Ähnlichkeitskriteriums und der umgebenden Disparitäten bestimmt. Dabei ist  $L_0(r, c, d)$  das Ähnlichkeitskriterium, mit dem der 3D-Punkt jeweils initialisiert wird.  $r$  und  $c$  sind die Bildkoordinaten und  $d$  die Disparität.  $S_n(r, c, d)$  drückt hierbei

den räumlich lokalen Bezug der einzelnen Punkte zueinander aus.  $\alpha$  ist eine Konstante, die die Stärke der Veränderung angibt. Wird sie zu klein gewählt, wird die Konvergenz nicht erreicht. Abschließend werden verdeckte Bereiche ermittelt und markiert.

In Beispielbildern hatten ca. 60 % der Korrelationswerte initial bereits die Position mit der maximalen Übereinstimmung inne. Für die restlichen Werte wurde das Iterationsverfahren angewendet.

Die in den Veröffentlichungen gezeigten Tiefenkarten sind sehr dicht, gleichmäßig und kontinuierlich, abgesehen von verdeckten Bereichen, in denen keine Tiefenwerte ermittelt werden können. Die Ermittlung der wahrscheinlichsten Position über eine Iteration unter Verwendung des Korrelationskriteriums und der unmittelbaren Nachbarschaft ist sehr rechenaufwändig und die Anzahl der Schritte, bis sich eine Konvergenz ergibt, kann im Vorfeld nur schwer bestimmt werden. Der iterative Ansatz ist stark kontrollflussabhängig und kann nur schwer parallelisiert werden, da insbesondere die Nachbarschaft mit einbezogen wird. Das Verfahren ist für eine sehr genaue Abbildung der Umgebung optimiert, für eine Hinderniserkennung in Echtzeit in eingebetteten Systemen aufgrund des Rechenaufwandes nur bedingt geeignet, auch wenn dichte Tiefenkarten die Segmentierung von Objekten erleichtern.

**Verfahren 2** Von Van der Mark und Garvilla [58] wird eine Gruppe von Auswerteverfahren für dichte Tiefenkarten bezüglich Leistungsfähigkeit und Performance untersucht. Ziel ist eine dichte Tiefenkarte zur sicheren Detektion des Umfeldes im Stadtverkehr in Echtzeit. Es werden zunächst verschiedene Verfahren, die nachfolgend kurz angerissen werden, eingeführt und dann in Kombinationen zu 7 Auswerteverfahren zusammengefügt, welche dann miteinander verglichen werden. In allen Verfahren wird die SAD-Funktion zur Korrelation eingesetzt. Da die SAD-Funktion gegenüber Helligkeitsunterschieden in den verglichenen Bildern empfindlich ist, werden die Bilder zunächst mit dem Marr-Hildreth-Operator [59] gefiltert. Hierbei wird auf einen 2D-Glättungsfilter mit Gauss-Verteilung ein Laplace-Filter angewendet, daher wird diese Operation auch oft Laplacian of Gaussian (LoG) genannt. Es entsteht die 2. Ableitung der Gauss-Verteilung.

Durch die Funktion

$$\Delta g(i, j) = -\frac{1}{\pi\sigma^4} \cdot e^{-\frac{i^2+j^2}{2\sigma^2}} \left(1 - \frac{i^2 + j^2}{2\sigma^2}\right) \quad (2.14)$$

wird der Koeffizient in Abhängigkeit von der Position in der Filtermatrix und damit der Filterkern beschrieben.  $i$  und  $j$  geben die Position innerhalb der Filtermaske an. Die Standardabweichung  $\sigma$  gibt die Breite der Gauss-Verteilung und damit auch die Breite des Filterkerns an. Für

ein  $\sigma$  von 1,0 und eine Filtergröße von 5x5 Pixel ergibt sich zum Beispiel folgender Filterkern

$$g(i, j) = \begin{bmatrix} 1 & 2 & 2 & 2 & 1 \\ 2 & 0 & -5 & 0 & 2 \\ 2 & -5 & -16 & -5 & 2 \\ 2 & 0 & -5 & 0 & 2 \\ 1 & 2 & 2 & 2 & 1 \end{bmatrix} \quad (2.15)$$

Im Folgenden werden verschiedene Ansätze zur Auswertung und Verbesserung von Tiefenkarten kurz umrissen, die in [58] ausführlich diskutiert werden.

**Suche in beide Richtungen** Hierbei wird sowohl die Suche mit dem linken Bild als Referenzbild und dem rechten Bild als Suchbild als auch der umgekehrte Weg mit dem rechten Bild als Referenzbild berechnet und die Sicherheit durch die Kombination beider Tiefenkarten gewonnen.

**Fünf-Fenster-Ansatz** Der Mehrfensteransatz wurde von Hirschmüller [33] detailliert untersucht. Er basiert auf der Annahme, dass die Betrachtung eines Objektes mit einem Fenster durch eine ungünstige Position des Fensters fehlerhafte Tiefenwerte erzeugen kann. Daher werden 5 quadratische Fenster, die sich gegenseitig überlappen, erstellt und die Gruppe von 5 Fenstern gemeinsam zu einem Tiefenwert zusammengefasst.

**Vertrauenswürdigkeit der Disparität** Zur Prüfung der Vertrauenswürdigkeit  $C$  der Disparität werden das im Sinne der entsprechenden Korrelationsfunktion beste  $C_1$  und das zweitbeste Extremum  $C_2$  ausgewählt. Durch den relativen Abstand (siehe Gleichung 2.16) zum zweitbesten Extremum ergibt sich ein Maß für die Eindeutigkeit dieser Korrelation und damit für die Qualität.

$$C = \frac{C_2 - C_1}{C_1} \quad (2.16)$$

**Suchoptimierung** Üblicherweise wird bei korrelationsbasierten Verfahren das Extremum der Disparitätskarte als Disparitätswert ausgewählt. Kraft und Jonker [45] haben einen Ansatz auf Basis dynamischer Programmierung untersucht. Dabei wird zunächst eine Kostenfunktion für die Ermittlung der Disparitäten erstellt und darüber der beste Disparitätswert durch Betrachtung der Nachbarschaft ermittelt.

**Recover-Ansatz** Beim Recover-Ansatz erfolgt ähnlich wie beim Verfahren 1 der Carnegie Mellon University die Betrachtung der Nachbarschaft. Es wird hierbei ebenfalls von der Möglichkeit einer eindeutigen Zuordnung ausgegangen. Es werden alle Extrema mit einem Zuverlässigkeitswert belegt. Ergibt sich aus der Nachbarschaft eine Diskontinuität, wird nach einem besser zur Nachbarschaft passenden Maximum gesucht.



**Auswertelgorithmen** Aus den für das Verfahren 2 beschriebenen Ansätzen zur Optimierung werden 7 verschiedene Algorithmen zur Stereobildauswertung auf synthetischen und realen Bildern miteinander verglichen. Es werden folgenden Kombinationen in [58] diskutiert.

$SAD_L$	einfache Suche von links nach rechts
$SAD_{rec}$	Recover-Ansatz: Überprüfung der Disparität durch die Nachbarschaft und durch die Eindeutigkeit
$SAD_{LR}$	zusätzliche Überprüfung durch die Suche von rechts nach links
$SAD_{MW5L}$	wie oben jedoch mit Fünf-Fenster-Ansatz
$SAD_{MW5rec}$	wie oben jedoch mit Fünf-Fenster-Ansatz
$SAD_{MW5R}$	wie oben jedoch mit Fünf-Fenster-Ansatz
$SAD_{DP}$	Suche mit dynamischer Programmierung

Den synthetisch erstellten Bildern werden kontrolliert Störquellen hinzugefügt. Bei der Untersuchung der realen Bilder werden rektifizierte Bilder oder Bilder, die im Normalfall der Stereophotogrammetrie aufgenommen wurden, vorausgesetzt.

Zur Bestimmung der Geschwindigkeit wurden die Algorithmen mit den SIMD-Erweiterungen SSE2 der Pentium 4 und Athlon 64 Prozessoren implementiert.

Van der Mark und Garvilla kommen zu dem Schluss, dass die SAD-Funktion mit der beidseitigen Suche ein Optimum bezüglich der Fehleranfälligkeit und der Geschwindigkeit darstellt. Der Fünf-Fenster-Ansatz erzeugt eine geringere Fehlerquote insbesondere an den Übergängen von einem Objekt zum anderen, allerdings erhöht sich die Verarbeitungszeit maßgeblich.

Der Einsatz der SAD-Funktion setzt eine gute Vorverarbeitung voraus, damit die Korrespondenzen gefunden werden. Die Filterung mit dem Marr-Hildreth-Operator ist sehr verbreitet. Der Einsatz der bezüglich Helligkeitsschwankungen robusteren SADMF würde die SIMD-optimierte Implementierung verhindern, da durch die lokale Mittelwertbildung alle Werte innerhalb des Fensters von den durch die Verschiebung hinzugefügten und entfallenden Grauwerten abhängen.

Ein weiteres Verfahren zur Erzeugung dichter Tiefenkarten wurde von der Firma Point Grey Research im System Bumblebee 2 implementiert. Dieses Verfahren wird im folgenden Abschnitt zusammen mit anderen Verfahren für gegenwärtig verfügbaren echtzeitfähigen Lösungen beschrieben.



### 2.4.5 Lösung des Korrespondenzproblems in verfügbaren echtzeitfähigen Stereoauswertesystemen

Eine der ersten echtzeitfähigen Hardwareimplementierungen zur Stereobildauswertung ist eine Kombination von mehreren ALUs<sup>3</sup> zur Realisierung einer SAD-Funktion von Saneyoshi [78]. Die damit erzeugten Tiefenkarten haben eine Auflösung von 100x50 3D-Punkten, erstellt aus Stereobildern mit einer Auflösung von 500x200 Pixel und einer Bildfrequenz von 10 Hz in Echtzeit.

Die amerikanische Firma Point Grey Research [73] bietet das System Bumblebee 2 [74] mit kontinuierlicher hardwareunterstützter Stereobildauswertung an. Das System besteht aus einem sehr kompakten Modul mit integriertem 2-Kamerasystem mit fixer Basisbreite. Die Auflösung der CCD-Kameras ist jedoch vergleichsweise gering. Für den Nahbereich im Innenraum ist das System gut geeignet. Für den Außeneinsatz ist es aufgrund der fest eingebauten CCD-Kameras nur bedingt geeignet. Das Gerät ist weiterhin primär für den Einsatz an einem PC vorgesehen und verfügt nur über eine Firewire-Schnittstelle. Die von der Firma mitgelieferten Treiber werden zum Erstellen einer Tiefenkarte benötigt.

Im Bumblebee 2 System wird ein Ansatz zur Erzeugung dichter Tiefenkarten verwendet, der dem im Abschnitt 2.4.4 ähnlich ist. Basis zur Erzeugung einer dichten Tiefenkarte sind rektifizierte Bilder. Zur Unterdrückung von Rauschen in den Bildern wird eine Tiefpassfilterung durchgeführt. Die Korrelation erfolgt mittels SAD-Funktion. Um eine gute Zuordnung zu gewährleisten, wird die Helligkeit beider Kameras gemeinsam angepasst.

Anschließend erfolgt eine Validierung der Disparität über die Textur. Auch in diesem Ansatz wird die Eindeutigkeit des Korrespondenzproblems vorausgesetzt. Um fehlerhafte 3D-Punkte auszuschließen, wird wie im Abschnitt 2.4.4 ebenfalls die Vertrauenswürdigkeit ermittelt. Mit Hilfe einer Subpixelinterpolation wird die Genauigkeit der Tiefenkarte erhöht. Dadurch werden fehlerhafte Punkte ausgeschlossen. Abschließend werden mit Hilfe einer Oberflächenfilterung Ausreißer entfernt.

Ein weiteres Beispiel ist das PCI-Board [21] mit Stereokamerakopf der Firma TYXZ [35], welches ein PC-basiertes System zur Erstellung von Online-Tiefenkarten auf Hardwarebasis darstellt. Weiterführende Algorithmen zur Objektdetektion und -erkennung können in Software auf einer leistungsfähigen PC-Workstation für eine schnelle kontinuierliche Verarbeitung realisiert werden. Die Qualität der 3D-Daten dieses Systems ist gut.

Auch dieses System setzt rektifizierte Bilder zur Ermittlung voraus. Werden die Bilder mit den systemkompatiblen Stereoköpfen der Firma TYXZ aufgenommen, können sie in Hardware rektifiziert werden. Auf die Bilder wird blockweise die Census-Transformation angewendet. Für die aus den Blockkombinationen resultierenden Binärvektoren beider Bilder wird die kürzeste Hamming-Distanz gesucht. Die Qualität der Hamming-Distanz wird bewertet und dient als Aussage über die Zuverlässigkeit dieses 3D-Punktes.

---

<sup>3</sup>Arithmetik Logic Unit

Die Census-Transformation wurde von Zabih und Woodfill [101] erstmals für die Stereobildverarbeitung eingesetzt. Sie binarisiert die Blöcke nach dem Grauwert des zentralen Pixels des Blockes.

$$\begin{bmatrix} 41 & 154 & 115 & 211 & 27 \\ 203 & 67 & 21 & 137 & 246 \\ 79 & 167 & (58) & 255 & 1 \\ 135 & 176 & 233 & 20 & 198 \\ 42 & 191 & 39 & 113 & 209 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & X & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \longrightarrow 01110\ 11011\ 11010\ 11101\ 01011 \tag{2.17}$$

In Ausdruck 2.17 ist links der zu transformierende Block dargestellt. Der Wert 58 ist der zentrale Pixel. Die Positionen, deren Grauwerte größer als 58 sind, werden mit 1 markiert, alle anderen mit 0. Auf der rechten Seite steht der resultierende Bitvektor, welcher für den Referenzblock und den Suchblock berechnet wird. Anschließend wird über die kürzeste Hamming-Distanz die ähnlichste Blockkombination und so die Disparität ermittelt.

Die Firma BOSCH [2] hat ein Steuergerät für Fahrzeuge entwickelt, welches sowohl eine 3D-Tiefenkarte erstellt als auch Objekte darin detektieren kann. Dieses System wird gegenwärtig bei verschiedenen namhaften Automobilherstellern zur Erforschung von Fahrerassistenzsystemen verwendet. Dies lässt auf eine rasche Weiterentwicklung sowie eine baldige Serieneinführung und Zertifizierung in naher Zukunft schließen.

### 2.4.6 Vor- und Nachteile beim Einsatz von Verfahren zur Erzeugung dichter Tiefenkarten

Die Vorteile von dichten Tiefenkarten liegen auf der Hand. Durch eine große Anzahl von Punkten kann ein Objekt oder eine Szene vollständig dreidimensional erfasst werden. Die Punkte, an denen keine 3D-Informationen vorhanden sind, da sie in einem der beiden Bilder verdeckt sind, sind ebenfalls markiert.

Diese Verfahren für dichte Tiefenkarten produzieren aus einem Stereobildpaar mit 1024x1024 Punkten Tiefenkarten mit bis zu 512x512 3D-Punkten. Für einen menschlichen Betrachter sind diese Tiefenkarten sehr angenehm. Aufgrund der durchgängigen Oberflächenstruktur sind die Objekte klar zu erkennen und können anhand ihrer Form gut klassifiziert werden. Ebenso wie ein Bild enthält eine dichte Tiefenkarte eine große Redundanz. So können fehlerhafte Blöcke gut durch die Betrachtung der Nachbarschaft ausgeglichen werden.

Nachteilig wirken sich dichte Tiefenkarten auf die Verarbeitungszeit aus. Dies ist insbesondere bei der Echtzeitverarbeitung auf eingebetteten Systemen problematisch. Dabei ergibt sich ein hoher Rechenaufwand bei der Erstellung der Tiefenkarten und bei der Nachverarbeitung. Durch die hohe Anzahl der Punkte ist bereits der Aufwand bei der blockbasierten Korrelation deutlich höher. Herkömmliche Verfahren mit einfacher blockbasierter Korrelation, wie z.B. das System Deep Sea der Firma TYZX, erzeugen bei einer Straßenszene ca. 30.000 - 50.000 verwendbare

3D-Punkte. Bei den beschriebenen Verfahren für dichte Tiefenkarten liegen die Auflösungen 5 - 8 mal höher. Durch die zusätzlichen Rechenschritte zur Berechnung der Eindeutigkeit und Sicherheit ggf. die iterative Verbesserung der 3D-Punkte und die Betrachtung der Nachbarschaft erhöht sich der Rechenaufwand zusätzlich.

Durch die hohe Redundanz sind für die Auswertung andere Algorithmen erforderlich. Ebenso wie aus einem Bild ein Objekt nur bedingt direkt erkennbar ist und z.B. ein Kantenfilter für die Erkennung eingesetzt werden muss, sind die Objekte in dichten Tiefenkarten auch nicht klar umrissen und müssen durch entsprechend komplexe Algorithmen extrahiert werden. Somit ist nach der Erstellung der 3D-Punkte für die Auswertung der Tiefenkarte ebenfalls ein erhöhter Arbeitsaufwand erforderlich.

Die 3D-Punkte stellen prinzipiell eine Abtastung der Oberfläche des Objektes dar. Nach dem Abtasttheorem ist mindestens die doppelte Abtastfrequenz der höchsten im Signal vorhandenen Frequenz erforderlich. Für die Hinderniserkennung stellt ein Objekt einen Impuls in der Tiefenkarte da. Die Informationen Höhe (Entfernung) und Breite des Impulses sind ausreichend, um das Objekt einzuordnen.

In Abhängigkeit von der Kamerakonfiguration und der minimal relevanten Objektbreite in einer bestimmten Entfernung ergibt sich somit eine zwingend erforderliche Abtastrate auf dem Bild, um eine Hindernisdetektion durchführen zu können.

Die minimal relevante Objektbreite ergibt sich durch die Aufgabenstellung. Für Prozesse der Qualitätssicherung kann diese Objektbreite im  $\mu\text{m}$ - Bereich und für die Hinderniserfassung im dm-Bereich liegen. Ist ausreichend Rechenleistung vorhanden, ist die Verwendung von dichten Tiefenkarten kein Hindernis. Kann jedoch nur auf eine begrenzte Rechenleistung zurückgegriffen werden, sollte der Einsatz von dichten Tiefenkarten für jede Anwendung kritisch hinterfragt werden.

## 2.4.7 Bildvorverarbeitung

Wichtig für das Ergebnis der Stereobildauswertung ist auch die Bildqualität. Die Bildschärfe wird an dieser Stelle nicht behandelt, da die Kameras nur auf eine Einstellung kalibriert werden können. Daher muss vor der Kalibrierung der Bereich der scharfen Abbildung ausgewählt werden. Weiterhin ist eine geringe Verzerrung der Bilder für eine gute Tiefenkarte erforderlich. Objektive mit geringer Brennweite besitzen einen großen Öffnungswinkel, der bei einer Umfeldfassung im Nahbereich erforderlich ist. Objektive mit einem großen Öffnungswinkel erzeugen somit auch stärkere Verzerrungen. Diese Thematik wird im Abschnitt [4.2.2.1](#) näher diskutiert.

### 2.4.7.1 Helligkeit

Zwei wesentliche Störquellen für die Stereobildauswertung sind das Bildrauschen und die Helligkeit. Die Anpassung der Helligkeit durch Variation der Blende ist zwar ein geringerer Eingriff

in den Strahlengang als die Verstellung des Brennpunktes, sie stellt ebenso einen Eingriff in den Strahlengang dar und sollte daher vermieden werden. Daher ist es vorteilhaft für die Stereophotogrammetrie, Anpassungen der Helligkeit durch die Variation der Integrationszeit vorzunehmen. Dies kann für Kameras mit einem globalen Shutter realisiert werden. Seit einigen Jahren sind auch CMOS-Kameras mit einem globalen Shutter erhältlich.

Ein anderer Ansatz ist die Anpassung der Helligkeit während des Vergleiches der Bilder durch entsprechende Faktoren. Dies ist für eine Echtzeitauswertung jedoch ungünstig.

Bei Einsatz einer Helligkeitsregelung für ein Stereokamerasystem ist es erforderlich, beide Kameras mit den gleichen Parametern anzusteuern [42]. Bei Einsatz identischer Kameras sind dann über weite Dynamikbereiche gute Resultate erzielbar. Durch Verwendung sehr gut aufeinander abgestimmter Kameras kann auf die Betrachtung der lokal mittelwertfreien Blöcke verzichtet werden.

### 2.4.7.2 Bildrauschen

Bildrauschen wirkt sich je nach Fenstergröße und Intensität sehr unterschiedlich aus. Insbesondere CMOS-Sensoren haben häufig ein hohes statisches Rauschen, welches sich mit steigender Temperatur erhöht. Um das Rauschen zwischen beiden Bildern auszugleichen, ist es zum einen möglich, robuste Korrelationskriterien (z.B. NKKFMF) zu verwenden. Andererseits können die Bilder vor der Verarbeitung durch entsprechende Glättungsfilter [12] vorverarbeitet werden. Wesentlich hierbei ist die Berücksichtigung der Tatsache, dass verschiedene Filter Verschiebungen um 0,5 oder 1 Pixel verursachen. Deshalb sollten symmetrische Filtermasken ohne Versatz verwendet werden.

## 2.5 Diskussion der Verfahren zur multikularen 3D-Vermessung

Die Verwendung eines Stereokamerasystems zur 3D-Vermessung ermöglicht die sichere und zuverlässige Ermittlung der 3D-Informationen über einen großen Entfernungsbereich. Ein solches Messsystem kann vollständig passiv realisiert werden, wodurch die Beeinträchtigung der Umwelt minimiert wird und für große Entfernungen keine extreme Lichtleistung erforderlich ist.

Die Ermittlung der 3D-Informationen ist unabhängig von Form und Größe der zu vermessenden Objekte. Viele monokulare Messsysteme bedürfen aktiver Beleuchtung mit bekannter Lichtquelle oder aber a priori Informationen über die zu vermessenden Objekte. Ausgenommen hiervon sind die Messmethoden auf Basis des optischen Flusses und die der Vermessung durch Variation des Fokus. Beide Methoden sind jedoch nur bei statischen oder bekannten dynamischen Situationen sicher einsetzbar. Dynamische Messsituationen, wie sie im Straßenverkehr oder bei fahrerlosen Transportfahrzeugen auftreten, können mit einem Stereokamerasystem mit hoher Bildfrequenz gut abgetastet werden.

Der wesentliche Nachteil der stereophotogrammetrischen Messmethoden ist der hohe Aufwand für die Auswertung der Bilder und die Gewinnung der 3D-Informationen. Die klassischen Methoden der digitalen Photogrammetrie mittels Interest-Operatoren bieten nur eine sehr dünne Tiefenkarte und sind vorwiegend auf die Vermessung von statischen Objekten wie z.B. Fassaden ausgerichtet. Ziel ist die hochpräzise Vermessung, sodass nur Punkte mit einer eindeutigen Korrespondenz verwendet werden. Dadurch könnten bei der Umfelderfassung einzelne Objekte unberücksichtigt bleiben.

Sehr genaue und dichte Tiefenkarten werden durch iterative Verfahren (siehe Abschnitte 2.4.2.2, 2.4.4) realisiert. Vorteil dieser Verfahren ist ihre nahezu vollständige Tiefenkarte mit Auflösungen bis zu einem 3D-Punkt je Pixel der Originalbilder. Durch diese Verfahren werden alle Objekte, die der Bildsensor erfassen kann, berücksichtigt. Jedoch werden in einigen Verfahren auch 3D-Punkte durch Auffüllen der Fläche bis zum nächsten Sprung ermittelt. Diese Punkte liegen wahrscheinlich in den meisten Fällen an den richtigen Stellen, die Korrektheit dieser Punkte kann jedoch nicht garantiert werden.

Die Tiefenkarten müssen unter Berücksichtigung einer Echtzeitbedingung (siehe Abschnitt 4.1) auf einem eingebetteten System (siehe Kapitel 3) ermittelt werden. Da für iterative Verfahren keine definierte Verarbeitungsdauer angegeben werden kann, sind sie nur bedingt für eine Echtzeitverarbeitung geeignet. Weiterhin werden durch die dichten Tiefenkarten zu viele redundante 3D-Daten ermittelt, sodass nachfolgende Auswertalgorithmen ebenfalls nur mit hohem Aufwand in Echtzeit arbeiten können.

Daher werden in dieser Arbeit blockbasierte Korrelationsmethoden verwendet, mit denen Tiefenkarten aufgabenspezifisch mit einer ausreichenden Dichte ermittelt werden können.

Als Korrelationskriterien werden die SADMF und die NKKFMF favorisiert, da sie sich bei verschiedenen Untersuchungen (siehe [9] und Anhang B) auch im Rahmen dieser Arbeit als vorteilhaft für die Auswertung von Stereobildsequenzen mit variierenden Beleuchtungsbedingungen erwiesen haben.



# Kapitel 3

## Echtzeitfähige Plattformen für die Stereobildauswertung in eingebetteten Systemen

Entscheidend für die Echtzeitfähigkeit einer Implementierung ist neben der Auswahl der Algorithmen vor allem die verwendete Plattform. In Kapitel 2 wurde auf die grundlegenden Algorithmen für die Stereobildverarbeitung eingegangen. Die Notwendigkeit, zwei Bilder gleichzeitig zu verarbeiten, stellt hohe Ansprüche an die verwendete Plattform für eine echtzeitfähige Implementierung. Die Leistungsfähigkeit der Plattform im Kontext dieser Algorithmen wird von verschiedenen Aspekten bestimmt. Bei der Bildverarbeitung stehen der Speicherzugriff, der Parallelisierungsgrad, die Performance und die Möglichkeiten zur Ein-/Ausgabe im Mittelpunkt.

Bei der Echtzeit-Bildverarbeitung ist die Schnittstelle zur Peripherie ein maßgeblicher Faktor, da das Datenaufkommen je nach Bildfrequenz im zweistelligen MByte/s-Bereich liegen kann. Nähere Ausführungen zu den erforderlichen Datenraten in einigen Beispielen findet man im Abschnitt 4.2. In diesem Kapitel wird allgemein auf die zur Verfügung stehenden Plattformen eingegangen.

Im folgenden Abschnitt wird die Einteilung nach datenfluss- und kontrollflussorientierten Plattformen beschrieben. Anschließend werden prozessorbasierte und hardwarebasierte Plattformen in ihren Eigenschaften verglichen. Prozessorbasierte Plattformen arbeiten mit einem zur Laufzeit dynamisch variierbaren Programm, während hardwarebasierte Plattformen vorwiegend starre, aber bezüglich Geschwindigkeit oder Energieeffizienz optimierte digitale Schaltungen sind. Sie werden üblicherweise mit ASICs oder programmierbaren FPGAs realisiert. Abschließend werden verschiedene Konzepte der digitalen Arithmetik mit ihren spezifischen Eigenschaften vorgestellt und bezüglich der Plattform diskutiert.

## 3.1 Datenfluss- und kontrollflussorientierte Plattformen

Die Begriffe Datenfluss und Kontrollfluss wurden während der Entwicklung und Untersuchung von Datenflussmaschinen in den 1980er Jahren geprägt. Der wesentliche Unterschied liegt in der Programmierung. Während bei den kontrollflussorientierten Maschinen die zeitliche Reihenfolge der Verarbeitung vorgegeben wird, werden bei den datenflussorientierten Maschinen die Daten direkt mit den Operationen verknüpft. Ändert sich ein Wert, werden automatisch alle Daten bezüglich der Verknüpfung aktualisiert.

### 3.1.1 Kontrollfluss

Als Kontrollfluss bezeichnet man die sequentielle Ausführung von Befehlen in einer bestimmten Reihenfolge. Die Reihenfolge der Ausführung der Anweisungen bei strenger Kontrollflussorientierung wird ausschließlich von der Position im Programm bestimmt. Eine kontrollflussorientierte Abarbeitung wird mit der "von Neumann"-Architektur assoziiert [95]. Die Darstellung des Kontrollflusses eines Programms kann mit einem Programm-Ablauf-Plan erfolgen. In der Unified Modelling Language (UML) wird der Kontrollfluss mit Hilfe des Aktivitätsdiagramms dargestellt.

Durch die bekannte Abfolge der Befehle können die benötigten Daten bestimmt und zur Verfügung gestellt werden. Dadurch wird die Speicherhierarchie effektiv genutzt, da die jeweils benötigten Teile aus dem nächst größeren, langsameren Speicher nachgeladen werden können. Die Speicherhierarchie und der Einsatz von Caches ist für die kontinuierliche Steigerung der Verarbeitungsgeschwindigkeit von Prozessoren von entscheidender Bedeutung.

Wesentliche Merkmale von kontrollflussorientierten Algorithmen sind Entscheidungen (If-Anweisung) und Schleifen. Plattformen wie z.B. Mikroprozessoren sind auf die Ausführung von kontrollflussorientierten Algorithmen durch einen Programmzähler und die Verarbeitung von Sprungbefehlen optimiert.

Eine Kontrollflussabhängigkeit liegt vor, wenn eine Anweisung durch eine andere Anweisung bedingt ausgeführt wird.

Ein besonderes Beispiel stellen hierbei die Abhängigkeiten innerhalb von Schleifen und die damit verbundenen Parallelisierungsmöglichkeiten dar. Dieses Problem ist nicht trivial und wird in der Loop Dependency Analysis im Compilerbau untersucht. Dadurch kann in Abhängigkeit vom Algorithmus eine Schleife stark parallelisiert werden. In Software sind die Parallelisierungsmöglichkeiten von der Implementierungsplattform abhängig. Die Abhängigkeiten innerhalb von Schleifen stellen somit keine reinen Abhängigkeiten im Sinne des Kontrollflusses dar.

Mit Hilfe des Kontrollflusses kann die Komplexität eines Programms bzw. Algorithmus in Bezug auf seine unterschiedlichen Ausführungspfade bestimmt werden. Hierzu wird z.B. die zyklomatische Komplexität nach McCabe verwendet.



### 3.1.2 Datenfluss

Stehen bei der Verarbeitung die Abhängigkeiten der Daten untereinander im Vordergrund, ist die Verarbeitung datenflussorientiert. Die Untersuchung der Datenflussmaschinen [96] war durch die Möglichkeit der Entwicklung eines datenflussorientierten Verarbeitungskonzeptes für einen Universalrechner motiviert, konnte bisher jedoch nicht realisiert werden. Problematisch war hierbei vor allem die schlechte Prädiktion der benötigten Datenbereiche und somit eine schlechte Ausnutzung der verschiedenen Speicherkonzepte in einer Speicherhierarchie. Eine vollständig assoziative Programmierung bezüglich Daten und Operanden auf universeller Ebene war zu komplex und konnte nur für sehr eingeschränkte Probleme gelöst werden. Dennoch sind verschiedene Ansätze der datenflussorientierten Maschinen, wie z.B. die Superskalarität [80], in die Prozessorenentwicklung eingeflossen. Auch die Aufteilung von Programmen in unabhängige Tasks und Threads, um eine bessere Parallelisierung durchführen zu können, ist durch die Untersuchung datenflussorientierter Rechner initiiert worden.

Durch Analyse der Datenflussabhängigkeiten in Algorithmen können Optimierungen des Algorithmus in Bezug auf die Implementierung durchgeführt werden. Der Datenfluss zeigt die wesentlichen Parallelisierungsmöglichkeiten auf, da hierdurch die unabhängigen Datenpfade ermittelt werden können. Die Eigenschaften von Datenflussabhängigkeiten werden in der Fachliteratur [3] beschrieben.

Durch die Analyse der Datenabhängigkeiten zwischen verschiedenen Anweisungen eines Algorithmus kann ein Datenflussgraph erstellt werden. Er zeigt deutlich die Parallelisierungsmöglichkeiten durch parallele Datenpfade, sogenannte nebenläufige Anweisungen, auf.

Die Parallelisierung von nebenläufigen Anweisungen kann nur durch parallele Strukturen der Implementierungsplattform realisiert werden. Stark nebenläufige Algorithmen werden am besten durch eine Hardwareimplementierung repräsentiert.

## 3.2 Prozessorbasierte Plattformen

Ein Prozessor ist im Allgemeinen die Zentraleinheit eines Datenverarbeitungssystems [60]. Prozessoren unterscheiden sich hinsichtlich ihrer inneren Struktur und damit auch in der Art der Verarbeitung der Daten. Es wurden verschiedene Methoden zur Einteilung von Prozessoren entwickelt. Die Einteilung bezüglich Datenfluss und Kontrollfluss wurde bereits beschrieben und geht über die reine Beschreibung von Prozessoren hinaus. Eine weitere Möglichkeit besteht in der Einteilung der Prozessoren nach Flynn [26] bezüglich der Kontroll- und Datenverarbeitungsstrukturen in:

SISD	single instruction single data - beschreibt die klassischen "von Neumann"-Prozessoren, welche stark kontrollflussorientiert sind;
SIMD	single instruction multi data - es wird ein Kontrollfluss auf mehreren ALUs ausgeführt: hierzu zählen Vektorrechner und auch DSPs. Es wird eine Art von feingranularer Parallelität ermöglicht;
MISD	multi instruction single data - entspricht den Datenflussmaschinen, bei denen auf ein Datenwort mehrere Kontrollflüsse wirken;
MIMD	multi instruction multi data - Beispiele sind Parallelrechner, bei denen mehrere Kontrollflüsse auf verschiedene Datenworte angewendet werden.

Mikroprozessoren für eingebettete Systeme sind oftmals nach der "von Neumann"-Architektur konzipiert und dementsprechend kontrollflussorientierte SISD-Systeme. Je nach Einsatzzweck sind sie für Steuerungsaufgaben in Form von Mikroprozessoren oder Mikrocontrollern oder für konstant hohen Datendurchsatz in Form von DSPs realisiert. Sie sind verhältnismäßig preisgünstig und mit Standard-Programmiersprachen wie C und C++ zu programmieren. Die Verarbeitungsgeschwindigkeiten reichen von wenigen MHz bis zu einem GHz.

Im Folgenden wird auf die beiden größten Gruppen von Prozessoren eingegangen. Im Anschluss werden einige alternative Prozessorkonzepte vorgestellt, die mittlerweile ebenfalls für den Einsatz in eingebetteten Systemen in Betracht kommen.

### 3.2.1 Universalprozessoren

Die Entwicklung der Mikroprozessoren ist stark mit der Entwicklung der Mikroelektronik verbunden. Der Intel 4004 war zusammen mit dem Texas Instruments TMS 1000 und dem Garrett AiResearch's CADC der erste kommerziell vermarktete Mikroprozessor. Der Name Mikroprozessor ist von der Größe gegenüber früheren diskret aufgebauten Prozessoren [18] geprägt. Ein Mikroprozessor umfasst eine ALU, ein Steuerwerk und verschiedenen IO-Ports, die auf einem Chip implementiert sind. Ein Merkmal vieler Mikroprozessoren ist die Mikroprogrammsteuerung. Durch die Mikroprogrammsteuerung kann ein sehr kompakter Prozessor eine Vielzahl von Befehlen ausführen. Frühere Prozessoren bestanden aus einer Vielzahl von Schaltkreisen. Bei den Mikroprozessoren wurden alle wesentlichen Komponenten auf einem Chip vereint. Die Befehle werden hierfür in einem Befehlsdecoder so aufgelöst, dass sie von der ALU verarbeitet werden können. Der Hauptvorteil besteht in der großen Flexibilität, wodurch die Prozessoren für nahezu jedes Problem eingesetzt werden können. Dadurch, dass viele Operationen, wie z.B. die Multiplikationen nicht direkt in Hardware implementiert sind, sondern nur durch die Mikroprogrammsteuerung ausgeführt werden, sind Mikroprozessoren im Vergleich zum Prozessortakt nicht sehr schnell. Die Ausführung erfolgt vorwiegend kontrollflussorientiert.

Während sich auf der einen Seite mit den 8086er Prozessoren die PCs entwickelten, arbeitet

die Mehrzahl der Mikroprozessoren und Mikrocontroller im Verborgenen in sogenannten eingebetteten Systemen. In diesen wurden sie häufig für Steuerungsaufgaben eingesetzt. So ist z.B. die gesamte Rechentechnik in den Space-Shuttles auf Basis von 8086-Schaltkreisen aufgebaut. Mit der Zeit entwickelten sich für eingebettete Mikroprozessoren in Industrie und Haushalt eigene Familien. Am bekanntesten ist die Familie der ARM-Prozessoren, die von verschiedenen Herstellern in einer Vielzahl von Konfigurationen vertrieben wird. Ebenso werden oft die “veralteten” PC-Prozessoren für eingebettete Systeme eingesetzt.

In den 8086-Prozessoren wurde erstmals eine Form des Pipelinings in Mikroprozessoren getestet. Beim Pipelining wird die Ausführung eines Befehls in zeitlich annähernd gleich lange Teile (Pipelinstufen) zerlegt. Durch eine Trennung der Pipelinstufen können die unterschiedlichen Aufgaben so für mehrere Befehle parallel durchgeführt werden. Die anfänglichen Probleme des Pipelinings des 8086 waren im 80286 behoben. Er wird noch in vielen eingebetteten Systemen eingesetzt. Seine Produktion wurde erst im Jahr 2007 eingestellt. Weitere Beispiele für 8-Bit-Prozessoren, die vorwiegend in eingebetteten Systemen verbreitet sind, sind der Zilog Z80 und Motorolas 6800.

Durch die Erforschung der Datenflussmaschine in den 1980er und 1990er Jahren ergaben sich wesentliche Neuerungen [17], wie zum Beispiel die Superskalarität, Multitasking und Multithreading für die Mikroprozessoren. Bei der Superskalarität ist in dem Prozessor mehr als eine Befehlspipeline vorhanden, dadurch können mehrere unabhängige Befehle gleichzeitig ausgeführt werden. Der Pentium von Intel war der erste Prozessor mit dieser Funktionalität. Weiterhin war durch die Verfügbarkeit von zwei Befehlsdecodern auch Multithreading möglich.

Durch eine höhere Anzahl der parallelen Befehlspipelines setzte sich AMDs Athlon gegen Intels Pipeliningstruktur mit einer geringeren Anzahl von deutlich längeren Pipelines im Pentium 4 durch. Beide Systeme waren jedoch im Gegensatz zu den früheren Mikroprozessoren sehr unwirtschaftlich bezüglich der Leistungsaufnahme und somit auch bezüglich der Wärmeabgabe. Aus diesem Grund fanden sie nie den Weg in eingebettete Systeme.

Aktuelle Prozessoren sind oftmals direkt als Mehrkernprozessoren mit 64 Bit Datenbus ausgelegt. Die einzelnen Prozessorkerne besitzen jeweils mehrere Befehlspipelines. Die ersten Systeme wurden 2005 angekündigt und sind mittlerweile sehr weit verbreitet. Durch neue Technologien sind die Leckströme minimiert, so der Leistungsverbrauch stark gesenkt und damit Stromsparausführungen (vorwiegend für den Einsatz in Notebooks) entwickelt worden. Diese sind prinzipiell auch für rechenintensive Aufgaben in eingebetteten Systemen einsetzbar. Erste Entwicklungsboards in dieser Richtung sind seit 2007 verfügbar. Da jedoch bei eingebetteten Systemen häufig Leistungsaufnahme und Wärmeabgabe eine entscheidende Rolle spielen, wird selbst der Einsatz von energieeffizienten PC-Prozessoren in eingebetteten Systemen nur sehr begrenzt erfolgen.

Im Gegensatz zu einem PC-System, welches als offenes System entworfen wurde und eingesetzt wird, ist die Peripherie bei eingebetteten Systemen im Allgemeinen direkt mit dem Prozessor verbunden. Dadurch kann die speziell eingesetzte Hardwarekonfiguration besser getestet und ein stabileres System erreicht werden.

Aktuelle Mikroprozessoren verfügen über IO-Schnittstellen mit großer Bandbreite, benötigen jedoch noch weitere Schaltkreise zur effektiven Verwaltung und Verteilung der Bandbreite auf verschiedene Peripheriebausteine. Durch diese Schnittstellen stellen Bildaufnahme und Verarbeitung mittlerweile ein geringeres Problem dar. Die überwiegend in eingebetteten Systemen eingesetzten Mikroprozessoren verfügen noch über Schnittstellen mit sehr geringer Bandbreite. Dies erschwert die Zuführung von großen Datenmengen, z.B. Bildern.

Mikroprozessoren werden überwiegend für Steuerungs- und Automatisierungsaufgaben eingesetzt. Für rechenintensive Aufgaben, wie z.B. die Stereobildverarbeitung werden in eingebetteten Systemen vorwiegend digitale Signalprozessoren verwendet.

### 3.2.2 Digitale Signalprozessoren

Bei digitalen Signalprozessoren steht eine datenflussorientierte Verarbeitung stärker im Vordergrund als bei Mikroprozessoren. Hierfür werden verschiedene Operationen, wie z.B. die Multiplikation oder die MAC-Operation direkt in Hardware implementiert und können so innerhalb von einem Takt ausgeführt werden. Die vorherrschende Rechnerarchitektur ist die Harvardstruktur, bei der getrennte Busse und Speicher für Befehle und Daten vorhanden sind. Oftmals werden mehrere Datenbusse und -speicher implementiert, wodurch eine flexible Verwaltung ermöglicht wird.

Die Verarbeitung der Daten erfolgt trotz stärkerer Orientierung auf den Datenfluss auch bei digitalen Signalprozessoren kontrollflussorientiert, d.h. es gibt ein Programm, welches zu jedem Zeitpunkt festlegt, welcher Befehl ausgeführt wird. Die Ausführung der Befehle ist jedoch weitestgehend in speziellen Strukturen parallelisiert. So erfolgt das Inkrementieren von Schleifenvariablen und von Adressen bei Ringpuffern oder auch die Bit-reverse Adressierung (wird für die Fast Fourier Transformation - FFT benötigt) in speziell dafür angelegten Hardwarestrukturen. Weiterhin sind mehrere Data-Address-Generation-Units (Adressrechenwerke) vorhanden. Diese Parallelisierung führt nur dann zu einer Beschleunigung, wenn das Programm die Möglichkeiten des Prozessors gut ausnutzt [47]. Digitale Signalprozessoren (DSP) sind besser zur schnellen Signalverarbeitung und somit auch zur Bildverarbeitung geeignet als universelle Mikroprozessoren, wenn regelmäßig wiederkehrende Befehlsfolgen schnell abgearbeitet werden müssen. Ein DSP führt viele Operationen mit einem Takt aus, daher bleibt die Taktrate niedrig und die Leistungsaufnahme gering. DSPs werden überwiegend in eingebetteten Systemen verwendet.

Ein weiteres wichtiges Unterscheidungsmerkmal von DSPs ist die Einteilung nach der verwendeten Arithmetik in Festkomma- und Gleitkomma-DSPs. In beiden Fällen sind die ALUs im Gegensatz zu den Mikroprozessoren fest verdrahtet. Da bei Gleitkommaberechnungen die Genauigkeit von der Bitbreite der Mantisse abhängt, werden der Gleitkommaaddierer mit sehr großer Bitbreite ausgelegt, um beide Mantissen auf den gleichen Exponenten umzuwandeln. Diese Rechenwerke unterscheiden sich in der Ausführungsgeschwindigkeit nur unwesentlich von ihren Festkommapendants, jedoch sind sie bedingt durch den komplexeren Aufbau in der Regel deutlich teurer [18].

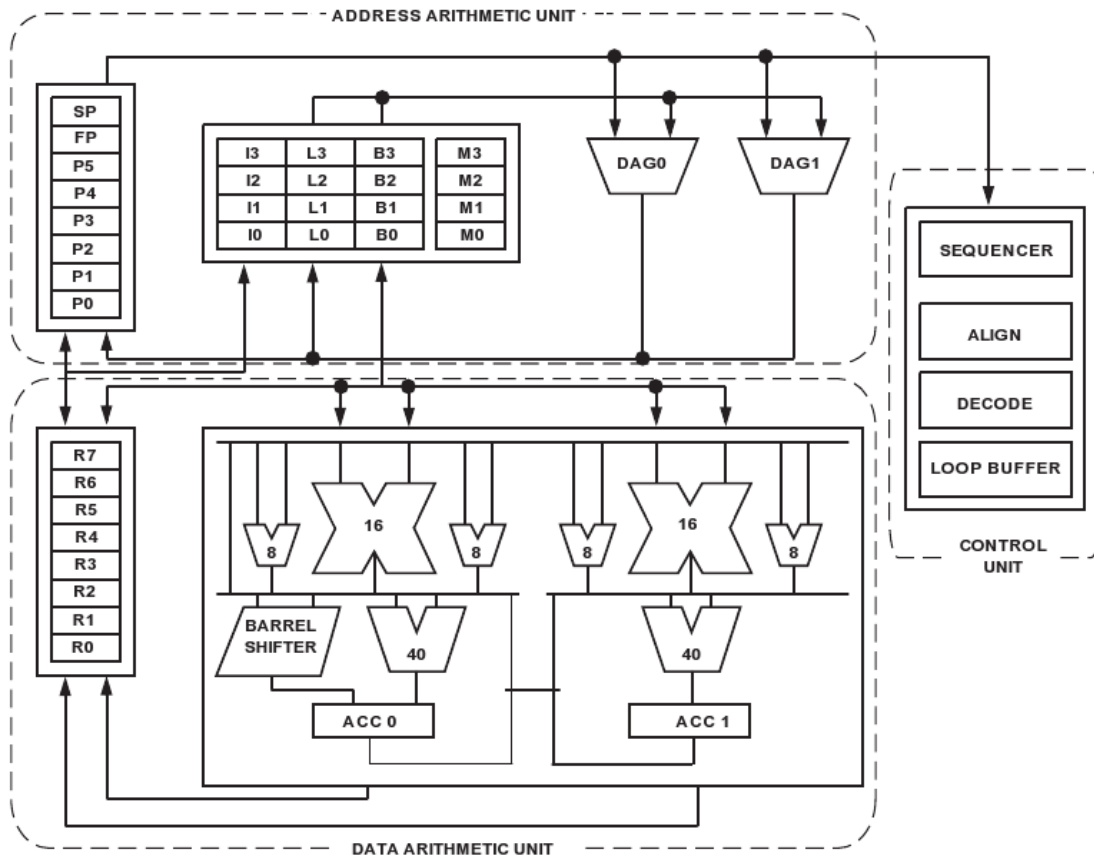


Abbildung 3.1: Blockschaltbild der ALU des Festkomma-DSPs BF533 [8]

Aufgrund ihrer Struktur lassen sich insbesondere Filter effektiv mit DSPs realisieren. Für die Stereobildverarbeitung sind die meisten DSPs nur bedingt tauglich, da sie nur eine geringe Anzahl von Eingabeports mit zu niedriger Bandbreite besitzen. Viele DSPs sind vorwiegend für eindimensionale Datenströme ausgelegt. Einzelne Modelle, wie zum Beispiel der MSC8144 [28] von Freescale oder die C64-Serien [84] von Texas Instruments für Festkomma-DSP, sowie der TigerSHARK TS201S [7] von Analog Devices sind hingegen aufgrund ihrer Rechenleistung für eine Stereobildverarbeitung geeignet, haben jedoch auch einen höheren Stromverbrauch.

In aktuellen DSPs sind neben den Konzepten zur Signalverarbeitung auch Komponenten integriert, die von Mikrocontrollern und universellen Mikroprozessoren stammen. Damit können auf DSPs Betriebssysteme ausgeführt werden. Ein Beispiel hierfür ist die Blackfin-Serie mit dem BF533 [8] von Analog-Devices. Die Blackfin Serie beinhaltet 32 Bit-DSPs mit zwei unabhängigen Rechenwerken mit jeweils einer 16 Bit MAC Operation. Die schnellsten Prozessoren dieser Serie laufen mit 600 MHz.

In Abbildung 3.1 ist die ALU des DSPs Blackfin BF533 von Analog Devices dargestellt. Sie verdeutlicht gut die Parallelisierungsmöglichkeiten innerhalb von DSPs. Es sind zwei Multiplizierer mit nachgeschaltetem Addierer vorhanden, so dass zwei MAC-Operationen gleichzeitig ausgeführt werden können. Alternativ dazu können vier 8 Bit-Operationen statt der Multipli-

kationen durchgeführt werden. Die korrekte Speicherung der Eingangsdaten vorausgesetzt ist somit die Berechnung von 2 SAD-Blockkombinationen gleichzeitig möglich.

Auf der anderen Seite entwickeln sich Mikroprozessoren durch Erweiterungen auch zu DSPs. Die Tricore-Serie von Infineon ist hierfür ein Beispiel. Dabei existieren in diesem Mikroprozessor drei vollständige Pipelines, wobei in einer eine 16 Bit MAC-Operation durchgeführt werden kann. Der Multiplizierer und der Addierer sind in diesem Fall fest verdrahtet.

### 3.2.3 Alternative Prozessorsysteme

Neben diesen beiden Arten von Prozessoren wurden in der letzten Zeit verschiedene andere Prozessoren entwickelt, die unterschiedliche Prozessorkonzepte verwirklichen und für den Einsatz in eingebetteten Systemen konzipiert sind. Nahezu alle dieser Prozessoren sind parallel in einer SIMD oder MIMD Struktur organisiert.

#### 3.2.3.1 CELL-Prozessor

Der Cell-Prozessor ist ein kleiner SIMD-Prozessor, der von Sony, Toshiba und IBM entwickelt wurde. Der Prozessor besteht aus einem Power Processing Element (PPE), 8 Synergistic Processing Elementen (SPE), einem Circular Data Bus und einer IO-Schnittstelle. [70]

Die PPE ist ein vollwertiger RISC-Prozessor, welcher die SPEs steuert und kann auf den Speicher der SPEs direkt zugreifen. Nur die PPE hat Zugriff auf die IO-Schnittstelle. Auf der PPE kann herkömmliche Software, wie z.B. ein Betriebssystem, laufen.

Durch die SPEs ergibt sich die hauptsächliche Rechenleistung. Sie können entweder jeweils ein Programm für sich ausführen und so als eine Art von MIMD-Prozessor fungieren, oder aber in einer Art SIMD-Struktur zusammenarbeiten. Dadurch ergibt sich ein großes Potential für Parallelisierungen. Problematisch hierbei ist, dass Programme nicht einfach von anderen Rechnern übernommen werden können, sondern aufgrund der SIMD/MIMD-Struktur direkt für den CELL-Prozessor geschrieben werden müssen. Die gängigen Programme unterstützen nicht die Möglichkeiten des CELL-Prozessors und würden nur auf dem PPE ausgeführt werden, wodurch der Großteil der Rechenleistung ungenutzt bliebe.

Durch den Einsatz als SIMD-Prozessor kann der CELL-Prozessor Bildverarbeitungsalgorithmen stark parallelisieren. Es bietet sich die Möglichkeit eines kompakten Bildverarbeitungsprozessors, der auch die Aufgaben der Datenorganisation übernehmen kann. Ein Engpass bezüglich der IO-Schnittstelle und das Blockieren des Prozessors durch Kopieroperationen können somit umgangen werden.

Der wesentliche Einsatz zum gegenwärtigen Zeitpunkt ist die Sony Playstation 3. An der University of Tennessee wird ein Cluster-Rechner betrieben, der aus vielen Playstation 3 besteht. Seit März 2008 bietet IBM Blade-Server auf Basis des Cell-Prozessors an [82].



### 3.2.3.2 Grafik und Physik-Prozessoren (SIMD)

Während vor einigen Jahren die Grafikprozessoren (GPU; Graphic Processing Unit) noch sehr schwer programmierbar waren, wurde durch die Einführung des sogenannten Stream Processing in Grafikprozessoren eine Art Universal GPU entwickelt. Sie stellen eine spezielle Form eines Vektorprozessors (SIMD) dar. Hierbei werden sowohl Integer als auch Floating Point Rechenwerke verwendet [60].

Beim Stream Processing handelt es sich um eine Form von eingeschränkter Parallelverarbeitung. Durch das Einschränken der Verarbeitungsmöglichkeiten kann die Parallelisierung des Programmablaufes weitestgehend automatisiert werden. Dadurch wird die SIMD-Programmierung wesentlich vereinfacht. Dabei können durch das Ermitteln von Datenabhängigkeiten auch DMA-Prozesse angestoßen werden.

In Grafik-Prozessoren arbeiten mehrere Pixel-Shader-Pipelines, die als Gleitkomma- und als Festkommapipelines verwendbar sind. Durch die feste Verdrahtung der Pixel-Shader-Pipelines wird ein hoher Datendurchsatz erreicht.

Einen ähnlichen Ansatz stellen die Physik-Prozessoren (Physic Processing Unit) [76] dar. Beim PhysX-Schaltkreis der Firma AGEIA kontrolliert ein RISC-Prozessor ein Feld von Gleitkomma-SIMD-Prozessoren. Die Stromaufnahme dieser Prozessoren ist zum gegenwärtigen Zeitpunkt wesentlich zu hoch, um einen Einsatz in eingebetteten Systemen zu verfolgen. Da diese Prozessoren zum gegenwärtigen Zeitpunkt zur Verwendung als Co-Prozessor ausgerichtet sind, sind sie für eingebettete Systeme nur bedingt geeignet.

### 3.2.3.3 Massiv Parallele MIMD-Prozessoren

Die Firma Rapport bietet einen Schaltkreis mit 256 8-Bit-Prozessoren an, die mit einer Geschwindigkeit von bis zu 125 MHz arbeiten. Damit kann ein theoretischer Datendurchsatz von 32 Gigabyte je Sekunde erreicht werden, wobei der Schaltkreis mit einer Leistungsaufnahmen von nur 500 mW angekündigt wurde [71].

Durch die parallele Struktur eignet sich dieser Chip sehr gut für Bildverarbeitungsaufgaben. Jedoch ist die Bitbreite der einzelnen Prozessoren verhältnismäßig gering. Bei aufwändigen Berechnungen, wie sie für die Stereobildverarbeitung erforderlich sind, könnte sich das negativ auswirken. Weiterhin kann die Synchronisierung der Datenströme durch die Verwendung von Software sehr aufwändig werden (siehe Abschnitt 3.1.2).

Im Jahr 2006 stellte die Firma Rapport einen Schaltkreis mit 1024 8-Bit-Prozessoren und einem 32-Bit-PowerPC-Prozessor vor. Aufgrund der geringen Leistungsaufnahme stellt dieser Ansatz trotzdem eine gute Möglichkeit zur Parallelisierung von Programmen und zur Entwicklung von datenflussorientierten Abläufen dar. Die Bitbreite ist aber für die Stereobildverarbeitung zu stark begrenzt.

Auch andere Firmen orientieren in diese Richtung, so hat die z.B.Firma NEC einen Schaltkreis

mit 20 32-Bit-Prozessoren vorgestellt, dieser soll aber vorwiegend für leistungsfähigere Netzwerkknoten eingesetzt werden und ist nicht so energieeffizient. Die deutlich höhere Bitbreite ist besser für die Algorithmen zur Stereobildauswertung geeignet.

### 3.3 Hardwarebasierte Plattformen

Bei hardwarebasierten Plattformen entsteht während des Design-Prozesses eine Struktur, die die Funktionalität abbildet. Diese Struktur ist vollständig parallel. Alle Prozessoren sind auf Hardwareplattformen implementiert. In der vorliegenden Arbeit sollen nur Plattformen für komplexe Designs mit Hilfe der Mikroelektronik betrachtet werden.

Es gibt sehr unterschiedliche Ausrichtungen von Hardwareplattformen, zum einen die anwendungsspezifischen Schaltkreise, die fest verdrahtete Elektronik enthalten und für einen speziellen Einsatzzweck entwickelt werden, auf der anderen Seite stehen die programmierbaren Logik-Schaltkreise in verschiedenen Variationen. Sie sind in der Lage, je nach Programmierung unterschiedliche Funktionen durchzuführen, und können in Abhängigkeit von der verwendeten Technologie unterschiedlich oft reprogrammiert werden.

#### 3.3.1 Anwendungsspezifische Schaltkreise

Die anwendungsspezifischen Schaltkreise (ASIC; engl.: Application specific integrated Circuit) können sehr flexibel aufgebaut werden, da das Design vor dem Silizium-Prozess entworfen wird. So kann ein ASIC aus Standardzellen, einem Gate-Array-Teil, verschiedenen vorentwickelten Anteilen aus Bibliotheken oder Intellectual Property<sup>1</sup> (IP) und natürlich einem vollkommen kundenspezifischen Anteil bestehen. Je nach Anteil der unterschiedlichen Teile variieren Kosten und Entwicklungszeit. Das Ergebnis dieses Designprozesses wird auch *System on a Chip (SoC)* genannt.

**Vollkundendesign** Beim Vollkundendesign werden ausgehend von einer Schaltung die elektronischen Bauelemente und ihre Verbindungsstruktur entwickelt. Dadurch kann in vielen Fällen die benötigte Fläche stark reduziert oder spezielle Eigenschaften für die Verarbeitung von analogen Signalen erreicht werden.

Durch das freie Design muss der Schaltkreis in jeder Entwicklungsphase auch vollständig verifiziert werden, wodurch eine lange Entwicklungszeit und sehr hohe Entwicklungskosten entstehen. Im Allgemeinen wird daher versucht, den Anteil des Vollkundendesigns möglichst gering zu halten.

---

<sup>1</sup>Intellectual Property beschreibt Geistiges Eigentum im Allgemeinen. Es kann z.B. durch Verfahren, Quellcode, Texte oder Bilder repräsentiert werden. Ausführungsformen in Hardware werden häufig als IP-Cores bezeichnet.



**Standardzelle** Um die Entwicklungskosten zu reduzieren, wurden Standardzellen entwickelt. Standardzellen sind Bibliotheken von vorentwickelten und getesteten Schaltungen für verschiedene Funktionalitäten. Die Komplexität der einzelnen Standardzellen ist sehr unterschiedlich. Durch die Verwendung von Standardzellen wird der Testaufwand für das Design wesentlich reduziert. Er liegt im Wesentlichen bei den Verbindungen und dem Zeitverhalten der Standardzellen zueinander.

**Gate Array Design** Das Gate Array Design stellt eine weitere Standardisierung, insbesondere für die digitale Datenverarbeitung, dar. Es kommen Masken mit einem fest definierten Array aus logischen Gates zum Einsatz, bei denen der einzige Freiheitsgrad in den Verbindungen mittels der Metallebenen besteht. Die Weiterentwicklung dieser Strategie führte durch die Kombination mit Programmable Logik Arrays in Richtung der Field Programmable Gate Arrays (FPGA), die in Abschnitt 3.3.2 näher beschrieben werden.

Wiederum wird der Testaufwand wesentlich reduziert, da das Gate-Array hinsichtlich des Zeitverhaltens in sich abgestimmt und getestet ist.

### 3.3.1.1 Design-Prozess

Im Design-Prozess werden über die Schaltung Masken für den Silizium-Prozess erstellt. Dieser Schritt ist sehr aufwändig, da intensive Tests und Simulationen durchgeführt werden müssen, um das Design zu verifizieren. Die eigentliche Produktion ist nach erfolgreichem Design im Vergleich zu anderen Lösungen sehr preiswert. Daher sind anwendungsspezifische Schaltkreise besonders bei großen Stückzahlen vorteilhaft. Ist eine Funktionalität aufgrund der Anforderungen, z.B. sehr hoher Datendurchsatz, nur mittels eines ASIC-Designs realisierbar, kann dieser Weg auch bei kleineren Stückzahlen effektiv sein.

Bei der Erstellung eines Designs wird, ausgehend von den Systemanforderungen, eine Beschreibung mit Hilfe grafischer Eingabemethoden oder einer Hardwarebeschreibungssprache (HDL; engl. Hardware Description Language; siehe Abschnitt 3.3.2.2) vorgenommen. Diese Beschreibung wird getestet und optimiert, bis die Anforderungen erfüllt werden. Anschließend wird aus der Beschreibung eine Netzliste erstellt und diese mit Hilfe der in Abschnitt 3.3.1 beschriebenen Möglichkeiten abgebildet. Nun sind die zeitlichen Abhängigkeiten bekannt und das Design muss erneut simuliert und getestet werden.

### 3.3.1.2 Vor- und Nachteile der Entwicklung von ASICs

Mit Hilfe von ASICs können sowohl primär datenflussorientierte Konzepte als auch mit Hilfe von Prozessor-IP-Cores, wie z.B. der Familie der ARM-Prozessoren, vorwiegend kontrollflussorientierte Konzepte umgesetzt werden. Dies hängt vollständig von den Anforderungen ab. In ASICs können sowohl analoge als auch digitale Verarbeitungsstränge realisiert werden.

Die wesentlichen Vorteile einer Implementierung in einen ASIC stellt die prinzipiell sehr hohe

erreichbare Geschwindigkeit bei gleichzeitig niedriger Leistungsaufnahme und Abwärme dar. Diesen Vorteilen steht ein langwieriger und kostenintensiver Entwicklungsprozess gegenüber. Insbesondere in der Erforschung neuer Hardware-Designs erscheint die Entwicklung mit Hilfe von ASICs nicht als sinnvoll, da für das Redesign der gesamte Design-Ablauf erneut durchgeführt werden muss. Dies erhöht den Aufwand beim Design von ASICs erheblich.

### 3.3.2 Programmierbare Logikbausteine

Ein FPGA (Field Programmable Gate Array) ist ein freiprogrammierbarer Logikschaltkreis. Die FPGAs sind eine Weiterentwicklung der Programmable Logic Arrays (PLAs) mit dem Ziel, deutlich komplexere Schaltungen zu ermöglichen. In PLAs kann mit Hilfe einer UND-ODER-Matrix eine beliebig komplexe rückkopplungsfreie logische Funktion abgebildet werden. Rückkopplungen wurden zu Beginn vor allem mit Hilfe äußerer Beschaltungen realisiert.

PROMs<sup>2</sup> und PALs<sup>3</sup> sind Variationen von PLAs mit etwas anderen Eigenschaften. Die PALs hatten nur eine programmierbare Und-Masken-Ebene. Die Oder-Maske ist fest verdrahtet. Bei den Weiterentwicklungen der PALs zu den GALs<sup>4</sup> wurden freie Ein-Ausgabe-Blöcke verwendet, in denen sich auch Register befinden. Dadurch können Ausgangsdaten auch ohne äußere Beschaltung wieder als Eingang verwendet werden. Dafür geht jedoch eine Ein-Ausgangszelle verloren [10].

1985/86 wurden von zwei Firmen Alternativen präsentiert. Die Firma Altera entwickelte die CPLDs (Complex Programmable Logic Device) und die Firma Xilinx die FPGAs. Bei den CPLDs werden mehrere PALs durch interne Verbindungen verknüpft. Dadurch ist es möglich, komplexe Designs mit Rückführungen zu implementieren. Unterstützt wird dies durch die Verfügbarkeit von Registern in den Makrozellen. CPLDs enthalten nur wenige PALs, stellen jedoch intern eine Vielzahl an Verschaltungsmöglichkeiten mittels der Und-Oder-Matrix bereit.

Die von Xilinx eingeführten FPGAs sind sehr viel feingliedriger. Hierbei werden viele kleine Logikzellen, die zwei logische Funktionen mit vier Eingängen und verschiedene Kombinationen abbilden können, über ein komplexes Netz von Leitungen verbunden.

Beide Ansätze haben Vor- und Nachteile. In FPGAs kann es aufgrund der Struktur dazu kommen, dass Logikzellen nicht verwendet werden können, da sie keine Verbindungen mehr zur Umgebung schalten können. In CPLDs stellt eher die Anzahl der verfügbaren Register ein Problem dar.

Mittlerweile bieten beide Firmen sowohl CPLDs als auch FPGAs verschiedener Größen an. Marktführer ist Xilinx direkt gefolgt von Altera. Weitere Anbieter sind Lattice, Actel, Cypress, Quick Logic und Atmel (in der Reihenfolge der Marktanteile [22]).

Inzwischen enthalten FPGAs teilweise festimplementierte Prozessoren (z.B. ARM9 oder Po-

---

<sup>2</sup>PROM - Programmable Read Only Memory

<sup>3</sup>PAL - Programmable Array Logic

<sup>4</sup>GAL - Gate Array Logic

werPC) bzw. DSP-Blöcke. DSP-Blöcke bestehen aus schnellen Multiplizierern mit fester Bitbreite und ermöglichen zusammen mit der umgebenden Logik die Erstellung von MAC-Operationen (siehe 3.2.2), sie können aber auch direkt als schnelle Multiplizierer eingesetzt werden. Weiterhin erfolgt eine stete Steigerung der IO-Fähigkeiten von FPGAs. Die IO-Zellen können eine Vielzahl unterschiedlicher Spannungsbereiche abdecken und sind teilweise auch als gepaarte Verbindungen für LVDS-Signale für bis zu mehreren 100 MHz geeignet.

### 3.3.2.1 Aufbau von FPGA

FPGAs bestehen aus einer regelmäßigen Struktur von Logikblöcken oder Logikzellen. Diese sind durch programmierbare Verbindungen miteinander verknüpft.

In Abbildung 3.2 ist die Struktur eines Logikelements eines FPGA der Stratix-Familie von Altera dargestellt. Es hat vier Eingänge und vier Ausgänge sowie zusätzlich diverse Steuereingänge und -ausgänge.

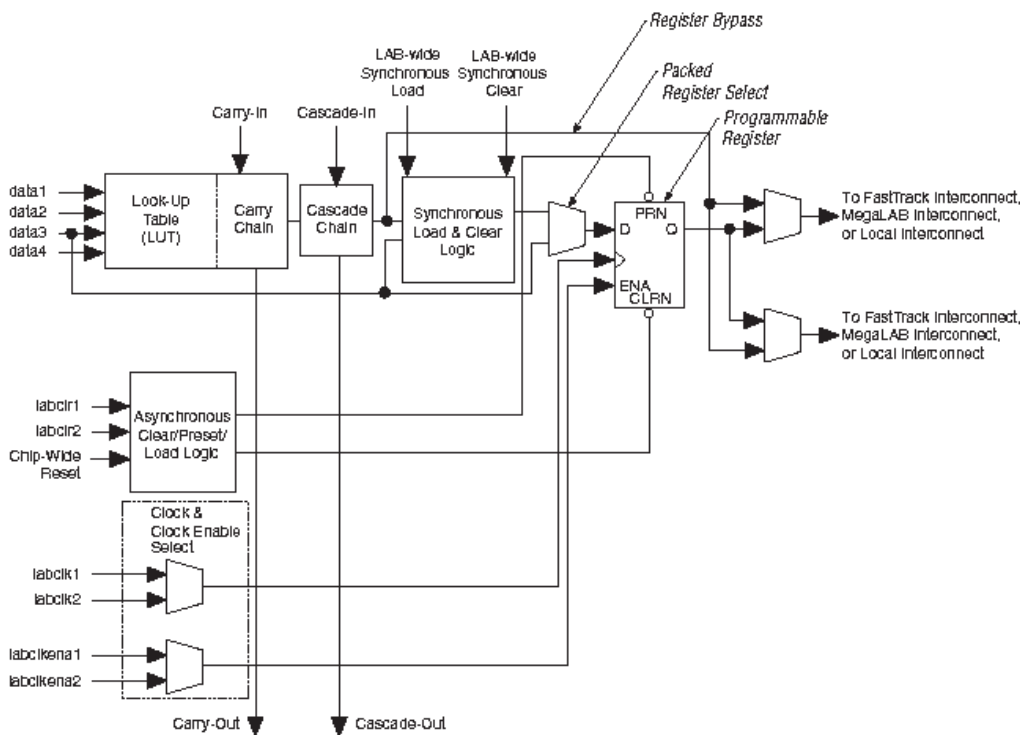


Abbildung 3.2: Logikelement eines FPGA; APEX20K-Reihe von Altera [22]

Die eigentliche Logik ist durch eine Look Up Table (LUT) realisiert. Eine LUT wird im Allgemeinen mit Hilfe von Speicherzellen und Multiplexern erzeugt. Daher kann garantiert werden, dass alle logischen Funktionen, die in der Logikzelle implementierbar sind, mit einer konstanten Verzögerung abgearbeitet werden können.

Zusätzlich sind die Zellen mit Funktionen ausgerüstet, die die Implementierung von Zählern und Arithmetik einfacher und effektiver machen. So ist zum Beispiel eine für Addierer akti-

vierbare Carry- Chain für benachbarte Elemente implementiert. Für Zähler sind die Eingänge Synchronous Load und Synchronous Clear vorhanden. Am Ausgang der Logikzelle befindet sich ein Register, mit dem unter anderem das Pipelining von Arithmetik realisiert werden kann.

Die Bitbreite der Logikblöcke ist sehr gering. Durch die Zusammenschaltung mehrerer Logikblöcke kann die Bitbreite nach Bedarf nahezu beliebig erhöht werden.

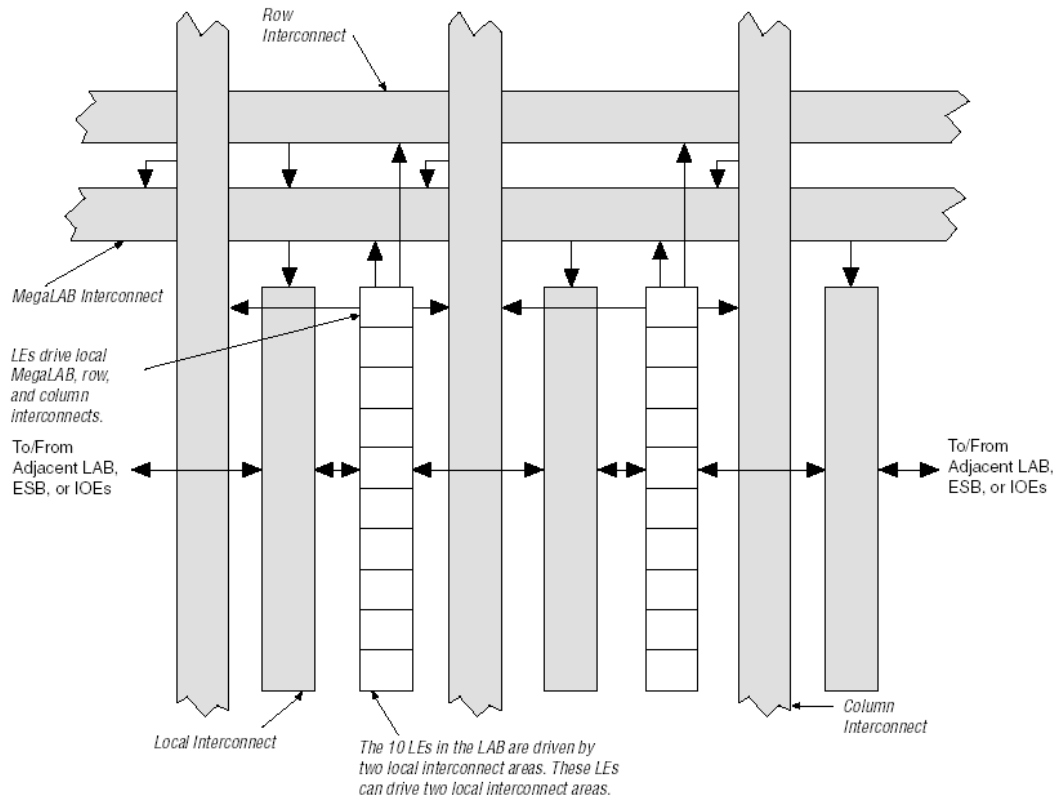


Abbildung 3.3: LAB-Struktur der APEX20K-Reihe von Altera [22]

Aus Abbildung 3.3 ist die innere Struktur eines Stratix-FPGA ersichtlich. In den Logic Array Blocks (LAB) sind jeweils zehn Logikelemente zusammengeschaltet. Zwischen den Reihen der LABs liegen DSP-Blöcke, RAM-Blöcke verschiedener Größe, die Verbindungsnetze und die Taktnetze.

Die Zusammenfassung von 10 Logikelementen zu einem LAB ermöglicht geschwindigkeitsoptimierte Verbindungen für Carry-Chains oder Zählerketten. Auch die Ergebnisse der LUTs können direkt ohne das globale Verbindungsnetzwerk zu belasten, verschaltet werden. Es entstehen schnellere und kompaktere Designs.

Jeder DSP-Block hat die Größe von acht LABs. In ihm befinden sich acht 9x9 Bit Multiplizierer, die auch zu vier 18x18 Bit Multiplizierern zusammen geschaltet werden können. Unter Verwendung lokal angrenzender Logikelemente können auch Multiplizierer größerer Bitbreite abgebildet werden. Der Name DSP-Block bezieht sich jedoch auf die Funktionalität der MAC-Operation, welche mit diesen Blöcken bezüglich Platz und Geschwindigkeit sehr effizient

implementiert werden kann. Durch diese Blöcke ist die Rechenleistung stark angestiegen.

Die aktuelle Konfiguration muss im FPGA gespeichert werden. Dies kann in Antifuses, EEPROMs und SRAMs im FPGA gespeichert werden.

Bei den **Antifuses** werden die Informationen dadurch permanent gespeichert, dass eine nicht-leitende Verbindung durch Anlegen einer äußeren Spannung leitend wird. Die Antifuses haben den Nachteil, dass das entsprechende Device nur einmal programmierbar ist. Dafür ist die Herstellung billiger und die Strukturen sind kleiner als bei EEPROM oder SRAM-Zellen.

Die **EEPROMS** (Electrical Erasable Programmable Read Only Memory) haben den Vorteil, dass sie ihre Information permanent speichern, aber dennoch löscher und reprogrammierbar sind. Nachteil gegenüber SRAM-Zellen ist die in der Anzahl begrenzte Reprogrammierbarkeit, welche üblicherweise bei bis zu 10.000 Programmiervorgängen liegt. Ein weiterer entscheidender Nachteil der Speicherung in EEPROMs ist, dass sie im Gegensatz zu den FPGAs nicht mit einem Standard-CMOS-Prozess hergestellt werden können und somit die Herstellung sehr aufwändig ist.

In den meisten aktuellen FPGA wird deshalb die **SRAM-Technologie** verwendet, da sie den Vorteil hat, dass das FPGA unbegrenzt oft programmiert werden kann. Einen weiteren Vorteil stellt die partielle oder vollständige Rekonfiguration im laufenden Betrieb dar. Damit wäre z.B. eine Umschaltung zwischen Tag- und Nacht-Betrieb für eine Hindernisdetektion im gleichen Schaltkreis im laufenden Betrieb möglich. Die SRAMs werden oft in Verbindung mit externen EEPROMs oder FlashRAMs eingesetzt, in denen die Konfigurationen dauerhaft gespeichert werden. Durch spezielle Betriebsmodi des FPGA können die Konfigurationen selbständig bei Zuführung der Betriebsspannung aus dem dauerhaften Speicher in das FPGA hinuntergeladen werden. So können die relativ teuren FPGAs sehr häufig programmiert werden, während die preiswerten EEPROMs und FlashRAMs am Ende ihrer Lebensdauer unabhängig vom FPGA ausgetauscht werden müssen.

### 3.3.2.2 Entwicklungsumgebung

Der Entwicklungsprozess eines FPGA-Designs besteht aus mehreren Schritten, bei dem jeweils mehrere Programme und Hilfsmittel erforderlich sind.

- Beschreibung der Funktion
- Überführung der Beschreibung in eine Netzliste
- funktionale Simulation
- Fitting und Routing -Process
- Assembler
- Timing-Simulation

Die Entwicklungsumgebung für das FPGA-Design enthält Werkzeuge zur Beschreibung der Funktion mittels grafischer Beschreibung oder Hardwarebeschreibungssprache, Überführung und Simulation einer Netzliste mit Hilfe eines Synthese-Tools sowie einen oder mehrerer Simulatoren. Weiterhin sind Werkzeuge für den Fitting und Routing-Process, mit deren Hilfe die Netzliste auf die Hardwarestruktur abgebildet wird, enthalten. Für die Verifikation werden Programme benötigt, die das Design in einer Simulation und letztendlich im Schaltkreis auf korrekte Funktion überprüfen können.

Im Folgenden wird zunächst auf Hardwarebeschreibungssprachen eingegangen und anschließend der Design-Prozess beschrieben.

**Beschreibungssprachen** Die Beschreibung von FPGA-Designs kann sowohl grafisch als auch mit Hilfe von Hardwarebeschreibungssprachen erfolgen. Ein wesentliches Merkmal dieser Beschreibungsmöglichkeiten ist die Darstellung von nebenläufigen Prozessen. Dies ist in gängigen Programmiersprachen, wie C/C++, Pascal u.ä. nicht vorgesehen. Grafische Beschreibungen mit Hilfe von Schaltungseingabeprogrammen haben den Nachteil, dass sie im Regelfall nicht standardisiert und damit an die spezielle Software eines Herstellers gebunden sind. Auf grafische Beschreibungsmöglichkeiten wird daher nicht weiter eingegangen.

Gängige Hardwarebeschreibungssprachen sind

- **A**dvanced **B**oolean **E**xpress **L**anguage (ABEL)
- **A**ltera **H**ardware **D**escription **L**anguage (AHDL)
- **V**erifying **L**ogic (Verilog)
- **V**HSIC **H**ardware **D**escription **L**anguage (VHDL)

ABEL ist eine Sprache zur Beschreibung von einfachen Designs für kleine FPGAs oder CPLDs. Die Funktion wird mit der Booleschen Algebra beschrieben. Es besteht die Möglichkeit, einfach Blöcke zu bilden. AHDL ist eine direkte Weiterentwicklung von ABEL mit besseren Möglichkeiten für hierarchische Designs.

VeriLog wurde im Jahr 1983 vorwiegend für die Simulation und Verifikation von elektronischen Schaltungen und Schaltkreisen entwickelt. Es wurde im Jahr 1995 vom IEEE im Standard 1364-1995 standardisiert und im Jahr 2001 zum Standard 1364-2001 erweitert. Mittlerweile ist Verilog neben VHDL eine der beiden dominierenden Hardwarebeschreibungssprachen.

Mit VHDL ist die Beschreibung, Simulation und Synthese von komplexen elektronischen Schaltungen und Schaltkreisen bis hin zu Mikroprozessoren möglich. VHDL bietet die Möglichkeiten, die detaillierten Informationen für den Synthese-Prozess hinzuzufügen. Ebenso kann die vollständige Simulationsumgebung mittels VHDL beschrieben und dem Design beigelegt werden. Dadurch kann zu jedem Zeitpunkt die Funktion eines Designs überprüft werden. Eine Beschreibung kann sowohl direkt über die Struktur einer Schaltung als auch als abstraktes Verhalten

erfolgen. Die großen Vorteile von VHDL liegen in der Beschreibung eines hierarchischen Designs, bei dem eine Art Kapselung einzelner Schaltungsteile erfolgen kann.

Die Entwicklung von VHDL wurde durch die große Anzahl an proprietären Hardwarebeschreibungs- und Designwerkzeugen und die damit verbundenen Inkompatibilität zwischen den Schaltungsentwürfen motiviert. Die Entwicklung begann Anfang der 1980er Jahre in einem Konsortium von Elektronik-Herstellern und -Anwendern. Im Jahr 1985 wurde VHDL erstmals veröffentlicht und 1987 als IEEE-Standard 1076-1987 angenommen. Es wurden die Erweiterungen IEEE 1076-1993 mit Verbesserungen für die Synthese und IEEE 1164 mit dem Datentyp `std_logic` für eine bessere Simulation sowie IEEE 1076-2002 und 1076-2008 mit verschiedenen Anpassungen an aktuelle Entwicklungen der Schaltungstechnik hinzugefügt. Eine andere Erweiterung für VHDL ist VHDL-AMS, mit der die Beschreibung von analogen Netzwerkelementen, wie Spulen und Kondensatoren, über Differentialgleichungen möglich ist.

Für den Einsatz von VHDL für FPGAs sind die Möglichkeiten für hierarchische Designs und die Verhaltensbeschreibungen wesentlich. Die Designs in dieser Arbeit wurde vollständig mit VHDL erstellt.

Eine weitere Möglichkeit zur Hardwarebeschreibung ist SystemC [64]. Mit SystemC wurde eine Klassenbibliothek für C++ geschaffen, mit der die Beschreibung von elektronischen Komponenten und auch nebenläufigen Prozessen möglich ist. Es lassen sich somit in SystemC sowohl Hardwarestrukturen als auch die auf dieser Hardware ausgeführte Software programmieren und im Verbund simulieren. SystemC wurde im Jahr 2005 als IEEE-Standard angenommen. Es gibt jedoch bisher nur wenige Hardware-Simulations- und Synthese-Tools die diese Beschreibungssprache akzeptieren. Sie kann insbesondere für Hardware-Software-Co-Designs eine wichtige Hilfe sein.

**Design & Synthese** Ein FPGA-Design beschreibt die logische Funktion der einzelnen Logikelemente und die Verknüpfung der Logikelemente miteinander. Ein FPGA-Design wird entweder grafisch oder mittels einer Beschreibungssprache beschrieben.

Nach der Beschreibung des Designs wird dieses durch Hilfsmittel für Design und Synthese in eine konkrete an die Hardware angepasste Struktur überführt. Ein wesentlicher Zwischenschritt hierbei ist die Netzliste. Sie beschreibt die Struktur der Logikelemente und ihrer Verbindungen zueinander und wird nach den syntaktischen und semantischen Prüfungen aus der grafischen oder der HDL<sup>5</sup>-Beschreibung erstellt.

Mit Hilfe einer Netzliste kann die Funktion der Beschreibung grundlegend ohne Berücksichtigung des Zeitverhaltens simuliert und geprüft werden. Anschließend wird die Netzliste durch Fitting und Routing auf den Strukturen des Schaltkreises abgebildet und das Zeitverhalten und der Ressourcenverbrauch optimiert. Danach erfolgt eine erneute Simulation und Prüfung der Funktion mit Berücksichtigung des Zeitverhaltens. Der Assembler erstellt ein Konfigurationsfile, das in den entsprechende Schaltkreis geladen werden kann.

---

<sup>5</sup>HDL - Hardware Description Language

Ein so erzeugtes FPGA-Design ist nur für einen bestimmten Schaltkreis verwendbar. Die ursprüngliche Beschreibung mittels Hardwarebeschreibungssprachen ist für eine Vielzahl von Schaltkreisen und teilweise sogar für die ASIC-Entwicklung verwendbar.

**Funktionale Prüfung** Bei der funktionalen Prüfung werden in Abhängigkeit von der Komplexität des Designs Bitvektoren angelegt und die Ergebnisse mit den Erwartungswerten verglichen. Die Anzahl der Vektoren  $n$  hängt dabei von der Anzahl der Eingänge und der Anzahl der Speicher ab. Bei  $m_e$  Eingängen und  $m_s$  Speichern müssen

$$n_t = m_e \cdot m_s + 1 \quad (3.1)$$

Vektoren für die Netzliste erstellt werden, um eine vollständige Prüfung durchzuführen. Weiterhin muss für jeden Vektor ein Erwartungswert für alle Ausgänge ermittelt werden. Eine fehlerhafte Simulation erfordert die Korrektur der Beschreibung.

**Place & Route** Nach erfolgreicher Simulation wird der Place and Route -Vorgang auf Basis der Netzliste gestartet. In diesem Prozess werden mit Hilfe heuristischer Methoden die Schaltungen der Netzliste auf die Logikelemente aufgeteilt und plaziert. Anschließend wird das Routing vorgenommen und das Zeitverhalten anhand der bekannten Struktur des ausgewählten FPGAs analysiert. Die Schritte Place & Route werden zyklisch nacheinander ausgeführt, bis eine optimale, mindestens aber dem vorgegebenen Zeitverhalten entsprechende Implementierung gefunden wurde.

Im Anschluss an das Place & Route wird mit Hilfe eines Assemblers das Konfigurationsfile für den Schaltkreis generiert.

**Timing Simulation und Testung im System** Nach Abschluss des Synthese-Prozesses ist eine weitere Simulation erforderlich, diesmal jedoch unter Berücksichtigung des analysierten Zeitverhaltens.

Das Design kann nun in den FPGA mit der entsprechenden Umgebung programmiert werden. Zur Verifikation des Designs müssen nun die entsprechenden Testvektoren an das gesamte Design (FPGA und Umgebung) gesendet und die Ergebnisse mit Erwartungswerten verglichen werden. Um dies realisieren zu können, werden Logik-Analysatoren verwendet. Ein Logik-Analysator fertigt ein Protokoll der Zustände einer digitalen Schaltung an und kann diese mit Erwartungswerten vergleichen und für den Menschen anschaulich darstellen. In FPGAs können die nicht benötigten Logikelemente und Speicherzellen ebenfalls in einen internen Logikanalysator umgewandelt werden, sodass auch die Schaltungen innerhalb des laufenden FPGAs prüfbar sind. Nach erfolgreicher Verifizierung ist der Design-Prozess abgeschlossen.



### 3.3.2.3 Vorteile und Nachteile der FPGA

FPGAs sind mit ihrer programmierbaren Hardwarestruktur ein Mittelweg zwischen software-programmierbaren Prozessoren mit fester, möglichst allgemein verwendbarer Hardwarestruktur und ASICs mit an die Anwendung angepasster und optimierter, aber ebenfalls fester Hardwarestruktur ohne Möglichkeit, diese zu programmieren.

Mit FPGAs können anwendungsspezifische Schaltkreise für digitale Schaltungen erstellt werden. Die wesentlichen Vorteile liegen dabei in der Programmierbarkeit und der Möglichkeit, nebenläufige Prozesse voll parallel zu implementieren. Weiterhin ist die große Anzahl und der vielfältige Einsatz der IO-Blöcke sehr vorteilhaft. Dadurch sind FPGAs flexibel verwendbar und können leicht mehrere parallele Datenströme aufnehmen und verarbeiten. Die Ausnutzung der Parallelität von Algorithmen ermöglicht es, mit FPGAs auch bei niedrigerem Takt einen höheren Datendurchsatz als mit Prozessoren zu erreichen. Durch die große Anzahl an PLLs, mit denen Takte verteilt und verändert werden können, sind sie gut als Zentrum eines eingebetteten Designs einsetzbar.

Nachteilig bei FPGAs ist der im Vergleich zu Prozessoren (Mikroprozessoren und DSP) vergleichbar aufwändige Entwicklungsprozess und somit auch die entsprechend längere Entwicklungszeit. Insbesondere für die Entwicklung und Untersuchung neuer Algorithmen ist die Implementierung in FPGAs daher nicht empfehlenswert.

Da in FPGAs durch das Zusammenschalten von Logikelementen programmierbare Hardwarestrukturen realisiert werden, sinkt bei steigender Komplexität häufig die Ausführungsgeschwindigkeit. Während die maximal erreichbare Geschwindigkeit bei ASICs im GHz-Bereich liegt, sind aktuelle FPGAs gegenwärtig auf ca. 500 MHz begrenzt. Im Vergleich zu ASICs ist der Entwicklungsprozess durch die bereits getestete und bekannte innere Struktur des FPGAs, welche in der FPGA-Entwurfssoftware gut simuliert werden kann, erheblich kürzer. Durch die direkte Möglichkeit der Programmierung können verschiedene Versionen einer Implementierung einfach und direkt einander gegenübergestellt werden.

In der Vergangenheit (bis 2005) waren FPGAs häufig sehr kostenintensiv, jedoch werden von vielen FPGA-Herstellern sowohl High-End, als auch Low-Budget-Produkte angeboten. In den Low-Budget-Schaltkreisfamilien werden üblicherweise die Strukturen der High-End-Schaltkreise vorangegangener Schaltkreisfamilien verwendet, wodurch sie eine erhebliche Leistungsfähigkeit aufweisen, die auch für eine Vielzahl von Bildverarbeitungsaufgaben, wie z.B. die in dieser Arbeit nachfolgend beschriebenen Algorithmen, ausreichend ist.

### 3.3.3 Vergleich von Hardware- und Softwarekonzepten zur Parallelisierung

Zur Realisierung einer echtzeitfähigen Implementierung von Algorithmen müssen sowohl die Möglichkeiten zur Steigerung der Verarbeitungsgeschwindigkeit der Algorithmen als auch der gewählten Plattform ausgenutzt werden. In Abhängigkeit von den Algorithmen entstehen da-

bei sehr unterschiedliche Anforderungen an die Implementierungsplattform. Innerhalb eines Algorithmus kann entweder der Datenfluss oder der Kontrollfluss dominieren. In kontrollflussorientierten Algorithmen sind bedingte Sprünge und Entscheidungen vorherrschend. Diese Algorithmen werden daher sehr gut durch serielle Anweisungen repräsentiert und können daher gut auf SISD-Plattformen implementiert werden. SISD-Plattformen, wie die "von-Neumann"-Rechnerarchitektur haben keine Parallelisierungsmöglichkeiten. Bedingte Sprünge, Entscheidungen und zum Teil auch Schleifen führen zu schwer abschätzbaren Abhängigkeiten in Kontroll- und Datenfluss. Dadurch sind Parallelisierungen schwer zu realisieren und bedürfen einer aufwändigen Analyse.

Ist ein SISD-System mit der mehrfachen Taktfrequenz der Eingangsdaten vorhanden, kann eine scheinbare Parallelisierung durch das Prinzip des Multitaskings erfolgen. Das Multitasking ist ein Softwareansatz, bei dem dieselbe Hardware zeitlich nacheinander von verschiedenen Programmen bzw. Arbeitsschritten eines Programmes verwendet wird. Es gibt im Wesentlichen zwei Organisationsformen, das preemtive und das kooperative Multitasking. Beim kooperativen Multitasking muss jedes Programm selbstständig den Prozessor nach einer gewissen Zeitspanne wieder freigeben. Dieses Verfahren eignet sich nicht für den Echtzeitbetrieb, es sei denn, die Software wird komplett darauf ausgelegt.

Durchgesetzt hat sich das präemtive Multitasking, bei dem das Betriebssystem dafür sorgt, dass alle Prozesse gemäß ihrer Priorität Prozessorzeit bekommen [86]. Dabei wird das sogenannte Zeitscheibenverfahren eingesetzt, bei dem zyklisch immer dieselben Prozesse für eine vorher definierte Zeit aufgerufen werden. Hierfür ist es notwendig, dass in regelmäßigen Abständen ein im Betriebssystem enthaltener Scheduler zur Ausführung kommt. Dies wird über Zeitinterrupte gesteuert. Für diese Art des Multitasking in Echtzeitrechnern gibt es spezielle Systemuhren, die mehrere unterschiedliche Zeitbereiche, bzw. spezielle Zeitpunkte mit Interrupten belegen können. Somit kann die Hardware effizienter genutzt werden und es wird nahezu zeitgleich auf mehrere Ereignisse reagiert.

Wesentliche Bedingung für die Echtzeitfähigkeit dieses Ansatzes sind ausreichende Reserven bezüglich der Rechenleistung [72], damit die jeweiligen Prozesse ihre Aufgaben zu Ende führen können und nicht in der Ausführung am Ende der Zeitscheibe unterbrochen werden. Ebenso führen viele parallele Prozesse zu einem System-Overhead, der das System zusätzlich ausbremst. Da bei der Stereobildverarbeitung ein sehr hohes Datenaufkommen verarbeitet werden muss, ist eine virtuelle Parallelisierung mittels Multitasking in einem eingebetteten System nicht einsetzbar. Rechnersysteme, die die erforderliche Rechenleistung besitzen, sind komplexe Systeme aus mehreren Schaltkreisen und damit physisch zu groß. Darüber hinaus sind Leistungsaufnahme und Abwärme mit mehreren 10 W zu hoch für ein eingebettetes System.

Bilden sich in einem Algorithmus mehrere unabhängige Kontrollflüsse heraus oder sind die Kontrollflüsse sehr stark von den bearbeiteten Daten abhängig, so bietet sich eine Implementierung auf einer MIMD-Plattform an. In MIMD-Plattformen werden mehrere Kontrollflüsse auf mehrere unabhängige Datenflüsse angewendet. Beispiele hierfür sind z.B. verteilte Systeme in der Informations- oder der Automatisierungstechnik. Während in der Informationstechnik vorwiegend Rechner mit einer hohen Rechenleistung vorhanden sind, werden in der Automa-

tisierungstechnik häufig viele eingebettete Systeme miteinander vernetzt. Die Kommunikation zwischen den einzelnen Rechnersystemen erfolgt durch leistungsfähige Netzwerke. In der Automatisierungstechnik kommen vorwiegend Systeme mit getrennten Speichern zum Einsatz, sodass der Datenaustausch vollständig über das Netzwerk abgewickelt wird. Die Datenmengen sind hierbei in der Regel gering. Für die schnelle Informationsverarbeitung werden je nach Anwendung und Datenaufkommen Systeme mit getrenntem oder gemeinsamem Speicher verwendet. In beiden Fällen ist eine Synchronisierung der Daten erforderlich, um stets konsistente Daten verfügbar zu haben.

Müssen viele Datenströme mit dem selben Kontrollfluss bearbeitet werden und ist dieser Kontrollfluss unabhängig von den Datenströmen, sind SIMD-Plattformen prädestiniert für die Verarbeitung. Zu den SIMD-Plattformen gehören neben den klassischen Vektorrechnern auch DSPs [18] und mittlerweile auch viele moderne Mikroprozessoren. Klassische Vektorrechner sind für eingebettete Systeme nicht verfügbar, der Cell-Prozessor kommt diesem Ansatz am nächsten. In eingebetteten Systemen werden für diese Aufgaben überwiegend DSPs verwendet. In SIMD-Systemen wird eine feingliedrige Parallelisierung vorgenommen.

Ein durch den Datenfluss dominierter Algorithmus enthält wenig Sprungbefehle und verknüpft die Daten vorwiegend direkt über Operatoren miteinander. Der Kontrollfluss ist hierbei nur bedingt bzw. nicht von den Daten abhängig, womit vorwiegend die Datenabhängigkeiten zum Tragen kommen. Durch die Abhängigkeiten der Daten untereinander entstehen häufig Strukturen nebenläufiger Anweisungen, also Anweisungen, die unabhängig voneinander parallel ausgeführt werden können. Zur Implementierung solcher Algorithmen sind SISD, MIMD und MISD-Plattformen geeignet.

Bei der Verwendung von Datenflussrechnern, wie sie von Ungerer [17] beschrieben wurden, werden diese nebenläufigen Strukturen automatisch berücksichtigt, da dort nur die Daten und der Operator zur Ausführung einer Anweisung definiert sind. Die Ausführungsreihenfolge wird durch die Verfügbarkeit der Daten bestimmt. Wie bereits in Abschnitt 3.1.2 beschrieben, haben sich die Datenflussrechner bisher nicht durchgesetzt. In der Einteilung nach Flynn werden die Datenflussrechner den MISD-Rechnern zugeschrieben.

Eine weitere Möglichkeit, datenflussorientierte Algorithmen zu implementieren, ist die Hardwareimplementierung. Sie bietet sich an, wenn nur ein konkreter Algorithmus, dieser jedoch sehr schnell und kompakt, ausgeführt werden soll. Dies ist bei eingebetteten Systemen häufig der Fall.

Bei der Parallelisierung der Datenverarbeitung in Hardware, wie z.B. einem FPGA oder ASIC, sind alle Verarbeitungsschritte als Strukturen parallel in Hardware aufgebaut und arbeiten zeitgleich. Durch die Datenflussabhängigkeiten, die sich im Datenflussgraph ergeben, kann direkt eine Struktur für die Hardwareimplementierung abgeleitet werden. Dadurch ergibt sich eine hohe Effektivität bei gleichmäßigen Datenströmen.

Wenn alle Teile der Verarbeitung explizit in Hardware realisiert werden, ist eine Erhöhung des Systemtaktes mit einer äquivalenten Zunahme der Verarbeitungsgeschwindigkeit gleichzusetzen. Die Geschwindigkeit der Logik in der Hardware muss im Allgemeinen nur der Geschwindigkeit

der Eingangsdaten entsprechen. Die Steigerung der Verarbeitungsgeschwindigkeit durch Erhöhung des Systemtaktes erfolgt auch bei Softwareimplementierungen. Wenn jedoch zwischen zu vielen Prozessen umgeschaltet werden muss, dann macht die Umschaltzeit und die hohe Zykluszeit das System ineffizient [86]. Die Umschaltzeit entsteht dadurch, dass zunächst die Daten und Befehle des einen Prozesses im RAM zwischengespeichert und anschließend die Daten und Befehle des nächsten Prozesses geladen werden müssen.

Die Verarbeitung von kontrollflussorientierten Algorithmen in Hardware führt in Abhängigkeit von der Ausführungswahrscheinlichkeit der Pfade im Kontrollflussgraph auch zur Realisierung von Schaltungen, die nur bei sehr seltenen Ereignissen aktiv werden. Die direkte Hardwareimplementierung von kontrollflussorientierten Algorithmen ist somit immer ineffizient. Wenn die Verarbeitung jedoch unmittelbar und sehr schnell durchgeführt werden muss, z.B. Notabschaltungssysteme, kann es dennoch sinnvoll sein, diese direkt in Hardware zu realisieren. Als Alternative bietet sich eine gut abgestimmte Vereinigung von Hardware und Software in einem Hardware-Software-Co-Design an.

Abschließend lässt sich feststellen, dass die wichtigste Ressource bei einer reinen Hardwareimplementierung nur die Chipfläche ist, während bei einer Softwarelösung die Zeit von entscheidender Bedeutung ist. Somit sollte die verwendete Hardware nach Möglichkeit so angesteuert werden, dass kontinuierlich Daten zur Verarbeitung vorhanden sind. Aufgrund ihrer Struktur hat eine direkte Hardwareimplementierung ausgeprägte zeitliche Randbedingungen, welche während des Entwurfsprozesses an die Echtzeitanforderungen angepasst werden müssen. Bei einer Softwarerealisierung unter Echtzeitrandbedingungen hingegen sollte die massive Datenverarbeitung vermieden werden bzw. auf andere Systeme verlagert werden, damit auf die einzelnen Ereignisse ausreichend schnell reagiert werden kann. Auf dieser Basis muss auch die Partitionierung eines Algorithmus auf Hardware und Software in einem Co-Design vorgenommen werden.

### 3.4 Arithmetik in der Digitaltechnik

In der Digitaltechnik werden alle Rechenvorschriften auf die Boolesche Algebra zurückgeführt, da sie sehr vorteilhaft mit der Mikroelektronik implementiert werden kann. Abstrahiert werden neben logischen Ausdrücken das binäre, das dezimale und das hexadezimale Zahlensystem verwendet.

Während das dezimale und das hexadezimale Zahlensystem bis auf wenige Ausnahmen nur für die Darstellung zur besseren Lesbarkeit verwendet wird, erfolgen die Rechenoperationen fast immer direkt im Binärsystem.

Zur Darstellung von negativen Zahlen hat sich die Zweierkomplement-Darstellung in Rechenwerken durchgesetzt. Hierbei werden positive Zahlen durch einfache Binärzahlen mit mindestens einer führenden Null und negative Zahlen durch das um 1 erhöhte Komplement mit mindestens einer führenden 1 dargestellt. Die Vorteile der Zweierkomplementdarstellung gegenüber anderen Darstellungen sind die eindeutige Repräsentation der Null und die Möglichkeit der Verwendung

einfacher Addierer für Addition und Subtraktion.

Die Berücksichtigung der Bereichsüberschreitung ist insbesondere bei Softwareimplementierungen relevant, da die Prozessoren nur über eine begrenzte Bitbreite verfügen. Während in der PC-Architektur in der Regel die Prozessoren über Rechenwerke mit 32 oder 64 Bit verfügen, sind im Bereich der eingebetteten Systeme die meisten Prozessoren mit 8 oder 16 Bit Rechenwerken ausgestattet. Somit kommt es häufiger zu Bereichsüberschreitungen. Zu beachten ist dies auch bei der Portierung auf einen anderen Prozessortyp.

In DSPs wird durch die fraktale Darstellung der Zahlen und zusätzliche führende Bits für den Programmierer eine Erleichterung ermöglicht. Der Grundgedanke der fraktalen Repräsentation ist die Normierung auf den maximal durch das Gesamtsystem zu verarbeitenden Wert. Dieser Wert wird damit durch die Zahl 1 repräsentiert. Die Darstellung der Zahl der Bitbreite  $n$  wird ausgehend von  $m \cdot 2^n$  durch  $m \cdot 2^{-n}$  ersetzt. Der Vorteil der fraktalen Darstellung besteht in dem eindeutigen Verhältnis von Zahlen unterschiedlicher Bitbreiten zueinander.

In einer Hardwareimplementierung kann prinzipiell die Bitbreite einer Zahl "beliebig" erweitert werden, sodass eine Bereichsüberschreitung vermieden werden kann. Wird aus technischen Gründen die Bitbreite eingeschränkt, ist eine Berücksichtigung der Bereichsüberschreitung ebenfalls erforderlich.

Zur Darstellung von gebrochenen rationalen Zahlen wird entweder die Festkomma- oder die Gleitkommadarstellung verwendet. Bei der Festkommadarstellung liegt die Definition der Position des Kommas im Algorithmus. Damit muss der Programmierer darauf achten, dass bei Rechenoperationen die Operanden den selben Zahlenbereich repräsentieren.

Bei der Gleitkommadarstellung wird die Zahl in Mantisse  $m_z$  und Exponent  $n_z$  unterteilt.

$$x_{2gl} = m_z \cdot 2^{n_z} \quad (3.2)$$

Bei Rechenoperationen in Prozessoren sind die Gleitkommaoperationen beliebt, da die Probleme der Bereichsüberschreitung hierbei durch die Arithmetik des Prozessors abgefangen werden und wesentlich größere Zahlenbereiche repräsentiert werden können. Multiplikationen sind mit Gleitkommaarithmetik sehr einfach zu realisieren, da nur die Mantissen vorzeichenrichtig multipliziert und die Exponenten der Zahlen vorzeichenrichtig addiert werden müssen. Bei Gleitkommaadditionen müssen die Summanden in Festkommazahlen des gleichen Exponenten umgewandelt werden, wodurch unter Umständen sehr große Zahlen entstehen, dadurch sind Additionen mit Gleitkommaarithmetik deutlich aufwändiger als mit Festkommaarithmetik.

Für eingebettete Systeme mit harten Echtzeitforderungen ist die Verwendung der Gleitkommaarithmetik problematisch, da Gleitkommaoperationen in der Regel deutlich mehr Takte benötigen als Festkommaoperationen. Ausnahmen bilden hierbei Gleitkomma-DSPs, die Gleitkommaoperationen ebenso schnell ausführen wie Festkommaoperationen. Gleitkommarechenwerke sind sehr aufwändig und daher sind diese Prozessoren deutlich komplexer als Festkomma-DSPs.

In Hardwareimplementierungen für die Signalverarbeitung wird aufgrund der variablen Bitbrei-

te bei arithmetischen Operationen im Allgemeinen auf Gleitkommaimplementierungen weitestgehend [61] verzichtet. Insbesondere Gleitkommaadditionen sind sehr aufwändig zu implementieren und in FPGAs im Verhältnis zu äquivalenten Festkommaaddierern langsam. Für spezielle Anforderungen gibt es spezielle Bibliotheken mit vorgefertigten Gleitkommakomponenten.

Die Division folgt prinzipiell der Division mit Rest und stellt somit ein sukzessives Verfahren dar. Die Anzahl der Schritte wird durch die Bitbreite des Dividierers bestimmt. Divisionen sind somit im Vergleich zu bisher besprochenen Rechenoperationen verhältnismäßig aufwändig. In Prozessoren kann die Division bei endlichen Brüchen nach Erreichen des korrekten Ergebnisses abgebrochen werden. Bei irrationalen Zahlen kann die Rechnung bis zu einer gewünschten Genauigkeit fortgesetzt werden. In Hardwareimplementierungen kann die Division ebenfalls als iteratives Verfahren in serieller Arithmetik implementiert werden. Für Systeme mit hohem Datendurchsatz ist jedoch eine parallele Implementierung aller Schritte der Division mittels Pipelining erforderlich. Dadurch sind nach der Implementierung maximale Genauigkeit und Zeitdauer der Berechnung festgelegt. Hardware-Dividierer sind sehr aufwändig und erfordern viele Ressourcen.

Für Divisionen durch Potenzen von 2 kann die Rechnung durch eine Bitverschiebeoperation ersetzt werden. In Hardwareimplementierungen können Schiebeoperationen, wenn sie im Algorithmus unabhängig von den Eingangsdaten sind, durch die feste Zuordnung von Bits ersetzt werden.

Wurzel- und Logarithmus-Operationen können nicht direkt implementiert werden. Sie werden in üblicher Weise durch Iterationsverfahren bis zur geforderten Genauigkeit berechnet. Ist eine Implementierung in Echtzeit erforderlich, stehen alternativ Ersatzfunktionen zur Verfügung, z.B. durch Polynome, die den Funktionsverlauf in einem bestimmten Arbeitsbereich nachbilden, oder Look-Up-Tables, bei denen die Ergebnisse der Operationen in einem Speicher abgelegt sind. Ist die Repräsentation eines großen Werte-Bereiches erforderlich, ist eine grobe Look-Up-Table mit nachgeschalteter Interpolation möglich. Ein Aufheben der Operation durch die entsprechende Umkehroperation sollte in allen Fällen in Betracht gezogen werden, in denen diese Gegenoperation direkt berechnet werden kann.

### **3.5 Diskussion einer echtzeitfähigen Plattform für die Bildverarbeitung**

Für die Verarbeitung von Stereobildern sind die in eingebetteten Systemen üblicherweise eingesetzten Universalprozessoren nur bedingt geeignet. Sie besitzen nur begrenzte Parallelisierungsmöglichkeiten, da sie vorwiegend zur Messwerterfassung und Steuerung bzw. Regelung eingesetzt werden. Eine Echtzeitfähigkeit wird in der Regel durch Einsatz eines Echtzeitbetriebssystems erreicht.

Für eine kontinuierliche Daten- und Signalverarbeitung in Echtzeit werden bei eingebetteten Systemen vorwiegend digitale Signalprozessoren (DSP) eingesetzt. Ihre Leistungsfähigkeit ist

in der letzten Dekade erheblich gestiegen. Die meisten DSPs orientieren sich jedoch weiterhin an der Verarbeitung eindimensionaler Signale. Multicore-DSPs sind durch die vergrößerten IO-Möglichkeiten für eine Verarbeitung von Stereobilddaten geeignet. Ihre Leistungsfähigkeit diesbezüglich wird in Abschnitt 4.3.1 näher erläutert.

In den letzten 5 Jahren wurden einige für die Bildverarbeitung interessante Ansätze entwickelt und auf den Markt gebracht. Es sind vorwiegend Ansätze, die SIMD- oder/und MIMD-Strukturen aufweisen. Die größten Fortschritte sind im Bereich der Graphic Processing Units, die eine SIMD-Struktur mit bis zu 128 Bit Fest- und Gleitkomma-Pipelines umsetzen, zu verzeichnen. Auch der CELL-Prozessor, der aufgrund seiner Struktur sowohl als SIMD- oder als MIMD-Rechner arbeiten kann, ist gut für eine Bearbeitung von Mehrbildaufnahmen geeignet. Nachteilig wirkt sich sowohl bei den GPUs, als auch beim CELL-Prozessor die hohe Leistungsaufnahme aus. Weiterhin sind GPUs gegenwärtig von einem übergeordneten System abhängig. Für die eingebetteten Systeme bleiben der MIMD-Ansatz mit 1024 8-Bit Prozessoren der Firma Rapport oder der Ansatz von NEC mit zwanzig 32 Bit-Prozessoren interessant. Sie wurden in der Arbeit jedoch nicht weiter betrachtet.

Eine Hardwareimplementierung eines Auswerteverfahrens für Stereobilder ist gut für eine Echtzeitverarbeitung geeignet. ASICs werden aufgrund der Entwicklungszeit und der hohen Kosten nicht weiter untersucht. Für die Stereobildverarbeitung bieten sich FPGA-basierte Systeme aufgrund ihrer großen Flexibilität an. FPGAs werden in einer großen Vielfalt bezüglich der Rechenleistung, der maximalen Geschwindigkeit und der Anzahl freiprogrammierbarer IO-Pins geliefert. Sie sind für die gängigen Temperaturbereiche verfügbar und werden in eingebetteten Systemen aufgrund ihrer Vielfalt und niedrigen Leistungsaufnahme eingesetzt. Die aktuell größten FPGAs besitzen 680.000 Logikelemente, nahezu 1.500 frei programmierbare Pins und erreichen Verarbeitungsgeschwindigkeiten von 500 MHz. Die Entwicklungen der letzten 5 Jahre haben die Leistungsfähigkeit nahezu vervierfacht. Die Implementierung von Softcore-Prozessoren in FPGAs wurde ebenfalls stark verbessert, sodass sich erhebliche Vorteile für die Verarbeitung von kontrollflussorientierten Algorithmen ergeben. Das in dieser Arbeit eingesetzte System basiert auf Schaltkreisen der Stratixfamilie der Firma Altera EP1S60 mit ca. 60.000 Logikelementen und einem maximalen Arbeitstakt von 300 MHz. Durch die direkte Programmierung der Logik kann eine starke Parallelisierung von Algorithmen bei der Implementierung erreicht werden.

Nachteil der FPGAs ist ihre aufwändige Programmierung. Sie erfordert viel Wissen über den internen Aufbau von FPGAs, um komplexe Aufgaben realisieren zu können. Eine gewisse Analogie ist bei der Programmierung von DSPs zu beobachten. Ist eine optimale Datenverarbeitung gefordert, muss die Abarbeitung sehr stark an die Gegebenheiten des DSPs angepasst werden. Durch die zunehmende Größe von FPGAs werden jedoch auch alternative Ansätze interessant. So kann aus Matlab-Simulink heraus eine Steuerung direkt für ein FPGA umgesetzt werden.

Im Kapitel 4 wird die Leistungsfähigkeit von DSPs und FPGAs bezüglich der Verarbeitung von Korrelationskriterien untersucht.





# Kapitel 4

## Konzept zur stereophotogrammetrischen Ermittlung von Tiefenkarten in Echtzeit

In diesem Kapitel wird ein Konzept zur Ermittlung von Tiefenkarten aus Stereobildern in Echtzeit aus konkreten Problemstellungen im Nahbereich und im Fernbereich abgeleitet. Dazu wird zum einen auf die Verfahren zur blockbasierten Korrelation aus Abschnitt 2.4.2.2, sowie die Korrelationskriterien SADMF und NKKFMF zurückgegriffen. Als Implementierungsplattformen werden DSPs (Abschnitt 3.2.2) und FPGAs (Abschnitt 3.3.2) diskutiert.

Dazu werden zunächst zwei Problemstellungen der Umfelderkennung angesprochen und die grundlegenden Probleme bei der Realisierung erläutert. Die konkreten Randbedingungen für eine Implementierung können erst anhand der konkreten Anwendung ermittelt werden. Die wesentlichen Faktoren sind der Messbereich, die erforderliche Genauigkeit und die maximale Verarbeitungszeit. Weiterhin werden anhand von Untersuchungen bezüglich des Rechenaufwandes von Korrelationskriterien sowie Algorithmen zur Stereobildauswertung die rechen-technischen Erfordernisse bestimmt. Daraus werden die Parameter für die Implementierungsplattform abgeleitet.

### 4.1 Umfelderkennung mit photogrammetrischer Stereobildauswertung in Echtzeit

Die Nutzung der Stereophotogrammetrie für die Steuerung von Fahrzeugen wird bereits seit den 1980er Jahren in größerem Umfang untersucht [94]. Die wesentlichen Faktoren für die Aufnahme von Informationen in einem steuerungstechnischen und regelungstechnischen Kontext sind die relevanten Informationen an sich, die Zuverlässigkeit und die Echtzeitfähigkeit (d.h. die rechtzeitige Ermittlung und Reaktion auf Veränderungen). Die Bedeutung dieser Begriffe kann zwar allgemein definiert werden, konkrete Vorgaben ergeben sich jedoch nur durch die Anwendung.

Im folgenden Kapitel sollen anhand der Anwendungsfälle Spurwechselassistent und Hindernisdetektion für fahrerlose Fahrzeuge die Faktoren relevante Information, Zuverlässigkeit und Echtzeitfähigkeit untersucht werden. Zunächst werden die beiden Anwendungsfälle grob vorgestellt.

**Auswahl relevanter Informationen** Die relevanten Informationen sind alle Informationen, die zur Lösung der steuerungs- und regelungstechnischen Aufgabe erforderlich sind. Im Fall der Umfelderkennung zählen zu den relevanten Informationen die Position, Größe und Geschwindigkeitsvektoren aller in der Nähe des Fahrzeugs und insbesondere der im Fahrweg befindlichen Objekte. Bei der Auswahl der Informationen führen zu viele Daten zur Überlastung des Auswertesystems und zu wenig Daten bewirken, dass nicht alle Objekte erfasst werden können.

**Zuverlässigkeit** Zuverlässigkeit bedeutet, dass die Werte mit einem bekannten Fehler ermittelt werden und dass das Messsystem ein Problem bzw. eine Fehlfunktion automatisch signalisiert. Die Zuverlässigkeit ist im Rahmen der Umfelderkennung sehr problematisch, da insbesondere im urbanen, aber auch im ländlichen Umfeld eine Vielzahl unterschiedlicher Objekte existieren, die ein Hindernis für das Fahrzeug darstellen können. Je nach Sensorsystem treten hierbei unterschiedliche Probleme auf. Während mittels Radar nah beieinander liegende Objekte nur schwer getrennt werden können, ist, abgesehen von der Entfernung, die korrekte Bestimmung der 3D-Position ein sehr komplexes Problem. Weiterhin können metallene Gegenstände in der Fahrbahn fehlerhafte Messungen hervorrufen. Für kamerabasierte Systeme bilden insbesondere unkooperative Messsituationen aufgrund der Beleuchtung ein Problem. Auch sich wiederholende Muster oder Regionen mit zu wenig Textur sind problematisch. Bei Laserscannern können die konkreten Punkte sehr zuverlässig ermittelt werden. Die Rekonstruktion der Umwelt ist in Abhängigkeit von der Anzahl und der Verteilung der Punkte unter Umständen sehr schwierig. Ebenso wie bei kamerabasierten Systemen entstehen Probleme durch unkooperative Messsituationen bezüglich Blendung. Für die Zuverlässigkeit sind neben den eingesetzten Sensoren und den verwendeten Algorithmen außerdem die Implementierung und damit auch die Implementierungsplattform relevant.

**Echtzeitfähigkeit** Die Echtzeitfähigkeit eines Systems wird durch die Eigenschaft definiert, dass eine dem Prozess angemessene Antwortzeit für eine Verarbeitung immer erreicht werden kann. Für die Antwortzeit sind vor allem der verwendete Algorithmus, die Menge der zu verarbeitenden Daten und die Implementierungsplattform entscheidend. Bei Hardwareimplementierungen ist die Antwortzeit jeder Teilschaltung wichtig für die Funktion der gesamten Schaltung. Die Beachtung einer definierten Antwortzeit ist daher im Entwurfsprozess inbegriffen. Somit sind Hardwareimplementierungen von ihrem Ansatz her der Echtzeitverarbeitung sehr ähnlich. Bei Softwarelösungen muss dies bei der Implementierung und ggf. auch durch Einsatz eines geeigneten Echtzeitbetriebssystems zusätzlich berücksichtigt werden. Der Entwurfsprozess der Software wird dadurch aufwändiger.

**Umfelderfassung** Die Umfelderfassung bezeichnet die Detektion, die Vermessung und ggf. auch die Klassifizierung aller Objekte in der Umgebung eines Fahrzeugs, welches im Allgemeinen als Egofahrzeug bezeichnet wird. Für Umfelderfassungssysteme müssen hierbei zwei prinzipielle Situationen unterschieden werden: Zum einen der Einsatz in einem automatisierten Fahrzeug. Hierbei müssen nur die physikalischen Randbedingungen für die konkrete Reaktion berücksichtigt werden. Zum anderen der Einsatz eines Umfelderfassungssystems im Rahmen eines Komfortsystems, z.B. PKW. In diesem Fall spielen sowohl die physikalischen Faktoren (z.B. Geschwindigkeit, max. Bremsbeschleunigung) wie auch die menschlichen Faktoren (z.B. die Reaktionszeit) eine Rolle.

Als Zielplattform soll prinzipiell eine Technologie verwendet werden, die in einem eingebetteten System in einem Fahrzeug eingesetzt werden kann. Somit wird die Auswahl der Plattformen auf mikroprozessorbasierte (z.B. ARM, MIPS, Z8 etc), DSP-basierte (Analog Devices Blackfin, TigerShark oder Freescale MSC8144, Texas Instruments (C64-Serie)) oder auf hardwarebasierte Plattformen (ASIC, FPGA, PLD) eingeschränkt.

## 4.2 Einsatzszenarien der Umfelderfassung

Durch eine große Anzahl verschiedener Anwendungen für die Umfelderfassung ergeben sich Sensorsysteme mit sehr unterschiedlichen Parametern.

Maximales Ziel der Umfelderfassung ist es, Daten über alle Objekte in der gesamten Umgebung des Fahrzeugs zu sammeln und auszuwerten. Dies kann nur durch eine Vielzahl von Sensoren, die über das Fahrzeug verteilt angebracht sind, realisiert werden. Für sicherheitsrelevante Sensorsysteme ergibt sich die Notwendigkeit der Redundanz. Ein wesentlicher Vorteil von kamerabasierten Sensoren ist die Tatsache, dass sie für eine große Anzahl von Messaufgaben eingesetzt werden können und somit als redundantes Sensorsystem für viele Anwendungsfälle im Fahrzeug zur Verfügung stehen.

Zwei wesentliche Einsatzszenarien, bei denen sich durch die Verwendung von kamerabasierten Systemen Vorteile ergeben, sind die Erfassung und Positionsbestimmung im Fernbereich (10 - 150 m) und die Hindernisdetektion im Nahbereich (1 - 20 m). Neben der Möglichkeit, Orientierungsmerkmale, die für Menschen erstellt wurden, zu nutzen, kann durch die hohe Auflösung der Kameras die Winkelauflösung gegenüber Radarsensoren deutlich erhöht und Objekte besser getrennt werden. Weiterhin sind optische Sensoren für einen anderen Frequenzbereich des elektromagnetischen Spektrums empfindlich, so dass durch eine Kombination mehrerer Sensoren das Gesamtsystem deutlich unempfindlicher gegenüber Störungen wird.

### 4.2.1 Objektdetektion im Fernbereich

Ein Anwendungsbeispiel für eine Objektdetektion im Bereich von 10 - 150 m liefert ein beispielhaftes Fahrerassistenzsystem, das den Bereich vor und hinter einem Fahrzeug überwacht

um Kollisionen zu vermeiden. Um dieses Szenario zu analysieren, wurden die Anforderungen für einen Spurwechsellassistenten und ein Adaptive-Cruise-Control-System (ACC) untersucht. Ein Spurwechsellassistent beobachtet den rückwärtigen Raum eines Fahrzeugs und bestimmt unter Berücksichtigung der Eigengeschwindigkeit, der Fremdgeschwindigkeit und des Abstandes, ob die Fahrer anderer Fahrzeuge sicher auf einen Spurwechsel reagieren können. Im Falle der Gefahr einer Kollision wird der Fahrer des eigenen Fahrzeugs gewarnt.

Die Erfassung des Umfeldes erfolgt bei hohen Fahrzeuggeschwindigkeiten und bei unterschiedlichen Beleuchtungsbedingungen. Weiterhin sind sowohl eine Positionsbestimmung von Objekten als auch eine grobe Klassifizierung in bewegte und statische Objekte sowie eine Spuruordnung erforderlich. Da das System nicht selbst in die Fahrzeugsteuerung eingreift, muss die Verzögerung der menschlichen Reaktion von ca. 1 Sekunde bei der Bestimmung der Echtzeitanforderungen mit eingerechnet werden.

Der Messbereich wurde auf 10 - 150 m festgesetzt und sollte mit einer Bildfrequenz von mindestens 25 Bildern/s abgetastet werden. Die untere Grenze ergibt sich aus der für die Messung über den großen Messbereich erforderlichen Basisbreite und dem Öffnungswinkel der Objektivs. Die obere Grenze des Messbereiches und die Abtastrate ergeben sich aus den zeitlichen Gegebenheiten bei unterschiedlichen Geschwindigkeiten. Es wurden Objektivs mit einem Öffnungswinkel von  $30^\circ$  und einer Brennweite von 25 mm verwendet. Die Basisbreite wird an die Breite eines Fahrzeugs angepasst und mit 70 cm angesetzt. Der Sichtbereich ist in diesem Fall relativ schmal im Vergleich zur Länge des Messbereiches, da dieses System vorwiegend für Autobahnen und Landstraßen eingesetzt wird und somit nur sehr weite Kurven berücksichtigt werden müssen. Um diesen Messbereich mit einem Kamerasystem abdecken zu können, ist der Einsatz des Normalfalls der Stereophotogrammetrie notwendig, da durch die parallelen Kameraachsen über einen sehr großen Entfernungsbereich eine Überdeckung der Bilder erreicht wird. Im allgemeinen Fall der Stereophotogrammetrie ist je nach Winkel der Kameras zueinander die Überdeckung in einem kleinen Messbereich maximal. Die Sehstrahlen laufen dann jedoch schnell auseinander.

Zunächst werden die Echtzeitbedingungen in Abschnitt 4.2.3.1 ermittelt. Im Anschluss werden der erforderliche Messbereich und die notwendige Auflösung diskutiert. Für die folgenden Untersuchungen wird angenommen, dass durch die Aufnahme eines Bildpaares die Position eines Fahrzeugs festgestellt werden kann. Die Position und der Bewegungsvektor sollten über mehrere Bilder verfolgt werden, um stabile und zuverlässige Werte zu erhalten. Mit einem Kalman-Filter kann eine solche Glättung erfolgen. Diese Zeit muss bei der Ermittlung der Echtzeitbedingung ebenfalls berücksichtigt werden.

Ausgehend davon, dass die Berechnung der Position und Geschwindigkeit und die Verifizierung für jedes Bild in Echtzeit berechnet werden können, ist eine Vermessung des Objektes über 5 Bilder [39] erforderlich, um stabile Ergebnisse mit dem statistischen Verfahren zur Auswertung zu erreichen. Daraus ergibt sich eine Zeitspanne von 200 ms. Bezieht man die menschliche Reaktionszeit mit 0,5 - 1s [24] mit ein, ergibt sich ein minimales Zeitfenster von 1,25 s.

Serienfahrzeuge aus Deutschland sind üblicherweise bei einer Geschwindigkeit von 250 km/h

abgeriegelt. Bei einer Geschwindigkeit von 250 km/h legt ein Fahrzeug in 1,25 s einen Weg von 86 m zurück. Für minimale Sicherheit in diesem Fall ist also ein Messbereich von 100 m erforderlich. Bei einem Messbereich von 150 m ergibt sich somit eine zusätzliche zeitliche Sicherheit von ca. 1 s.

In der Tabelle 4.1 ist der zurückgelegte Weg in m über der Zeit bei verschiedenen Geschwindigkeiten dargestellt. Beim Auffahren auf die Autobahn treten Differenzgeschwindigkeiten bis 100 km/h sehr häufig auf. Dieser Fall kann als Worst Case eingestuft werden. Dabei könnte es im Extremfall vorkommen, dass das eigene Fahrzeug steht und ein neu im Messbereich auftauchendes Fahrzeug mit einer Geschwindigkeit von 100 - 250 km/h im Rückraum erscheint. Weiterhin ergibt sich ein Bereich von Differenzgeschwindigkeiten für den Einsatz eines solchen Sensors im Frontbereich. Bei Fahrten auf Landstraßen begegnen sich häufig Fahrzeuge, die im Bereich von 100 - 120 km/h mit entgegengesetzten Bewegungsvektoren fahren.

Zeit [ms]	zurückgelegter Weg [m]		
	bei 50 km/h	bei 100 km/h	bei 250 km/h
1000	13,80	27,70	69,40
500	6,90	13,80	34,70
100	1,38	2,70	6,94
40	0,55	1,10	2,77
20	0,78	0,55	1,38

Tabelle 4.1: Echtzeitanforderung: Vergleich des Einflusses verschiedener Geschwindigkeiten

Um ein Fahrzeug im Verkehr verfolgen zu können, sollte mindestens einmal pro Fahrzeuglänge ein Positionswert vorliegen. Die Länge von PKWs variiert üblicherweise im Bereich von 3 - 7 m. Wird von einer maximale Geschwindigkeitsdifferenz von 250 km/h ausgegangen, muss eine Abtastung mit 20 Bildern/s und somit alle 40 ms erfolgen. Mit einer Abtastrate von 20 Bildern/s kann eine Vermessung im Rückraum sicher realisiert werden. Im Gegenverkehrsfall für eine Geschwindigkeitsdifferenz von 250 km/h ist diese Abtastrate auch gerade noch ausreichend. Da jedoch andere physikalische Parameter gelten und ein größerer Sicherheitsabstand bei der Detektion eines entgegenkommenden Fahrzeugs erforderlich ist, sollte die Abtastrate nochmals verdoppelt werden. Somit kann dieser Anwendungsfall mit einer Bildfrequenz von 50 Bildern/s umgesetzt werden. Die höhere erforderliche Sicherheit resultiert aus dem sich weiter stark reduzierenden Abstand beider Fahrzeuge im Falle einer Notbremsung oder eines Ausweichmanövers, während im Rückraum oder bei der Fahrt auf der Autobahn beide Fahrzeuge in die gleiche Richtung fahren und sich somit selbst bei einer Vollbremsung der Abstand deutlich langsamer verringert.

Bei einem Einsatz im Außenbereich haben CMOS-Kameras durch ihren großen Dynamikbereich für die Helligkeit und ihre Stabilität gegenüber Blendung wesentliche Vorteile im Vergleich zu CCD-Kameras.

Im Abschnitt 4.2.3 wird die erforderliche Auflösung der Kameras diskutiert. Aufgrund des Ab-

tasttheorems ergibt sich für eine Objektdetektion und Positionsvermessung, sowie für die Spuruordnung eine erforderliche Auflösung von 10 cm in X-Richtung. Die Auflösung der Z-Richtung (der Entfernung) sollte für eine möglichst exakte Geschwindigkeitsbestimmung unterhalb von 5% relativem Fehler [39] bleiben. Die Höhe eines Objektes ist nur von geringer Relevanz.

Für die geforderte Genauigkeit in Z-Richtung sollte die Auflösung der Bildzeilen mindestens 1024 Pixel betragen [88]. Die Spaltenauflösung kann hingegen relativ gering sein. Farbe ist bei der Erfassung von Fahrzeugen eher hinderlich, da ein einfacher Algorithmus durch die Vielzahl der Farbinformationen gestört wird. Komplexere Algorithmen benötigen aber mehr Rechenleistung.

Es wurde ein Messsystem mit CMOS-Kameras mit einer Auflösung von 1024x1024 verwendet. Die Basisbreite wurde auf 70 cm festgesetzt.

Bei der Ermittlung der Positionen und somit Differenzgeschwindigkeiten von Fahrzeugen ist es hilfreich, dass nur Fahrzeuge, die sich annähern, als Gefahrenpotentiale in Betracht gezogen werden müssen. Somit können alle anderen Objekte während der Clusterung von der weiteren Bearbeitung ausgeschlossen werden.

### 4.2.2 Objektdetektion im Nahbereich

Kamerabasierte Komfort- und Sicherheitssysteme sind auch für den Nahbereich bei Fahrten in Innenstädten relevant. Die Geschwindigkeit des Egofahrzeugs liegt in Städten bei 50 - 60 km/h. Es gibt verschiedene Anwendungsbereiche. Neben einem Spurwechselassistent ist auch ein Kreuzungsassistent interessant. In Städten sind die Fahrzeuge zwar insgesamt langsamer, fahren jedoch mit einem geringeren Sicherheitsabstand. Weiterhin herrscht ein flexibleres Fahrverhalten. Aus Tabelle 4.1 kann entnommen werden, dass ein Fahrzeug mit 50 km/h ca. 0,55 m in 40 ms zurücklegt. Wird eine Zeit von 200 ms (5 Bilder bei 25 B/s) für die Detektion eines neuen Fahrzeugs angesetzt, so durchquert ein Fahrzeug mit 50 km/h in 200 ms ca. 3 m und im Gegenverkehrsfall werden aufgrund der Differenzgeschwindigkeit von 100 km/h ca. 6 m durchfahren. Daher ist mit einem Messbereich von 1 - 20 m eine Zeit von 5 - 10 Sekunden ausreichend. In diesem Bereich werden die Fahrzeuge sehr gut abgebildet.

Um einen Messbereich von 1 - 20 m realisieren zu können, sind ein größerer Öffnungswinkel und eine kürzere Basisbreite erforderlich. Bei einer Basisbreite von 5 cm und einem Öffnungswinkel von 90° wird in 1,5 m Abstand bereits eine volle Spurbreite und in 4,5 m eine dreispurige Straße erfasst. Die Positionsvermessung von Objekten im Umfeld sollte aufgrund des flexibleren Fahrverhaltens mit einem maximalen relativen Fehler von 2% realisiert werden. Aufgrund der sehr komplexen Fahrsituationen sind Fahrerassistenzsysteme für den Stadtverkehr gegenwärtig noch im Entwicklungsstadium.

Ein weiteres interessantes Anwendungsfeld für die Umfelderkennung liegt in der Robotik und bei fahrerlosen Transportfahrzeugen. Im Folgenden soll die Problematik bei fahrerlosen Transportfahrzeugen kurz diskutiert werden.

Fahrerlose Transportfahrzeuge übernehmen vorwiegend monotone Transportaufgaben in der Industrie und im Gesundheitswesen. In der Industrie bieten sich auch für Menschen unangenehme oder gefährliche Umgebungen als Einsatzort an. Die Fahrzeuge müssen in dieser Umgebung navigieren und auf Hindernisse reagieren. In Gebäuden bewegen sich die Fahrzeuge üblicherweise mit recht niedrigen Geschwindigkeiten im Bereich bis 3 km/h. Sollen Transportaufgaben im Außenbereich übernommen werden, sind häufig längere Strecken zu bewältigen. Somit muss das Fahrzeug schneller als 3 km/h fahren, um effektiv zu sein. Die Sicherheitssysteme müssen dann für eine höhere Geschwindigkeit ausgelegt sein.

Aufgabe von fahrerlosen Transportfahrzeugen ist häufig der Transport von Material auf Betriebsgeländen. Die Fahrzeuge navigieren mit Hilfe eines Systems auf der Basis von Magnet-Transpondern oder Leitdrähten in der Fahrbahn über das Gelände. Um eine Kollision mit herkömmlichen LKW oder Gefahrensituationen für Personen zu verhindern, ist ein System zur Hindernisdetektion montiert. Es besteht oft aus zwei Komponenten, einem Laserscanner, der berührungslos die Entfernung zu erhobenen Objekten bestimmt und einer opto-mechanischen Notabschaltung, welche das Fahrzeug bei Kontakt mit einem Gegenstand unverzüglich stoppt. Die zulässige Maximalgeschwindigkeit ist aufgrund der Notabschaltung auf 3 km/h begrenzt, da das Fahrzeug zum Stehen kommen muss, bevor eine eventuell gestürzte Person überrollt wird. Da die Notabschaltung gegenwärtig durch physischen Kontakt mit dem Hindernis ausgelöst wird, muss diese um den maximalen Bremsweg vorgelagert sein. Dies schränkt die maximale Geschwindigkeit stark ein.

Während im Straßenverkehr sowohl Messungen im Fernbereich als auch im Nahbereich relevant sind, ist bei der Umfelderkennung für autonome Fahrzeuge oder Roboter vorwiegend der Nahbereich interessant. Somit zählen großer Sichtbereich, eine hohe Genauigkeit und die Erfassung aller Objekte in der Umgebung des Fahrzeugs/Roboters zu den wichtigsten Eigenschaften. Je nach Zielsetzung sollen nur die Hindernisse in einem schmalen Fahrkorridor oder alle Objekte in der Umgebung zur Planung von Ausweichrouten bzw. zur Realisierung einer freien Navigation erkannt werden. Für sehr komplexe Systeme könnten die erfassten Objekte mit Hilfe von weiterführenden Bildverarbeitungsalgorithmen identifiziert werden. Beispiele wären neben der Erkennung von Menschen und beweglichen Objekten auch die Identifizierung von Schildern oder Landmarken im Umfeld.

Ähnlich dem Stadtverkehr ist der Bereich unmittelbar vor dem Fahrzeug wichtig. Mit einem Sichtbereich bis ca. 20 m kann das Fahrzeug auch aus höheren Geschwindigkeiten von 10 oder 15 km/h rechtzeitig abbremsen, ohne dass eine Beschädigung der Ladung zu erwarten ist. Da die LKWs auch Kurvenfahrten durchführen müssen, ist eine Sicht mit einem größeren Öffnungswinkel, z.B. 90° nötig. Jedoch soll an Engpässen die Wahrnehmung von Objekten neben der Fahrspur nicht zu einem Fehlalarm führen. Die Hinderniserkennung aus der Tiefenkarte an sich ist entsprechend aufwändiger, da in ihr hier alle Objekte berücksichtigt werden müssen.

Die Funktion der Hindernisdetektion reduziert die Messaufgaben hierbei auf das Erkennen und Vermessen der Position zum Objekt. Weiterhin muss insbesondere bei Kurvenfahrten festgestellt werden, ob das Objekt in den Fahrraum hineinreicht. Somit ist eine Messung mit einem relativen Fehler von ca. 2% in X- und Z-Richtung ausreichend.

Eine zweite Aufgabe ist die Umfelderkennung zur Navigation des Fahrzeugs. Dazu gehört die Umfahrung von Hindernissen. Größere Umbaumaßnahmen der Fahrstrecke können durch Anbringen von optischen Marken, die mit Kameras auswertbar sind, vermieden werden. Einen weiteren Anwendungsfall stellt das positionsgenaue Anfahren einer Laderampe mit einem LKW in einem komplizierten Gelände dar.

Für diese Aufgaben gelten erhöhte Anforderungen. Zur Navigation über ein Gelände muss die Position alle 100 - 200 ms ermittelt [30] werden. Bei geringeren Abtastraten kann es durch die zu große Totzeit zu Schwingungen kommen. Insbesondere für die Berechnung von Ausweichrouten oder die Navigation nach optischen Marken müssen alle relevanten Objekte und Personen ermittelt werden. Dies erfordert eine größere Genauigkeit als bei der Hindernisdetektion. In diesem Fall ist eine Genauigkeit der Positionsermittlung bis auf eine Winkeldifferenz von  $1^\circ$  notwendig. Dies entspricht ungefähr der einfachen Pixelgenauigkeit der Kamera.

Für beide Anwendungsfälle bei FTF ist eine Bildfrequenz von 10 Bildern/s ausreichend, da zum einen deutlich geringere Geschwindigkeiten auftreten, Objekte in einem Messbereich von 1 - 20 m sicherer detektiert werden können und weiterhin die Reaktionszeit eines Fahrers entfällt, da das Fahrzeug direkt gesteuert wird.

Da das Fahrzeug im Außenbereich arbeitet, ist es ständig widrigen Witterungsbedingungen ausgesetzt. So muss ein Sensor aus einer späteren Serienfertigung einem Temperaturbereich von  $-20^\circ\text{C}$  bis  $50^\circ\text{C}$  standhalten, bzw. gegen diese entsprechend geschützt sein. Weiterhin treten Regen, Schnee, Staub und Dampf auf, die sowohl die technische Funktion des Sensors als auch die prinzipielle Sicht beeinflussen können. Hier sind robuste Algorithmen und ein geeigneter Schutz des Sensors erforderlich.

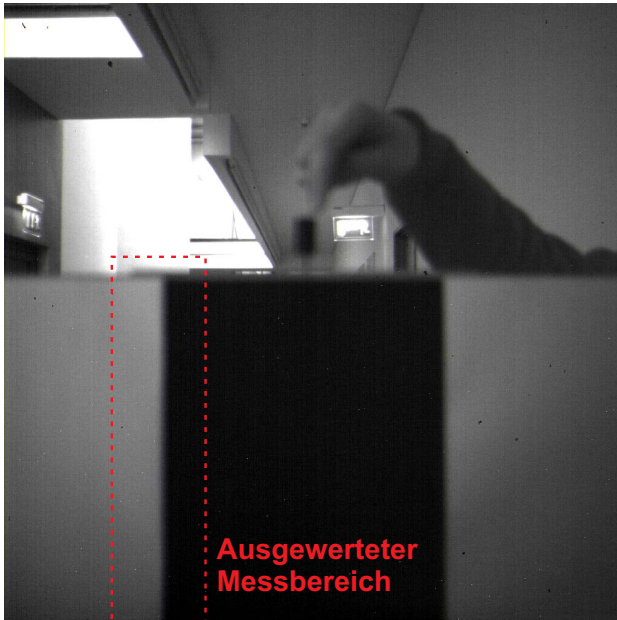
Aufgrund der hohen Sicherheitsanforderungen sind für den Betrieb von automatischen Fahrzeugen zwei unabhängige Messsysteme für die Hindernisdetektion erforderlich. Kamerasysteme können sowohl für die Umfelderkennung als auch für die Navigation zur Redundanz der Messung eingesetzt werden. Hierfür können die Bilddaten der Kameras an verschiedene Auswertesysteme verteilt werden.

### 4.2.2.1 Auswirkung der Brennweite eines Objektivs auf die Stereobildauswertung

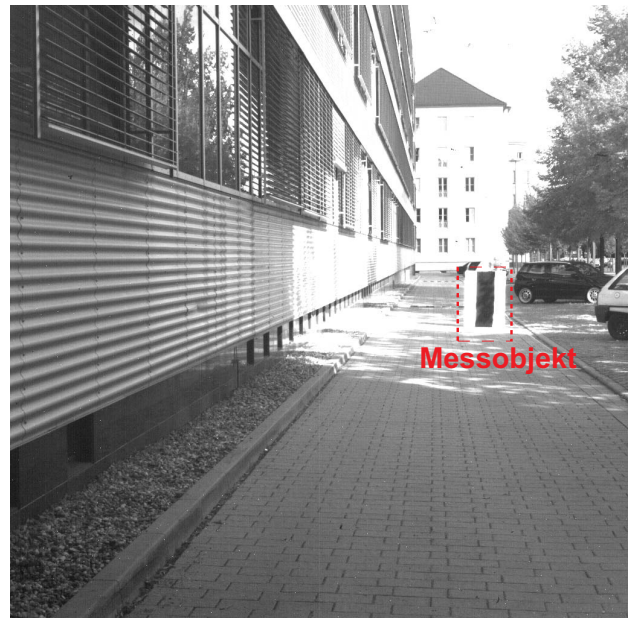
Bei der Vermessung im Nahbereich müssen bei der Umfelderkennung in der Regel deutlich größere Öffnungswinkel  $\alpha > 90^\circ$  für das Messsystem erreicht werden. Dies kann prinzipiell durch die Verwendung von Objektiven mit geringer Brennweite ( $f < 8\text{ mm}$ ) realisiert werden. Problematisch hierbei sind die durch die Linsenkrümmung entstehenden radiometrischen Verzeichnungen. Anhand der in dieser Arbeit verwendeten Kamerasysteme wird die Auswirkung der Linsenkrümmung und die Auswirkung der Rektifizierung des Bumblebee 2-Systems, welches im Abschnitt 2.4.5 näher beschrieben wurde, untersucht.

In Abbildung 4.1 sind vier Bilder mit unterschiedlichen Verzeichnungen dargestellt. In (a) ist ein Bild einer Photonfocus-Kamera mit einem 25 mm Objektiv bei einer Vermessung mit einer

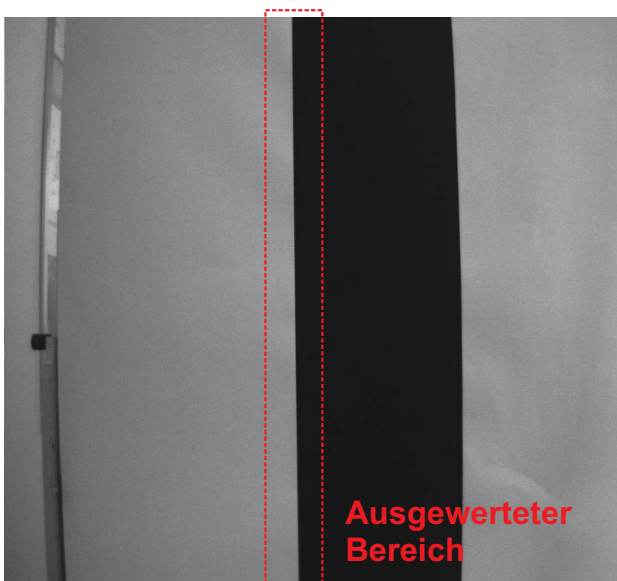




(a) *Photonfocus*:  $f = 12,5 \text{ mm}$



(b) *Photonfocus*:  $f = 25 \text{ mm}$



(c) *Bumblebee 2*:  $f = 6,3 \text{ mm}$  (Originalbild)



(d) *Bumblebee 2*:  $f = 6,3 \text{ mm}$  (rektifiziert)

Abbildung 4.1: optischer Vergleich der Aufnahmen des Standardmessobjektes mit verschiedenen Brennweiten

Messtafel dargestellt. Es zeigt nahezu keine Verzerrungen, abgesehen von den äußeren Rändern. Abbildung 4.1(a) zeigt ein Bild mit derselben Kamera mit einem Objektiv mit einer Brennweite von 12,5 mm, indem die Verzeichnungen an den Rändern und in den Ecken bereits deutlich werden, während im Zentrum das Bild noch als unverzerrt erscheint. Die Abbildungen 4.1(c) und (d) zeigen dasselbe Bild des Bumblebee Stereokamerasystems jeweils als Original und als rektifizierte Version. Während (c) auch im Zentrum eine deutlich sichtbare Verzeichnung aufweist und somit eine korrekte Messung sehr problematisch erscheint, sieht das rektifizierte Bild in (d) besser aus.

In Abbildung 4.2 ist der Verlauf der Disparität über die Zeilen dargestellt. Verwendet wurden hierfür die in Abbildung 4.1 dargestellten Bilder. Im Fall der Brennweite von 12,5 mm kann entlang des Objektes (siehe Abbildung 4.2(a)) nahezu keine Verzerrung festgestellt werden. Die Disparität variiert hier um ca. 0,5 Pixel. Daher wurde der Fall mit einer Brennweite von 25 mm nicht weiter betrachtet.

Den Verlauf der Disparität über die Zeilen bei den Bildern des Bumblebee 2 (siehe Abbildungen 4.2(b) und 4.2(c)) zeigt eine Variation von 6 Pixeln über die gesamte Höhe des Bildes sowohl im kalibrierten als auch im nicht kalibrierten Fall. Dies bedeutet, dass die Verzerrung der Linse bei der Kalibrierung nicht ausgeglichen wird. Das Bild wird im Wesentlichen gedreht und verschoben, sodass die Aufnahmen dem Normalfall der Stereophotogrammetrie entsprechen.

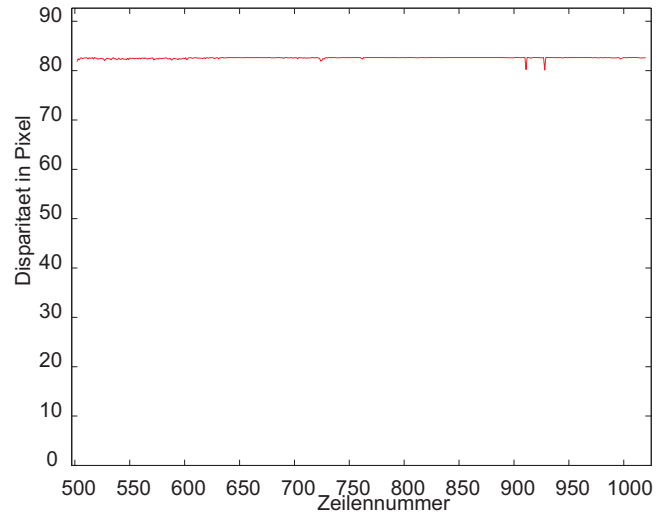
In jedem Fall ist für eine Messung mit hoher Präzision eine Berücksichtigung der Linsenverzerrungen bei Objektiven mit kurzer Brennweite erforderlich. Für Messsysteme, die mit einem eingebetteten System Daten von Stereobildern verarbeiten, sollten Linsen mit geringer Verzerrung verwendet werden, da eine Entzerrung eines Bildes eine aufwändige Rechenoperation ist. Die Untersuchung wurde auf der Basis von vorhandenem Bildmaterial durchgeführt und soll an dieser Stelle nur das Problem verdeutlichen.

### 4.2.3 Untersuchung der erforderlichen Auflösung in x- und z-Richtung

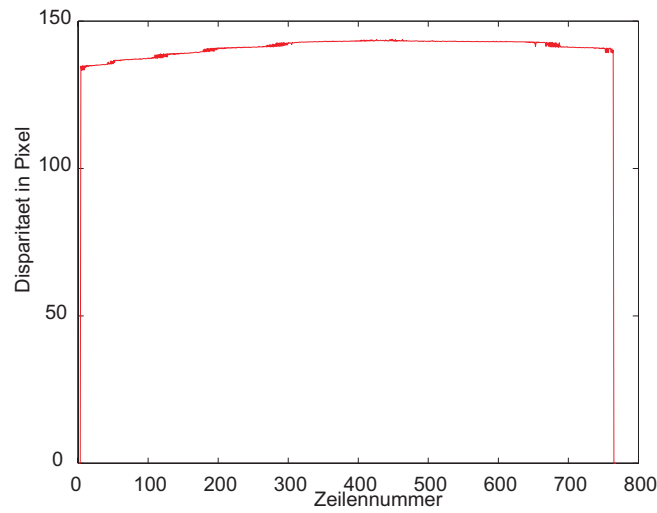
Zum Erreichen der Echtzeitanforderungen ist die erforderliche Auflösung der Tiefenkarte von entscheidender Bedeutung. Hierzu wurde die Veränderung der Abbildgröße bezüglich der Entfernung untersucht und Anforderungen bezüglich der Auflösung in X-Richtung ermittelt.

Für die Suche nach korrespondierenden Punkten werden in dieser Arbeit blockbasierte Verfahren verwendet, wie sie in Abschnitt 2.4.2.2 beschrieben werden. Hierbei wird ein Referenzblock aus einem Bild ausgeschnitten und mit verschiedenen gleich großen Blöcken eines anderen Bildes verglichen. Für die Auflösung der Tiefenkarte in dieser Arbeit ist die Anzahl der Referenzblöcke signifikant, da für jeden Referenzblock eine Disparitätssuche gestartet wird. In den meisten Verfahren besitzt der Block eine feste Größe und wird in einem regelmäßigen Raster aus dem Bild ausgeschnitten.

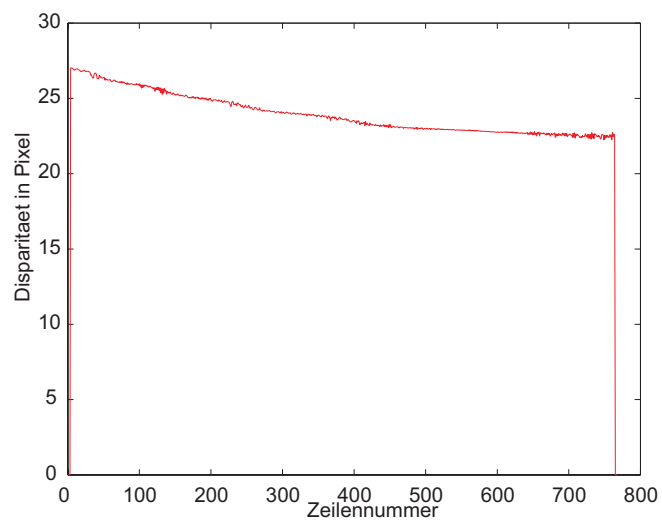
Betrachtet man die Größe der Abbildung des PKWs und der Person in Tabelle 4.2 ist erkennbar, dass mit steigender Entfernung die Abbildgröße asymptotisch gegen Null strebt. Die



(a) *Photonfokus  $f = 12,5 \text{ mm}$*



(b) *Bumblebee 2  $f = 6,3 \text{ mm}$  Originalbild*



(c) *Bumblebee 2 rektifiziert*

Abbildung 4.2: Vergleich der Variation der Disparität aufgrund optischer Einflüsse bei verschiedenen Brennweiten und Vorverarbeitungen

Entfernung [m]	PKW [Pixel]	Person [Pixel]
150	27	8
100	38	12
75	54	19
50	76	25
25	159	48
10	380	114

Tabelle 4.2: Größe verschiedener Messobjekte in unterschiedlicher Entfernung (Kamera: Photonfocus MV-D1024, 1024x1024 Pixel; Objektiv Linos MevisC 25 mm)

Abbildgröße wird durch den eingenommenen Winkel bestimmt.

Dadurch ergeben sich zwei Untersuchungsgegenstände: Die Häufigkeit der Abtastung eines Objektes und die Größe des Blockes im Vergleich zur Größe der Abbildung des vermessenen Objektes.

Abstrahiert man ein Objekt als einen Rechteckimpuls mit der Breite entsprechend der Pixelanzahl in der Abbildung und tastet diesen Rechteckimpuls ab, so ergibt sich über das Abtasttheorem eine Aussage, in welchen Abständen [Pixel] im Bild mit Hilfe von Referenzblöcken abgetastet werden muss, damit das Objekt registriert wird.

Das Abtasttheorem ist für die Abtastung von analogen Werten abgeleitet worden. Der Grauwertverlauf an sich ist bereits ein digitales, also abgetastetes Signal. Für die Erstellung der Tiefenkarte wird das Bild erneut abgetastet. Diese Abtastung erfolgt durch das Ausschneiden von Blöcken im Referenzbild. Diese Abtastung beschreibt jedoch nur, ob ein Objekt erkannt wird. Es wird keine Aussage über die Genauigkeit der Entfernungsmessung getroffen.

Wird die Existenz eines Objektes als relevante Information angenommen, kann der Übergang zur Abstraktion des Objektes durch einen Peak, beschrieben durch Breite und Höhe, erfolgen. Dementsprechend ergibt sich als notwendige Bedingung, dass ein Objekt in x-Richtung mit mindestens zwei Referenzblöcken vom System abgetastet werden muss.

Wenn ein Objekt in einem Referenzblock abgebildet wird, jedoch nicht die dominierende Information in diesem Block darstellt, dann wird das entsprechende Objekt nicht korrekt vermessen. Dies resultiert aus der Betrachtung des besten Extremums bei der blockbasierten Korrelation. Würden auch Nebenmaxima für die Auswertung herangezogen werden, könnte das Objekt auch mit Blöcken, in denen es weniger als 50% des Blockes belegt, vermessen werden. Dies erfordert jedoch aufwändige Vergleichs und Iterationskriterien, wie die in Abschnitt 2.4.4 beschriebenen Verfahren zur Ermittlung dichter Tiefenkarten zeigen. Da in dieser Arbeit aus Gründen des Rechenaufwandes im Wesentlichen einfache blockbasierte Korrelationsverfahren verwendet werden, sind Blöcke, in denen das Objekt garantiert abgebildet wird, nur solche Blöcke, in denen das Objekt mehr als 50% der Information darstellt. Dies ist die hinreichende Bedingung für die Erkennung eines Objektes. Ist der Informationsgehalt in der Umgebung sehr niedrig, so kann auch ein deutlich geringerer Prozentsatz ausreichend sein. Dieser Fall kann jedoch nicht

als allgemeingültig angenommen werden.

Die Blockgröße muss sich nach der kleinsten Abbildung des kleinsten relevanten Objektes im Messbereich richten. Durch die der Entfernung indirekt proportionale Abbildgröße entsteht die kleinste Abbildung in der größten Messentfernung. Bezüglich des Beispiels aus Tabelle 4.2 bedeutet dies, dass durch einen Block von 16x1 Pixeln, der alle 16 Pixel ausgeschnitten wird, ein PKW in 150 m Entfernung in 68,75 % der möglichen Blockpositionen sicher in der Tiefenkarte abgebildet wird. In den anderen Fällen liegt ein Block weniger als 50% auf dem Objekt. Dies setzt voraus, dass die Informationen in den Blöcken eindeutig sind. Ist das Objekt sehr homogen oder liegt eine Textur die mehrdeutig ist, auf dem Objekt, so existiert in diesen Fällen keine eindeutige Abbildung des Objektes. Die Objektkanten stellen in der Regel eindeutige Abbildungen dar.

Wird die Häufigkeit der Abtastung verdoppelt und somit alle 8 Pixel ein Referenzblock über das Objekt gelegt, so liegen in 100 % der möglichen Blockpositionen 2 Referenzblöcke in x-Richtung auf der Abbildung des PKW in 150 m Entfernung. Dies konnte durch die Messungen in Abschnitt 6.1 verifiziert werden. Bei einer Abtastrate der Referenzblöcke von 128 Referenzblöcken/Zeile konnte das Objekt auch oberhalb von 130 m mit 5 - 10 3D-Punkten detektiert werden. Mit einer Abtastrate der Referenzblöcke von 16 Pixeln hingegen war das Objekt oberhalb von 130 m nur zufällig zu detektieren. In den meisten Bildern lagen auf dem Objekt nur 1 oder 2 valide 3D-Punkte, mit denen ein Objekt nicht sicher detektiert werden kann.

Betrachtet man weiterhin die Hindernisdetektion für FTF, bei denen die kleinsten relevanten Objekte durch Menschen repräsentiert werden, so würde eine Abtastung alle 16 Pixel ausreichen, wenn der Oberkörper einer Person als wesentliches Merkmal angenommen wird. Für fahrerlose Transportfahrzeuge wird jedoch als minimal wahrnehmbares Objekt für ein Sicherheitssystem bei FTF ein menschlicher Unterschenkel festgelegt. Unter dieser Maßgabe sind auch hier Referenzblöcke mit einer Abtastrate von 128 Referenzblöcken/Zeile erforderlich.

#### 4.2.3.1 Echtzeitbedingungen und Messbereiche

Nach Analyse der beiden Einsatzszenarien zeigen sich Ähnlichkeiten in den Anforderungen an die Auswertung hinsichtlich der zu ermittelnden Informationen. Die Bestimmung der eigenen Position, die Detektion von Hindernissen und die Auswertung optischer Merkmale in der Szene, ebenso wie die Verwendung eines passiven Messsystems und letztendlich auch die zeitlichen Randbedingungen zeigen, dass beide Probleme mit einem ähnlichen Sensorsystem gelöst werden können. Bei den folgenden Annahmen werden identische Kameras mit unterschiedlichen Objektiven für Nah- und Fernbereich vorausgesetzt. In der Tabelle 4.3 sind die Messbereiche und erforderlichen Bildfrequenzen und damit die Echtzeitforderungen der oben diskutierten Szenarien für den Nahbereich und den Fernbereich dargestellt.

Die Echtzeitbedingungen beschreiben in diesem Fall die kontinuierliche Verarbeitung von Bildern. Für die Verarbeitung eines Bildes bleibt genau die Zeit, die für die Aufnahme des nächsten Bildes benötigt wird. Nur durch Einhaltung dieser Bedingung kann eine kontinuierliche Stereo-

	Nahbereich		Fernbereich	
	Hindernisdetek- tion für FTF	Frontsicht bei Stadtfahrten	Spurwechsel- assistent	Frontsicht bei Fahrten auf Landstraßen
Messbereich	1-20 m	1-20 m	10-150 m	10-150 m
Bildfrequenz	10 Bilder/s	25 Bilder/s		25 Bilder/s (optimal 50 Bilder/s)
Geschwindigkeit Egofahrzeug	3-5 km/h	50 km/h	bis 250 km/h	bis 120 km/h
Differenz-ge- schwindigkeit	ca. 20 km/h	ca. 100 km/h	bis 250 km/h	
Öffnungswinkel	$\geq 90^\circ$		30° bis 60°	
max. relati- ver Fehler in X-Richtung	2-5 %		10 %	
max. relati- ver Fehler in Z-Richtung	2 %		5 %	

Tabelle 4.3: Vergleich der Echtzeitanforderungen und Messbereiche der Szenarien

bildauswertung garantiert werden.

Kann die Auswertung parallel zur Bildaufnahme für eine Bildfrequenz von 25 Bildern/s realisiert werden, können prinzipiell alle Szenarien realisiert werden. Die wesentlichen Unterschiede zwischen dem Nah- und dem Fernbereich liegen in den Messbereichen, den Öffnungswinkeln sowie den Genauigkeitsanforderungen für die Vermessung der X- und Z-Koordinaten.

Mit einem System aus zwei Kameras mit 1024x1024 Pixeln können die Genauigkeitsanforderungen durch Variation der Objektive und der Anordnung der Kameras realisiert werden. Für die Vermessung im Fernbereich ist ein Kamerasystem im Normalfall der Stereophotogrammetrie mit einer Basisbreite von etwa 70 cm erforderlich. Der Öffnungswinkel der Objektive bestimmt auch den Öffnungswinkel des Messsystems. Bei einem Objektiv mit 25 mm Brennweite erhält man einen Öffnungswinkel von ca. 60°.

Für die Vermessung im Nahbereich kann ein Öffnungswinkel von ca. 90° mit Hilfe von Objektiven mit einer Brennweite von 12,5 mm erreicht werden. Die Basisbreite kann hierbei aufgrund des geringeren Messbereiches auf ca. 5 cm verringert werden.

### 4.3 Rechenaufwand von Algorithmen zur Stereobildauswertung

Beide Szenarien erfordern, wie in den Abschnitten 4.2.1 und 4.2.2 erläutert, eine Stereobildauswertung mit 25 Bildern/s. In Zukunft kann jedoch durch veränderte Szenarien schnell eine deutlich höhere Bildfrequenz erforderlich sein, wie der Ausblick in Abschnitt 4.2.1 bezüglich einer Situationserkennung für die Frontsicht auf der Landstraße zeigt.

Um eine Aussage über den Rechenaufwand für Algorithmen der Stereobildauswertung treffen zu können, müssen diese bezüglich ihrer Struktur und der erforderlichen arithmetischen Operationen untersucht werden. Eine konkrete Aussage über den Rechenaufwand einer Operation ohne Berücksichtigung einer konkreten Implementierungsplattform ist sehr schwierig.

In der Informatik wird zur Analyse der Komplexität häufig eine Turing-Maschine verwendet. Eine Turing-Maschine ist ein Automat, der eine prozessororientierte Plattform anhand einer Rechenmaschine darstellt, die aus einem Speicherband und einem Lese/Schreibkopf aufgebaut ist. Bei den durchgeführten Untersuchungen geht es im Allgemeinen um die Berechenbarkeit bzw. um die prinzipielle Lösbarkeit von algorithmischen Problemen.

Bezüglich der in Kapitel 2 vorgestellten Algorithmen kann angenommen werden, dass sie implementierbar sind. Die Lösbarkeit hängt immer von den konkreten Daten ab. Daher ergeben sich durch die Betrachtung mit einer Turing-Maschine nur bedingt neue Informationen.

Da eine Untersuchung der Implementierung für einen speziellen Prozessor aufgrund der starken Unterschiede ebensowenig charakteristische Aussagekraft hat, wurden die Untersuchungen bezüglich des Rechenaufwandes der Stereobildauswertung sowohl auf DSPs als auch anhand einer Hardwareimplementierung durchgeführt.

Aussagen bezüglich des Rechenaufwandes sind auf Basis einer Datenfluss- und Kontrollflussanalyse möglich. Da die Korrelationskriterien nahezu ausschließlich Datenflussabhängigkeiten aufweisen, wird ihr Rechenaufwand anhand einer angenommenen FPGA-Implementierung bestimmt. Hardwareimplementierungen werden ähnlich wie Turing-Maschinen und verschiedene anderer endlicher Automaten zur Bestimmung des Rechenaufwandes verwendet.

#### 4.3.1 Korrelationskriterien in Softwareimplementierungen

Der Rechenaufwand in Software wird entscheidend von der Architektur des Prozessors bestimmt. Da die Mikroprozessoren, die für eingebettete Systeme eingesetzt werden, vorwiegend auf steuerungstechnische Aufgaben optimiert sind, werden die für die Signalverarbeitung optimierten DSP zum Vergleich verwendet. Einige Mikroprozessoren für eingebettete Systeme, wie z.B. der auf ARM-Technologie basierende Intel Xscale, haben durch Anpassung der Architektur eine den DSPs vergleichbare Leistungsfähigkeit.

Der Vergleich [47] wird durch eine Implementierung der SADMF-Funktion und der QNKKFMF



Bildauffösung [Pixel]	Festkomma-DSP BF533 (600 MHz)			Gleitkomma TS201S 500MHz		
	1024x1024	Zyklen je Zeile	512x384	1024x1024	Zyklen je Zeile	512x384
Disparitäts- bereich [Pixel]	256		128	256		128
SADMF	0,5	97192	4-5	1,2	375691	10
SADMF DSP-opt.	1	497341	10	-	-	-
QNKKFMF	-	-	-	0,25	937755	2

Tabelle 4.4: Übersicht der erreichbaren Bildfrequenzen [Bildern/s] und des *Rechenaufwandes* [Zyklen] von Korrelationskriterien in Software in DPSs

jeweils auf einem Festkomma-DSP (Blackfin von Analog Devices) und einem Gleitkomma-DSP (TigerShark von Analog Devices) vorgenommen. Dazu wurden beide Funktionen auf beiden Prozessoren implementiert und ein ausführlicher Vergleich verschiedener Implementierungsoptimierungen durchgeführt.

Die Implementierungen wurden durch verschiedene Möglichkeiten bezüglich ihrer Leistungsfähigkeit optimiert. Dabei wurde zunächst ein einfaches korrelationsbasiertes Verfahren verfolgt. Im Vergleich benötigt die QNKKFMF in der Festkommaimplementierung 7-mal soviel Rechenleistung wie das SADMF-Kriterium. Bei einer Gleitkommarealisierung kann durch die bessere Anpassung des Wertebereichs an die Notwendigkeiten der QNKKFMF der Mehraufwand auf das Fünffache des SADMF-Kriteriums bei Gleitkommaimplementierung reduziert werden. Bei diesem Vergleich wurden 16 Pixel durch einen Block erfasst.

Die erreichbaren Geschwindigkeiten lagen im Bereich von 0,5 Bildern/s ohne speziell optimierte DSP-Operationen und 1 Bild/s mit optimierten DSP-Operationen bei einer Bildauflösung von 1024x1024 Pixeln und einer maximalen Disparität von 256 Pixeln auf Basis der SADMF-Funktion. Mit kleineren Bildern und geringerem Suchbereich für die Disparität (512x384x128 Pixel) könnte eine Bildwiederholrate von 10 Bildern/s erreicht werden. Die QNKKFMF ist auf einem Festkomma-DSP nur bedingt echtzeitfähig implementierbar.

Auf dem 600 MHz schnellen Gleitkomma-DSP TigerShark können sowohl die SADMF als auch die QNKKFMF mit einer höheren Geschwindigkeit implementiert werden. Dies liegt zum einen an der höheren Taktfrequenz und zum anderen an dem größeren Parallelisierungsgrad, der durch die deutlich komplexere Architektur des TigerShark erreicht werden kann. In beiden Fällen wurden keine speziell optimierten DSP-Funktionen verwendet. Hierbei konnte für die SADMF-Funktion eine Bildfrequenz von 1,2 Bildern und die QNKKFMF eine Bildfrequenz von 0,25 Bildern/s bei 1024x1024 Pixel und 256 Pixel Disparität ermittelt werden. Wird wieder die reduzierte Bildgröße von 512x384 Pixel mit 128 Pixel Disparität angesetzt, so kann ebenfalls eine Geschwindigkeit von 10 Bildern/s für die SAD-Funktion und von 2 Bildern/s für die QNKKFMF erreicht werden.



### 4.3.2 Korrelationskriterien in Hardwareimplementierungen

In Hardware ergeben sich zwei Varianten für die Implementierung eines Korrelationskriteriums. Die parallele Implementierung ist für maximalen Datendurchsatz ausgelegt und benötigt daher auch sehr viele Ressourcen, während die sequentielle Implementierung auf geringen Ressourcenverbrauch optimiert ist und daher einen deutlich höheren Arbeitstakt erfordert, um eine vergleichbare Datenrate mit der parallelen Implementierung zu erreichen. Die sequentielle Implementierung ähnelt der Verarbeitung in Software.

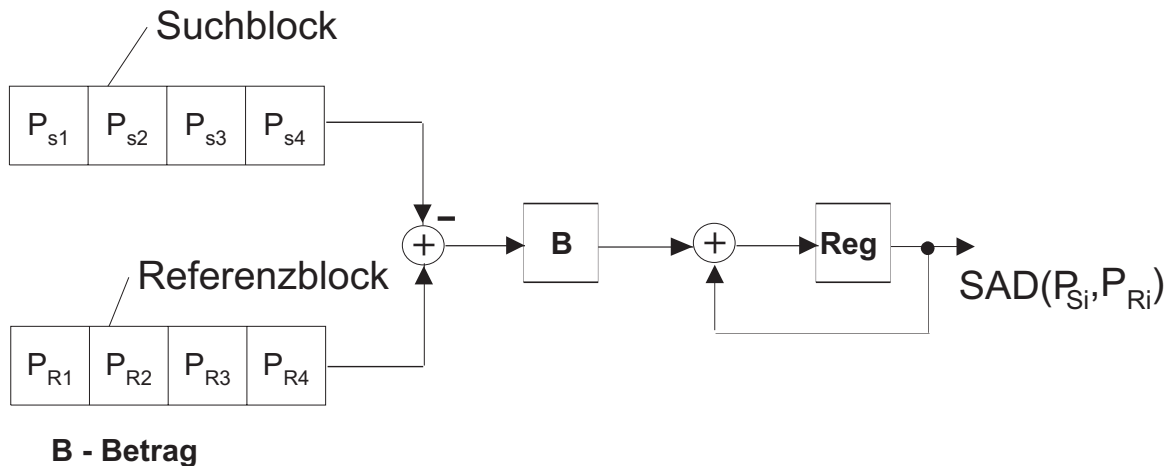


Abbildung 4.3: Sequentielle Implementierung der SAD

In Abbildung 4.3 ist die sequentielle Verarbeitung der SAD-Funktion für eine Blockgröße von vier Pixeln abgebildet. Sie realisiert das in Gleichung 2.6 angegebene Korrelationskriterium. Dabei wird nach der Subtraktion der Grauwerte der Pixel der Betrag (durch B dargestellt) der Differenz gebildet und diese anschließend akkumuliert.

Eine sequentielle Implementierung wird verwendet, wenn die Signale mit geringer Datenrate eintreffen und ein geringer Ressourcenverbrauch entscheidend ist. Die Folge der arithmetischen Operationen wird nur einmal in einer Pipelinestruktur implementiert. Daher muss zur Berechnung der Korrelationsfunktion eines Blockes jede Blockkombination nacheinander berechnet werden. Bei einem Pixeltakt von 28 MHz, einer Bildauflösung von 1024x1024 Pixeln, 64 Blöcken je Zeile und einer maximalen Disparität von 256 Pixeln ergibt sich somit ein erforderlicher Takt von 1,792 GHz für eine blockbasierte Korrelation. Diese Geschwindigkeit kann in einem FPGA gegenwärtig nicht realisiert werden, daher wird auf die weitere Betrachtung von sequentiellen Implementierungen an dieser Stelle verzichtet. In Tabelle 4.5 sind zum Vergleich einige Implementierungsvarianten mit erreichbaren Geschwindigkeiten gegenübergestellt.

Bei der parallelen Implementierung werden die arithmetischen Operationen für eine Blockkombination jeweils gleichzeitig berechnet. Daher erhöht sich die Anzahl der Takte bei der Änderung der Fenstergröße nicht. Jedoch erhöht sich der Bedarf an Logikelementen mit der Steigerung der Fenstergröße entsprechend. Die parallele Verarbeitung kann nur in einer Hardwareimplementierung adäquat umgesetzt werden. Bei der parallelen Implementierung hebt sich auch die

Bitbreite der Pixel		8 Bit			12 Bit			16 Bit		
		LE	Perf. [MHz]	Pipel.	LE	Perf. [MHz]	Pipel.	LE	Perf. [MHz]	Pipel.
seq.	SAD	168	220	4	250	220	4	328	190	4
	SADMF	1270	184	21	1480	181	21	1686	181	21
	SSD	226	149	6	388	140	6	562	138	9
	SSDMF	1328	131	23	1618	124	23	1920	121	26
	QNKKFMF	4675	114	92	5423	98	108	6395	60	122
par.	SAD	1534	195	6	1726	190	6	1918	190	6
	SADMF	3738	184	23	4186	181	23	4634	181	23
	SSD	4728	149	9	5118	140	11	5502	138	16
	SSDMF	6932	131	26	7578	124	29	8218	121	35
	QNKKFMF	17672	114	92	20502	98	108	28914	60	122

Tabelle 4.5: Vergleich des Hardwareaufwands verschiedener Korrelationskriterien auf einem Altera Stratix EP1S60 (LE - Logikelemente; Perf.- Performance; Pipel. - Pipelinestufen)

SSD deutlich von der SAD ab. Die SAD-Funktion ist sowohl in einer Software- als auch in einer Hardwareimplementierung das, unter den betrachteten, am günstigsten zu implementierende Korrelationskriterium.

Die schematische Darstellung der SAD in Abbildung 4.4 verdeutlicht die Parallelisierung der Verarbeitung im Vergleich zur sequentiellen Verarbeitung in Abbildung 4.3. Hierbei werden gleichzeitig die Differenzen der Grauwerte aller Pixel berechnet und anschließend der Betrag gebildet. Durch Summation der parallel vorliegenden absoluten Differenzen erhält man die SAD.

Im Anhang B.2 werden die Korrelationskriterien bezüglich ihrer Struktur untersucht und entsprechend des Rechenaufwands für verschiedene Bitbreiten und Fenstergrößen untersucht. Die Bitbreiten wurden in Abhängigkeit von unterschiedlichen Aufnahmesystemen gewählt. Während die Industriekameras mit der größten Verbreitung derzeit immer noch auf Basis von 8 Bit Grauwerten bzw. 8 Bit je Farbkanal arbeiten, werden Kameras mit einem hohen Dynamikumfang, vorwiegend auf APS(CMOS)-Sensoren basierend, mit 10 bis 12 Bit Grauwertaufösung hergestellt. Systeme für Aufnahmen mit High Dynamik Range (HDR) liefern in der Regel 16 Bit je Pixel. In Abbildung 4.7 ist qualitativ die Abhängigkeit des Ressourcenverbrauchs der Korrelationskriterien von der Bitbreite im Bereich von 8 Bit bis 24 Bit dargestellt.

Hierbei ergibt sich für die SAD-Funktion sowie die SADMF-Funktion eine lineare Abhängigkeit von der Bitbreite, da nur Additionen und Subtraktionen zur Berechnung erforderlich sind. Bei der SSD und der SSDMF lässt sich eine leichte quadratische Abhängigkeit erkennen. Dies resultiert aus der in der SSD enthaltenen Multiplikation. Wird eine auf Datendurchsatz optimierte Implementierung der Multiplikation verwendet, so steigt der Bedarf an Logikeinheiten quadratisch [61] in Abhängigkeit von der Bitbreite. Mit größerer Fensterbreite steigt der Anteil der quadratischen Abhängigkeit von der Bitbreite, da mehr Multiplizierer parallel implementiert

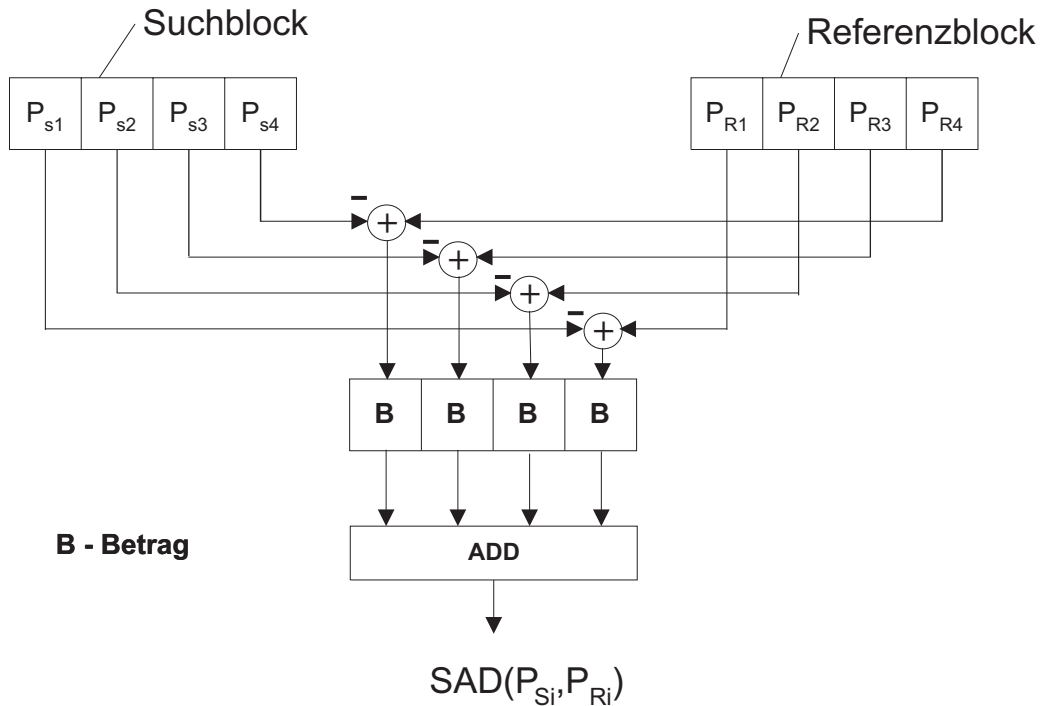


Abbildung 4.4: Parallele Implementierung der SAD

werden.

Der größte Ressourcenaufwand von den betrachteten Funktionen entsteht durch die QNKKFMF. Dies resultiert vor allem aus der für die Normierung erforderlichen Division. Eine Implementierung der Division ist nur mit einem sehr hohen Aufwand möglich, da sie durch sukzessives Berechnen für jedes Bit ermittelt wird. Wird dies für einen hohen Datendurchsatz mit einem starken Pipelining implementiert, ergibt sich eine starke quadratische Abhängigkeit von der Bitbreite. Der zweite Aspekt ist die parallele Berechnung des Zählers und des Nenners der Division. Beide Terme enthalten Multiplikationen und werden parallel implementiert.

Für die Implementierung der Wurzelfunktion werden in Mayer-Bäse [61] und in Tornow [88] verschiedene Ansätze diskutiert. Zum einen kann eine Look-Up-Table eingerichtet und zwischen den Werten interpoliert werden oder es kann die Funktion in einer Taylorreihe entwickelt werden und so implementiert werden. Beide Verfahren sind sehr aufwändig. Da über die Korrelation nur die Position des Maximums der NKKFMF berechnet werden soll, kann durch eine Quadrierung der NKKFMF zur QNKKFMF (Gleichung 5.4) die Berechnung der Wurzel vollständig umgangen werden.

Bei der parallelen Implementierung der QNKKFMF (siehe Abbildung 4.5) werden Zähler und Nenner gleichzeitig berechnet. Die Pixel der beiden Blöcke werden für den Zähler miteinander multipliziert und für den Nenner jeweils quadriert (durch  $Q$  dargestellt) und anschließend getrennt aufsummiert. Abschließend erfolgt die Division des quadrierten Zählers und des Produktes im Nenner.

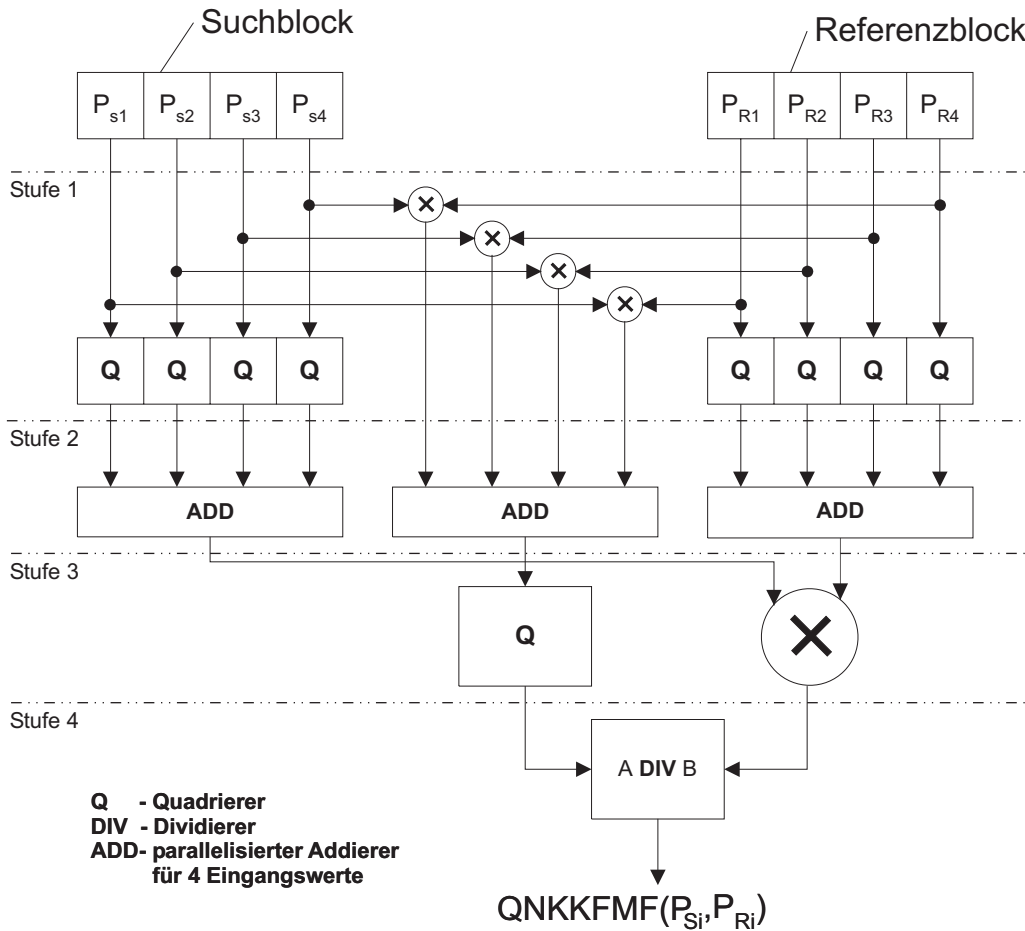


Abbildung 4.5: Parallele Implementierung der QNKKFMF

Die Quadrierung hat noch einen weiteren Vorteil. Bei einer Realisierung mit der Wurzelfunktion wäre die Berechnung des Zählers zwei Operationen vor der Berechnung des Nenners abgeschlossen. Da sowohl die verbleibende Multiplikation im Nenner als auch die Ermittlung der Wurzelfunktion mehrere Takte benötigen, müsste das Ergebnis mit entsprechend vielen Registern gespeichert werden, um einen hohen Datendurchsatz mit massivem Pipelining zu erreichen. Durch die Quadrierung der Funktion wird eine Operation vom Nenner in den Zähler verschoben. Da in beiden Fällen eine Multiplikation erfolgt, entsteht eine Symmetrie bei der Verarbeitung von Zähler und Nenner.

Weiterhin wird der Rechenaufwand von der Größe des Blockes stark beeinflusst. Je größer der Block, umso mehr Operationen müssen parallel ausgeführt werden. Bei eindimensionalen Fenstern steigt der Bedarf kontinuierlich mit jedem hinzugefügten Pixel. Wird ein 2D-Block gewählt, steigt die Anzahl der Pixel im Block stark an. Die Form des Blockes hat dabei auf den Rechenaufwand der Implementierung keinen wesentlichen Einfluss, daher werden ausschließlich die erforderlichen Pixel des Blockes angegeben und die Abhängigkeit von der Blockgröße bezüglich eines eindimensionalen Fensters beschrieben.

Bei der Betrachtung des Anstiegs des Rechenaufwandes in Abbildung 4.6 fällt auf, dass die

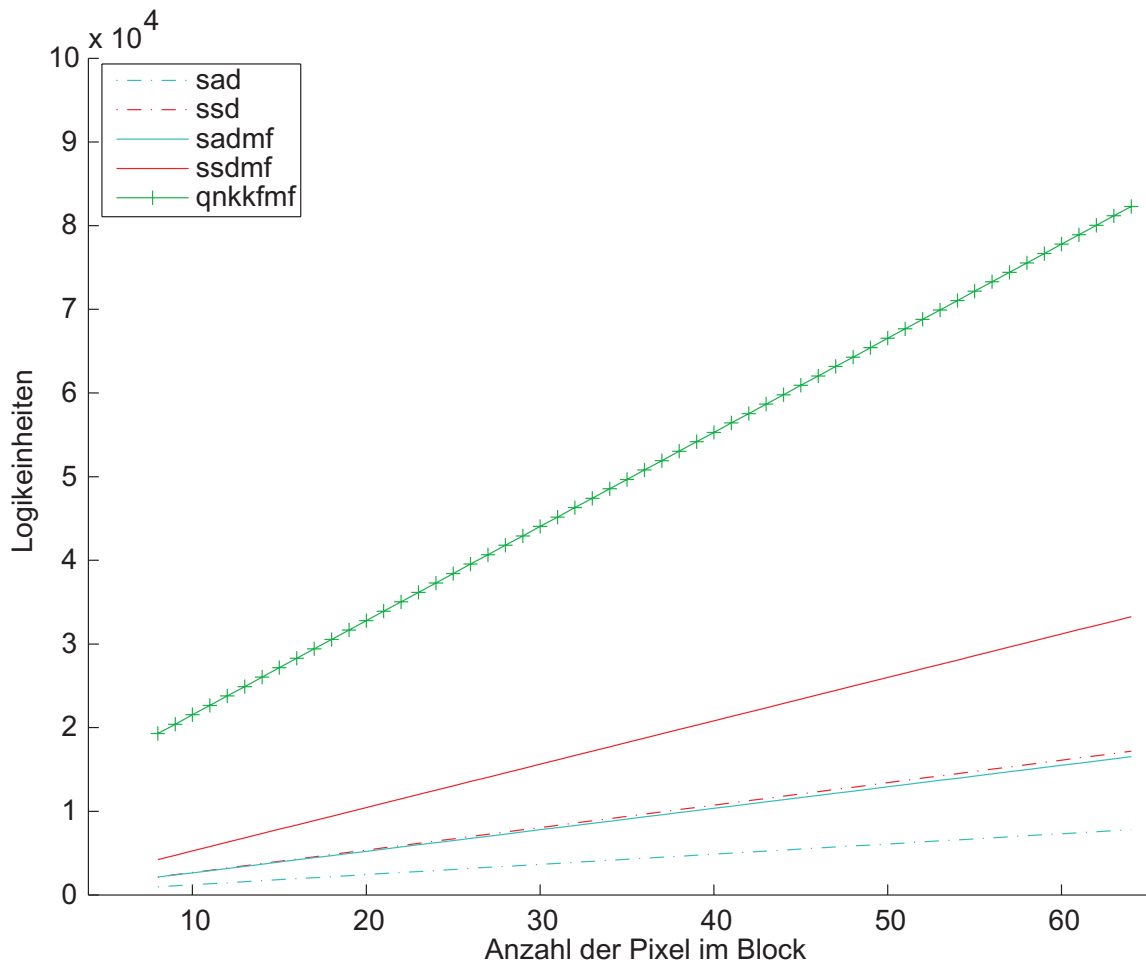


Abbildung 4.6: Rechenaufwand der parallelen Implementierung von Korrelationskriterien bei der Variation der Blockgröße (8 - 64 Pixel)

Abhängigkeit von der Fenstergröße bei allen Funktionen nahezu linear ist. Mit jedem neuen Pixel ergibt sich ein vom Verfahren abhängiger konstanter Betrag an Logikelementen, der den Rechenaufwand erhöht.

Problematisch ist hierbei die Berücksichtigung der mittelwertfreien Pixel. Zur Berechnung des Mittelwertes wird die Division durch eine Shift-Operation ersetzt. Diese kann jedoch nur bei Divisionen durch  $2^i; i \in \mathbb{R}$  erfolgen. Für Fensterbreiten, die nicht in diesem Raster liegen, müssen die Mittelwerte durch die sukzessive Rückführung auf eine Division durch  $2^i$  zurückgeführt werden. Diese Operationen erzeugen ein komplexes Timing bei der Erstellung der mittelwertfreien Pixel.

Für die weitere Verarbeitung wurden die SADMF und die QNKKFMF ausgewählt. Die Auswahl der QNKKFMF erfolgte aufgrund ihrer Stabilität auch bei verrauschten Bildern [9]. Die SSD und die SAD, sowie die mittelwertfreien Varianten sind in ihrem Verhalten bei der Verarbeitung von Bildern sehr ähnlich (siehe Anhang B). Der Rechenaufwand der SAD ist gegenüber der SSD-Funktion in Implementierungen in Software und in Hardware deutlich geringer. Für eine

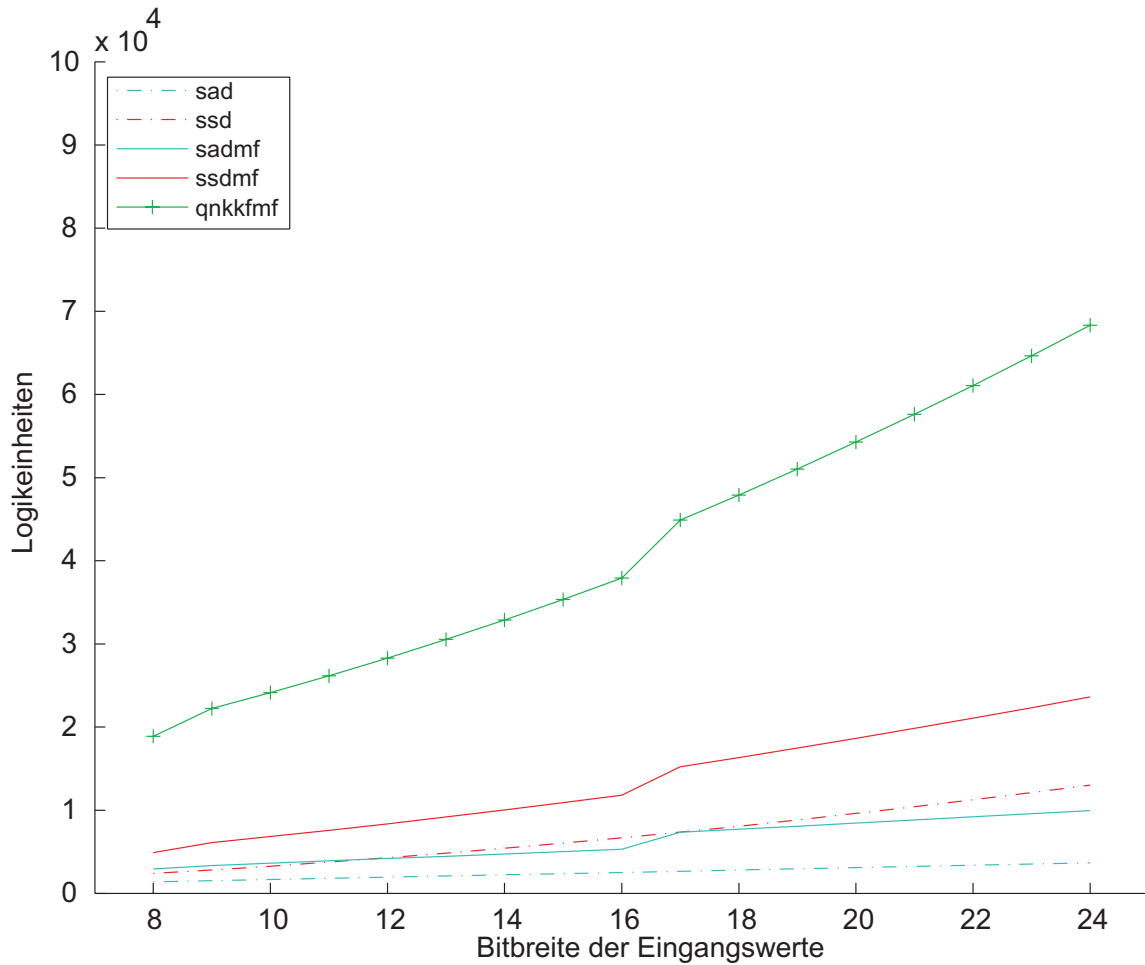


Abbildung 4.7: Rechenaufwand der parallelen Implementierung von Korrelationskriterien in Abhängigkeit von der Bitbreite im Bereich (8 - 24 Bit) der Eingangswerte (qualitativ)

Verarbeitung von Bildern mit wechselnden Beleuchtungsverhältnissen ist eine mittelwertfreie Verarbeitung erforderlich, wenn eine schnelle Helligkeitsregelung nicht realisiert werden kann.

### 4.3.3 Vergleich des Rechenaufwandes auf verschiedenen Plattformen

Nachdem nun unterschiedliche Korrelationskriterien und unterschiedliche Plattformen untersucht wurden, kann festgehalten werden, dass die SADMF-Funktion aufgrund ihrer einfacheren Operationen am ehesten auf eingebetteten Plattformen implementierbar ist. Die QNKKFMF ist deutlich aufwändiger, hat jedoch in einigen Aufnahmesituationen durchaus Vorteile gegenüber der SADMF.

Für eingebettete Systeme im Echtzeitbetrieb ist eine Implementierung der QNKKFMF gegenwärtig nur in Hardware sinnvoll realisierbar. Auf Prozessoren sind sowohl die Division als auch die erforderliche doppelte Genauigkeit bei einer Reihe von Gleitkommaberechnungen hinderlich.

Ebenso ist die Implementierung auf einem Prozessor, der ausschließlich über Festkommaarithmetik verfügt, für die diskutierten Anwendungsfälle nicht in Echtzeit realisierbar.

Auf einer Hardwareplattform sind sowohl die SADMF als auch die QNKKFMF in Echtzeit realisierbar. Hierbei ist auf die durchgängige Parallelisierung zu achten, damit mittels eines minimalen Taktes die maximale Geschwindigkeit erreicht werden kann. Bei einer parallelen Implementierung liegt der Rechenaufwand der QNKKFMF bei dem ca. 16-fachen des Rechenaufwandes für die SADMF. Dies ist begründet in den erforderlichen Multiplikationen, welche an sich bereits aufwändiger sind als Additionen und durch den zusätzlichen Aufwand zur Berechnung des Nenners sowie der Division.

Ein weiterer Vorteil der Hardwareplattform ist die Möglichkeit der Mehrfachinstanziierung. Dabei wird eine Hardwarestruktur, wie z.B. die SADMF oder die QNKKFMF, in Teilen oder vollständig mehrfach parallel implementiert. Hierbei muss beachtet werden, dass dadurch der erforderliche Hardwareaufwand steigt. Die erreichbare Geschwindigkeit reduziert sich aber nicht, sofern ein Engpass bei der Zuführung der Daten vermieden wird.

Somit ist bei der Entwicklung der Hardwarestruktur auf gute Parallelisierbarkeit und eine effektive Datenzuführung zu achten. Eine kontinuierliche Datenzuführung und ein guter Datenfluss sind in Hardware wichtiger, als unnötige Berechnungen zu vermeiden.

In Software sollte hingegen eine Vorauswahl getroffen werden, welche Daten relevant sind und welche Berechnungen erforderlich sind, damit Rechenleistung eingespart wird. Die Möglichkeiten der Parallelisierung sind in den Architekturen aktueller Prozessoren deutlich angestiegen, aber sie können nicht für alle Algorithmen in vollem Umfang genutzt werden. Die parallele Implementierung von mehreren Korrelationen sollte nur mit Multicore-Prozessoren unter Berücksichtigung der Probleme bei der Datenzuführung durchgeführt werden. In einem Single-Prozessor bieten sich hingegen für die hier betrachteten Anwendungsfälle keine Möglichkeiten einer parallelen Mehrfachinstanziierung der Korrelationen für Echtzeitsysteme.

## 4.4 Echtzeitfähige Implementierung durch datenflussorientierte Stereobildauswertung

Aus den Untersuchungen in diesem Kapitel können die Anforderungen für eine echtzeitfähige Stereobildauswertung für die zwei betrachteten Szenarien Spurwechselassistent für Straßenfahrzeuge und Hindernisdetektion für ein fahrerloses Transportfahrzeug abgeleitet werden. Um alle relevanten Objekte zu erfassen, ist eine regelmäßige Abtastung des Bildes erforderlich, damit ein dichtes Netz aus 3D-Punkten entsteht. Die Berechnung einer dichten Tiefenkarte im Sinne der Verfahren in Kapitel 2 erzeugt viele, jedoch nicht unbedingt zuverlässige Daten, während die nachfolgenden Systeme eine deutlich höhere Datenrate bewältigen müssen. Es ist erforderlich, einen Kompromiss zwischen einer dichten Tiefenkarte und den verhältnismäßig dünnen, aber dafür zuverlässigen Tiefenkarten direkt aus der photogrammetrischen Auswertung zu finden.

Für beide Szenarien gilt, dass sich aus der Reaktionszeit und den physikalischen Parametern der Bewegung die Echtzeitbedingung und der Messbereich ergeben. In Abhängigkeit von der Größe der relevanten Objekte, ihrer Abbildung im erforderlichen Messbereich und der Blockgröße kann die Abtastrate bestimmt werden.

Die Blockgröße wurde aus informationstechnischen Gründen auf 16x1 Pixel festgelegt. Die Größe von 16 Pixeln resultiert aus den Möglichkeiten der aktuellen Hardware zur Berechnung der QNKKFMF. Durch die Ausrichtung des Blockes in Suchrichtung ist der Informationsgehalt ebenfalls in Richtung der Disparität am größten.

Mit dieser Blockgröße ergibt sich für das Szenario "Spurwechsellassistent" eine Abtastrate über dem Bild von 128 Blöcken/Zeile, d.h. alle 8 Pixel muss ein Referenzblock platziert werden. Wird für das Szenario "fahrerloses Transportfahrzeug" als kleinstes Objekt der Oberkörper eines Menschen angenommen, so ist eine Schrittweite von 16 Pixeln ausreichend, da ein Mensch in 25 m Entfernung noch etwa 48 Pixel breit ist. Wird jedoch aus Sicherheitsgründen angenommen, dass bereits eine Extremität, z.B. ein Unterschenkel erfasst werden muss, dann sollte die Schrittweite der Referenzblöcke mindestens bei 8 Pixeln (optimal 4 Pixel) liegen.

Durch diese Abtastraten ergeben sich wegen der Disparitäten, die in beiden Fällen in einem Bereich bis 256 Pixel ermittelt werden müssen, Datenmengen, die von den, in eingebetteten Systemen verfügbaren Prozessoren nicht innerhalb der ermittelten Echtzeitbedingungen bewältigt werden können. Für den Fernbereich wurde eine erforderliche zeitliche Abtastrate und somit eine Bildfrequenz von 25 Bildern/s ermittelt. Für den Nahbereich wurden aufgrund der direkten Steuerung des Systems und die deutlich geringere Geschwindigkeit eine Bildfrequenz von 10 Bildern/s ermittelt.

Aufgrund dieser Anforderungen und des für die Korrelationskriterien ermittelten Rechenaufwandes in Hardware und in Software wurde eine Entscheidung für die Umsetzung in Hardware getroffen. Als Hardwareplattform wurde ein FPGA gewählt, da es sowohl Hardwarestrukturen für eine echtzeitfähige Stereobildauswertung als auch Strukturen für einen Prozessor aufnehmen kann und somit auch eine kontrollflussorientierte Auswertung der Tiefenkarten ermöglicht.

Der erforderliche Rechenaufwand für eine Implementierung mittels der einfachen blockbasierten Korrelation ist aufgrund der hohen Abtastraten und des großen Messbereiches und der daraus resultierenden Disparität auch für eine Hardwareimplementierung sehr hoch. Daher muss der Rechenaufwand auf das erforderliche Maß reduziert werden.

Die Unterschiede in der Abbildbreite eines Objektes über den Messbereich bei dem Szenario "Nahbereich" sind zwar deutlich, aber sie ermöglichen dennoch eine Verarbeitung mit einer einfachen blockbasierten Korrelation. Für das Szenario "Fernbereich" ist eine Anpassung der Blockgröße oder der Bildauflösung an das zu vermessende Objekt erforderlich. Daher wurde hierfür ein hierarchisches Verfahren zur Auswertung von Stereobildern entwickelt, welches die Anpassung an die Objektgröße berücksichtigt.

Während für die Vermessung im Fernbereich große Brennweiten ( $f \geq 25$  mm) verwendet werden, sind im Nahbereich aufgrund des großen Öffnungswinkels deutlich kleinere Brennweiten



erforderlich. Die Betrachtung der Auswirkung der Brennweite zeigt, dass bei einer Brennweite von 12,5 mm eine Entzerrung des Bildes noch nicht erforderlich ist. Bei Brennweiten unterhalb von 6 mm ist eine Entzerrung zwingend notwendig, wodurch sich der Rechenaufwand stark erhöhen würde. Werden einfache Kalibriermethoden verwendet, welche für eingebettete Systeme adäquat sind, sollten niedrige Brennweiten ( $f \geq 12,5$  mm) vermieden werden.



# Kapitel 5

## Umsetzung von zwei Verfahren zur Stereobildauswertung in Echtzeit auf einem FPGA

Die Auswertung von Stereobildern ist ein aufwändiger Verarbeitungsschritt, wobei der Rechenaufwand mit den Anforderungen an die Tiefenkarte stark variiert. Die in dieser Arbeit betrachteten Szenarien erfordern eine Implementierung der Stereobildauswertung mit mindestens 25 Bildern/s. In Kapitel 4 wurde die Entscheidung für eine Hardwareimplementierung auf einem FPGA durch den hohen Rechenaufwand der blockbasierten Korrelation über den großen Disparitätsbereich und durch die harten Echtzeitanforderungen begründet.

Wird ein Algorithmus vorwiegend von Datenflussabhängigkeiten dominiert, kann er für einen hohen Datendurchsatz mit geringem Ressourcenverbrauch umgesetzt werden. Die Datenflussabhängigkeiten bestimmen dabei die Struktur des Algorithmus.

In diesem Kapitel werden zwei Verfahren zur Stereobildauswertung für eine datenflussorientierte Implementierung auf einem FPGA untersucht.

- Ein **hierarchisches Korrelationsverfahren** wird vorgestellt, durch das der Rechenaufwand bei der Stereobildauswertung deutlich gesenkt wird.
- Durch eine massiv parallele Implementierung der Korrelation wird eine **echtzeitfähige Struktur** für ein **lineares blockbasiertes Korrelationsverfahren** erreicht.

Beide Verfahren ermöglichen eine Implementierung der Stereobildauswertung auf einem FPGA unter Berücksichtigung der in Kapitel 4 abgeleiteten Echtzeitanforderungen. Beide Verfahren werden anhand der Hardwareimplementierung hinsichtlich der erforderlichen Ressourcen und der Verarbeitungsgeschwindigkeit untersucht. Im Kapitel 6 werden beide Verfahren bezüglich der Bildverarbeitungseigenschaften verglichen.

## 5.1 Hierarchisches Verfahren zur datenflussorientierten Stereobildauswertung

Bei der herkömmlichen Anwendung des Blockmatchingverfahrens auf die Disparitätsbestimmung werden für nahe Objekte deutlich mehr Tiefenpunkte aufgenommen als für ferne Objekte. Bei einem Messbereich von 10 bis 150 m und einer Objektbreite von 2 m wird das Objekt mit dem in Abschnitt 4.2.3 verwendeten Kamerasystem in der Nähe (10 m) auf ca. 386 Pixeln und in der Ferne (150 m) nur noch auf 26 Pixeln abgebildet<sup>1</sup>. Der gleiche Effekt ergibt sich für die Disparität (10 m: 180 Pixel; 150 m: 11 Pixel). Die Auflösung der Disparität bestimmt die Auflösung in Z-Richtung. Diese deutlich höhere Genauigkeit ist für die reine Positionsbestimmung nicht relevant, teilweise sogar störend, da die nachfolgenden Algorithmen den höheren Datenaufwand ebenso verarbeiten müssen.

Der Ansatz des hierarchischen Verfahrens ist es, die Auflösung über den gesamten Messbereich nahezu konstant zu halten. Um dies zu erreichen, wird eine Bildpyramide [55] verwendet. Im Gegensatz zur klassischen Suche mittels einer Bildpyramide wird jedoch in allen Ebenen das gesamte Bild mit einem sehr kleinen Suchbereich [87] bearbeitet.

Das hierarchische Verfahren ist vor allem für die Positionsbestimmung von Objekten über einen sehr großen Entfernungsbereich interessant und wurde im Rahmen dieser Arbeit speziell hierfür entwickelt. Der Ansatz des Suchverfahrens beruht auf dem Ziel der ähnlichen Abbildung eines Objektes in unterschiedlichen Entfernungen. Nachfolgend wird das Prinzip genauer beschrieben.

### 5.1.1 Prinzip des hierarchischen Verfahrens

Bei der Abtastung von analogen Signalen gilt das Abtasttheorem nach Shannon [36]. Es besagt, dass ein Signal mit mindestens der doppelten Frequenz seiner höchsten auftretenden Frequenz abgetastet werden muss, damit das Signal rekonstruierbar ist. Bei der Erstellung einer Tiefenkarte besteht das Signal aus dem Tiefenverlauf eines Objektes oder einer Szene. Die höchste Frequenz hängt von der Aufgabenstellung ab. Soll eine genaue Oberflächenrekonstruktion eines Objektes mittels der Stereophotogrammetrie durchgeführt werden, sind die betrachteten Strukturen verhältnismäßig fein. Hier wird eine sehr dichte Tiefenkarte benötigt. Sind jedoch nur die Position und die Abmaße des Objektes von Interesse, kann eine sehr viel gröbere Abtastung gewählt werden. Da sowohl Personen als auch Fahrzeuge (PKWs, LKWs und Motorräder) durch Abstraktion im Tiefenverlauf als nicht periodisches Rechtecksignal betrachtet werden können, ist die Breite des schmalsten Impulses für die Bestimmung der Abtastfrequenz relevant.

Die Informationen, die von einem Messobjekt relevant sind, stellen sich meist als Einzelimpuls mit endlicher Tiefe und endlicher Breite dar. Durch die Wirkung der Optik verändert sich jedoch das Abbild in Abhängigkeit von seiner Position, insbesondere von der Entfernung. Wird nun eine konstante Abtastfrequenz über das gesamte Bild angenommen, so wird ein nahes Objekt

---

<sup>1</sup>Kamerasystem 2x Photonfocus MV-D1024; Basisbreite 70 cm; Objektive: MeVis C Brennweite 25 mm

deutlich öfter abgetastet als dasselbe Objekt in der Ferne.

Dadurch entsteht für nahe Objekte eine höhere Informationsdichte, deutlich höheres Datenaufkommen und somit eine deutlich höhere erforderliche Rechenleistung. Für die nachfolgenden Algorithmen wäre eine Normierung der Abbildgröße auf die Entfernung nach der Positionsvermessung ideal. Dabei könnte eine konstante Repräsentation eines Objektes erreicht werden. Für die Vermessung an sich wäre der Aufwand für nahe Objekte nach wie vor deutlich größer als für ferne Objekte. Der Rechenaufwand für die Positionsvermessung macht jedoch einen beträchtlichen Anteil des gesamten Rechenaufwandes aus. Weiterhin sind herkömmliche Verfahren zur Lösung des Korrespondenzproblems nicht datenflussorientiert. Aus diesem Grund wurde das hierarchische Verfahren im Rahmen dieser Arbeit [87] entwickelt, welches eine signifikante Einsparung an Rechenaufwand sowohl für die Positionsvermessung als auch für die nachfolgenden Algorithmen erreicht.

Das Prinzip des Verfahrens kann durch folgende Schritte beschrieben werden:

1. Die Bildebenen  $E_l(x_{max}, y_{max})$  der Bildpyramide werden mit  $l$  als Index definiert. Aus der originalen Bildebene  $E_0(x_{max}, y_{max})$  mit  $l = 0$  in hoher Auflösung werden  $n$ , mit  $n \in \mathfrak{R}$  weitere Bildebenen mit niedrigeren Auflösungen generiert. Die Ebenen unterscheiden sich dadurch, dass sie mit steigender Ebenenzahl eine um den Faktor  $k$  mit  $k = \frac{1}{2}$  verringerte Auflösung besitzen. Der Faktor  $k$  wird je nach Anwendung frei festgelegt.
2. Die Ebenen werden im Sinne des Blockmatchingverfahrens in der Form durchsucht, dass ausgehend von Blöcken konstanter Größe (in Pixeln) in allen Ebenen die gleiche Suchweite (in Pixeln) verwendet wird.
3. Anschließend wird von allen Blöcken in allen Ebenen das lokale Extremum bestimmt.
4. Diese lokalen Extrema werden über alle Ebenen spaltenweise verglichen und so die Disparität mit jeweils maximaler Genauigkeit bestimmt.

Durch dieses Verfahren wird der gesamte Messbereich in  $n$  Messbereiche unterteilt. In diesen variiert die Darstellung des entsprechenden Messobjektes ebenfalls, jedoch bei weitem nicht so stark wie beim einfachen Blockmatching. Alle Messbereiche werden quasi simultan vermessen. Anschließend werden die Informationen entweder in einem globalen Messbereich zusammengefügt oder lokal ausgewertet. Weiterhin ist das Messverfahren sehr gut datenflussorientiert in Hardware implementierbar, wenn die Blockgröße der Suchweite in den Ebenen angepasst wird. Daraus resultiert eine Bildpyramide mit einem Faktor  $k = 1/2$ . Für dieses Verfahren wurde im September 2008 ein Patent [90] erteilt.

### 5.1.2 Generierung der Ebenen und der Messbereiche

Zur Erstellung der Ebenen muss die Auflösung der originalen Bildebene in mehreren Stufen verringert werden. Durch diesen Schritt ergeben sich auch die entsprechend abgestuften Messbereiche (siehe Tabelle 5.1). Prinzipiell kann jede Kombination von Auflösungsschritten verwendet

werden. Durch Halbierung der Auflösung von Stufe zu Stufe erhält man einen Messbereich, der logarithmisch abgestuft ist. Dieser Algorithmus lässt sich optimal in Systeme implementieren, die mit digitalen Zahlen im Binärsystem rechnen. Dies trifft auf den überwiegenden Teil der aktuell verfügbaren datenverarbeitenden Maschinen zu. Logarithmische Verfahren werden im Allgemeinen zur kompakten Darstellung von großen Funktionsbereichen verwendet.

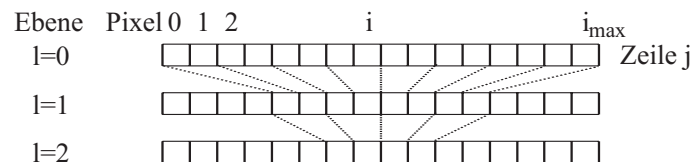


Abbildung 5.1: Generierung der Ebenen durch Mittelwertbildung

Die Generierung der Ebenen kann durch unterschiedliche Verfahren (Selektion, Mittelwert, Wichtung mit Gauss) erreicht werden. Da alle Pixel jedoch die gleiche Relevanz für das resultierende Messergebnis haben, werden durch eine einfache Mittelwertbildung alle Pixel bei der Erzeugung der Ebenen gleichermaßen berücksichtigt. Die verschiedenen Ebenen können direkt von der Originalauflösung abgeleitet werden. Wird eine Mittelwertbildung bei der Erstellung der Ebenen angewendet, kann eine sukzessive Ermittlung der Pixel aller Ebenen durchgeführt werden, sofern keine Rundung vorgenommen wird. Dazu wird dasselbe Verfahren, das auf die Originalbildebene angewendet wurde, für jede der neu erstellten Ebenen verwendet. Dies ist in Abbildung 5.1) grafisch dargestellt. Die Ergebnisse der direkten und der sukzessiven Berechnung der Pixel der Ebenen sind identisch. Für die Implementierung in Hardware erweist es sich als vorteilhaft, da so die Erstellung aller Ebenen gleichzeitig erfolgen kann (siehe Abschnitt 5.1.5.1). Weiterhin ist vorteilhaft, dass die Division durch 2 im Binärsystem sehr einfach durch eine Verschiebung der Zahl um eine Stelle nach rechts erfolgen kann.

Beim hierarchischen Suchverfahren repräsentiert jede Ebene einen bestimmten Messbereich. Die Verteilung der verschiedenen Messbereiche auf die einzelnen Ebenen hängt vom stereophotogrammetrischen Kamerasystem ab, besonders von der Kamerakonstante und der Basisbreite. Die konkrete Auflösung der Kamera spielt für die Größe der einzelnen Messbereiche keine Rolle. Die Breite des Suchbereiches (im Beispiel 16 Pixel in Originalauflösung) ist hingegen eine wichtige Größe. Die Aufteilung der Messbereiche für ein Beispielsystem mit zwei Kameras vom Typ JAI CV-M4 und einer Basisbreite von einem Meter ist in Tabelle 5.1 dargestellt. In der am höchsten aufgelösten Ebene ergibt sich ein Problem mit dem Messbereich. Er umfasst hier theoretisch 1200 - 110 m, praktisch sind in diesem Fall die relevanten Objekte (also Fahrzeuge) in dieser Entfernung nur noch 1 Pixel groß. In ca. 400 m sind PKWs ca. 10 Pixel breit und somit prinzipiell praktisch vermessbar, wenn die Sichtverhältnisse es zulassen.

Bei einer Originalzeile mit 1024 Pixeln ergibt sich in der 5. Ebene die mit dem hierarchischen Verfahren maximal zu ermittelnde Disparität von 512 Pixeln (siehe Tabelle 5.1).

Ebene/k	Auflösung		Suchbereich [Pixel]	benötigte Korrelationswerte [Pixel]	Messbereich [m]
	1/	2			
0	1/1	1024x1024	16	16	400(1200) - 110
1	1/2	512x1024	32	24	110 - 57
2	1/4	256x1024	64	32	57 - 28
3	1/8	128x1024	128	40	28 - 14
4	1/16	64x1024	256	48	14 - 7
5	1/32	32x1024	512	56	7 - 3,5

Tabelle 5.1: Messbereichsübersicht für ein Beispielsystem

### 5.1.3 Ermittlung der Position der lokalen Extrema

In den so erstellten Ebenen wird eine Suche nach Korrespondenzen durchgeführt. Es werden im Referenzbild mit all seinen Ebenen Blöcke definiert. Diese Blöcke werden, ausgehend von der Blockposition der Referenzblöcke, wie in Abbildung 5.2 dargestellt, pixelweise über das Suchbild der jeweiligen Ebene verschoben, wobei für jede so entstehende Blockkombination ein Korrelationskriterium angewendet wird. Die Suchweite entspricht in diesem Fall der Blockbreite. Somit ergibt sich für jeden Referenzblock der Referenzbildebene eine Folge von Korrelationswerten. Diese Folge von Korrelationswerten wird nach Extrema durchsucht. Je nach verwendetem Korrelationskriterium sind dabei Maxima oder Minima für die größtmögliche Übereinstimmung maßgebend.

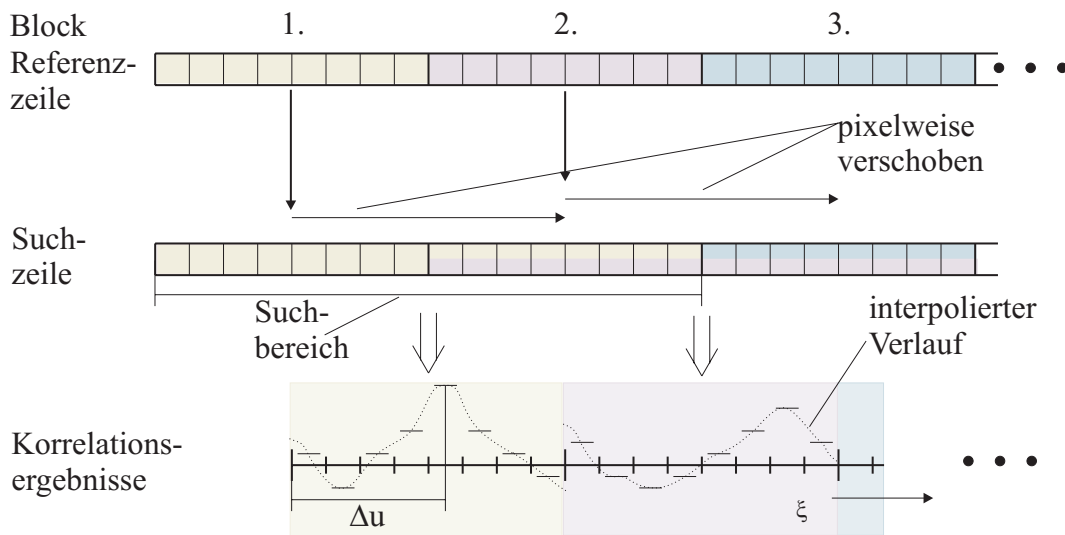


Abbildung 5.2: Typisches Beispiel einer Korrelationsfunktion (KKFMF)

Im konkreten Fall wurde die normierte mittelwertfreie Kreuzkorrelationsfunktion verwendet, da sie bei der Verwendung des Systems im Außenbereich Vorteile bezüglich der Robustheit bei kritischen Beleuchtungssituationen hat [9]. Die Blöcke erstrecken sich über 16x1 Pixel. Für jeden der Referenzblöcke werden 16 Korrelationswerte ermittelt. Da die Disparität mit dem obenstehenden Verfahren nur auf Pixelgenauigkeit ermittelt werden kann, wird im Anschluss

noch eine Subpixelinterpolation auf Basis der quadratischen Interpolation durchgeführt (siehe Abschnitt 2.3.4).

### 5.1.4 Erstellen der Tiefenkarte aus den lokalen Extrema

Nachdem die lokalen Extrema bestimmt sind, müssen die Disparitätskarten der Ebenen zu einer Tiefenkarte zusammengefügt werden. Dafür stehen zwei prinzipielle Ansätze zur Verfügung. Zum einen kann man bei der Erstellung der Tiefenkarte weiterhin nach dem ebenenübergreifenden Maximum der Korrelationsfunktion suchen, zum anderen kann die Position des Maximums durch entsprechende Maxima höherer Ebenen bestimmt werden. Die Maxima in den höheren Ebenen repräsentieren die gleiche Entfernung wie die der niederen Ebene, jedoch mit einer geringeren Auflösung. Bei beiden Verfahren sind für den Vergleich bzw. die Bestätigung die in den zwei Beispielen gekennzeichneten Blöcke in Abbildung 5.3 relevant. Die beiden Pfeile durchlaufen die für den Vergleich verwendeten Blöcke in der Richtung, in der der Vergleich durchgeführt wird.

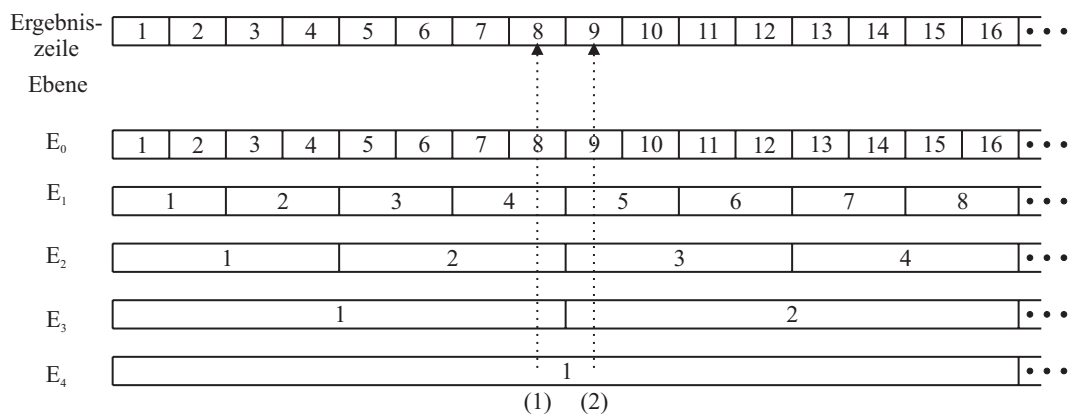


Abbildung 5.3: Auswahl der Blöcke zum Vergleich der lokalen Korrelationsmaxima

Bei der Auswertung der Ebenen nach den besten Maxima ist zu beachten, dass dies nicht mit allen Korrelationskriterien einfach möglich ist. Um den Vergleich ohne Bezug zum entsprechenden Referenzblock durchführen zu können, müssen die Werte normiert sein. Beachtet man dies nicht, gewinnt unter Umständen die falsche Ebene mit einem Nebenextremum. Von den vorgestellten Kriterien kommen hierfür also nur die NKKFMF und die NSSDMF in Frage. Die Entscheidung für die NKKFMF resultierte aus den zwischen 1 und -1 begrenzten Ausgangswerten.

Die zweite Variante der Auswertung orientiert sich an der Position der Maxima in den Ebenen und dem Vergleich dieser zueinander. Wenn in einer niedrigen Ebene ein Maximum überprüft werden soll, dann wird die Ebene darüber in dem entsprechenden darüberliegenden Block ebenfalls nach Maxima durchsucht. Befindet sich dort ebenfalls das Maximum, möglicherweise auch nicht so ausgeprägt, so wird die Position bestätigt. Dieses Verfahren kann jeweils in die nächst niedriger aufgelöste Ebene übertragen werden. Die Position in der höheren Ebene muss jedoch entsprechend dem Verfahren zur Erstellung der Ebenen angepasst werden.



In Abbildung 5.4 ist ein Beispiel für die Auswertung der Blöcke über den maximalen Korrelationswert und über die Position der Maxima über alle Ebenen dargestellt.

Die Disparitäten  $u_l$  und die x-Koordinaten  $x_l$  in der Tiefenkarte sind jedoch nur für die jeweilige Ebene gültig, in der sie ermittelt wurden. Für die Berechnung der Objektkoordinaten aus der Disparität und den Bildkoordinaten mittels der Gleichungen 2.1, 2.2, 2.3 sind die Disparitäten  $u'$  und Koordinaten  $x'$  in der Originalauflösung der Stereobilder erforderlich. Diese können mit Gleichung 5.2 und 5.3 berechnet werden. Dabei gibt  $l$  die zugehörige Nummer an. Zur Berechnung von  $x_l$  werden  $s$ , die Differenz zwischen den Mittelpunkten zweier benachbarter Referenzblöcke, und  $\nu_l$ , die positionsbezogene Nummer des entsprechenden Referenzblockes in der Ebene, benötigt [38]. Es gilt

$$x_l = \nu_l \cdot s \quad (5.1)$$

Die Fenstergröße des Referenz- und des Suchblockes ist in Abschnitt 2.4.2 mit  $m \times n$  definiert.  $\frac{m}{2}$  stellt somit die Mitte des Referenzblockes dar. Damit folgt

$$x' = (x_l + \frac{m}{2}) \cdot 2^l \quad (5.2)$$

$$u' = (u_l + u_{min}) \cdot 2^l \quad (5.3)$$

$u_{min}$  in Gleichung 5.3 ist ein Offsetwert, der in bestimmten Szenarien erforderlich ist und repräsentiert die minimale Disparität. Im hier betrachteten Fall ist  $u_{min}$  Null. Alle Größen außer  $\nu_l$  werden in Pixel gemessen.

### 5.1.5 Hardwareimplementierung des hierarchischen Algorithmus

Im Folgenden wird die Hardwareimplementierung des Algorithmus beschrieben. Dabei stand die Optimierung des Algorithmus unter Einhaltung der durch die Hardware gegebenen Randbedingungen im Vordergrund. Dazu zählen insbesondere die Eigenschaften der Kamera wie der Pixeltakt und die Bildgröße sowie die spezifischen Eigenschaften der Hardwareplattform.

Um einen freien Austausch verschiedener Implementierungsvarianten zu ermöglichen, wird das gesamte Verfahren zur Erstellung der Tiefenkarte als modulares System aufgebaut. In Abbildung 5.5 ist dieses Prinzip stark vereinfacht dargestellt. Nach der Bildaufnahme werden zunächst die beiden Bilder einzeln aufbereitet, die Ebenen generiert und dem Hardwarekorrelator zugeführt. Dieser kann auf unterschiedliche Art realisiert werden, wie anschließend diskutiert wird. Abschließend werden die Werte der Disparitäten und dann die Tiefenkarte aus den Korrelationswerten ermittelt. Vor- und Nachbereitung der Daten sind wesentlich von dem verwendeten Korrelationsansatz abhängig.

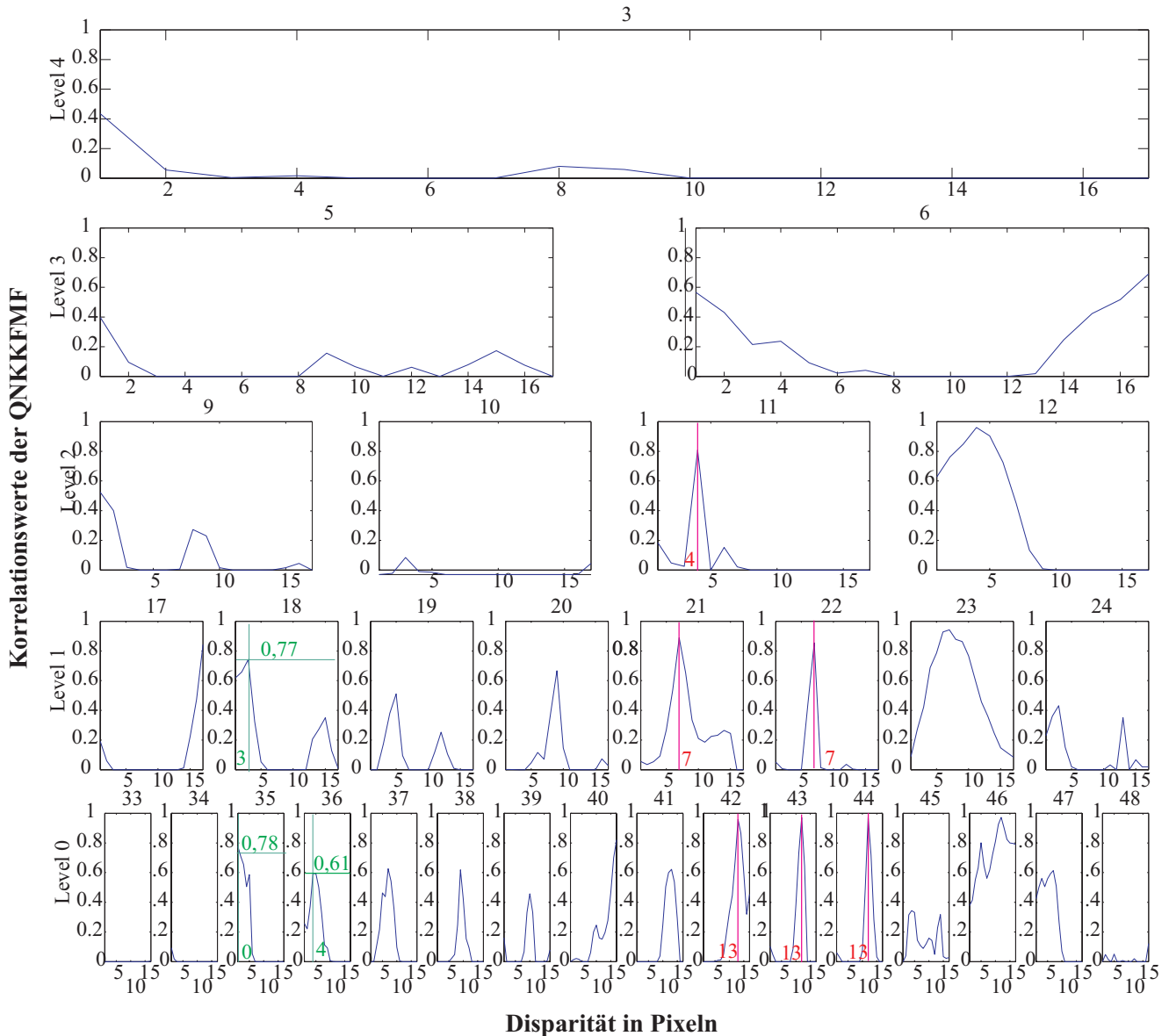


Abbildung 5.4: Beispiele für die Zusammenführung der Ebenen

**Ermittlung der Disparität über das Maximum (grün):** Hierbei stellen die Blöcke 35 und 36 des Levels 0 sowie die entsprechenden darüber liegenden Blöcke ein gutes Beispiel dar. Das Maximum im Block 18 im Level 1 liegt bei einer Disparität von 4 Pixeln mit einem Wert 0,77, während die Blöcke 35 und 36 des Levels 0 ein Maximum bei (0;0,78) und eines bei (5;0,61) haben. Hierbei gewinnt für den Block 35 des Levels 0 die Disparität 0 und für den Block 36 die Disparität 3 des Blockes 18 des Levels 1. Die Disparität 3 des Levels 1 kann mit Gleichung 5.3 in die Disparität 6 der Originalauflösung (Ebene 0) umgewandelt werden.

**Ermittlung der Disparität über die Position der Maxima zueinander (rot):** Für die Maxima der Blöcke 42, 43 und 44 des Levels 0 mit der Disparität 13 finden sich entsprechend im Level 1 die Blöcke 21 und 22 mit der Disparität 7 und im Level 2 der Block 11 mit der Disparität 4 wieder. Dadurch wird jeweils die halbierte Disparität in den Ebenen mit der entsprechend halben Auflösung verifiziert.

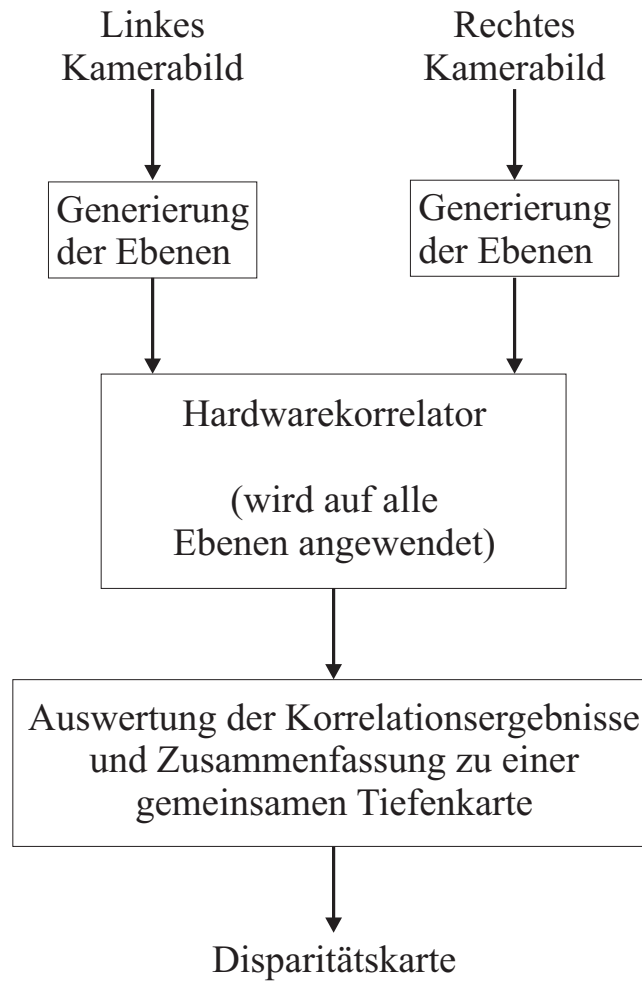


Abbildung 5.5: Modularer Aufbau

Für eine kontinuierliche Verarbeitung der Stereobilddaten in Echtzeit auf einer Hardwareimplementierung ist ein vollständig synchrones System notwendig. Dies bedeutet, dass die Kameras mit einem zum Systemtakt synchronisierten Takt arbeiten und der Arbeitstakt identisch bzw. synchron zum Kameratakt ist. Die Realisierung eines vollständig synchronen Designs wird in Abschnitt 7.3 beschrieben.

#### 5.1.5.1 Generierung der Ebenen und Zuführung der Daten für eine Echtzeitverarbeitung

Die Verarbeitung der Bilder erfolgt vollständig zeilenorientiert, da aufgrund der Epipolaregeometrie beim Stereonormalfall der Suchbereich des korrespondierenden Punktes auf die Zeile eingeschränkt werden kann (siehe Abschnitt 2.3.3). Dadurch wird der erforderliche Speicherplatz stark reduziert. Durch eine Mittelung mehrerer Zeilen (z.B. 3-4 Zeilen in Spaltenrichtung) kann der Aufwand zur Erzeugung des Stereonormalfalles reduziert werden.

Zur Generierung der Ebenen wird wiederholt von zwei aufeinanderfolgenden Pixeln einer Zeile

der Mittelwert gebildet und dieser in Speichern, die den jeweiligen Ebenen zugeordnet sind, abgespeichert. Anschließend werden die Daten der Ebene nacheinander ausgelesen.

In Abbildung 5.6 ist die Struktur zur Generierung der Bildebenen dargestellt. Die vier Blöcke *Add2* addieren zwei aufeinander folgende Pixel und geben den Mittelwert aus. Die Division erfolgt durch eine Shift-Operation. Die *Add2*-Blöcke generieren ebenfalls die Steuersignale, damit der nachfolgende Block die Werte korrekt übernimmt. Diese Mittelwerte zweier aufeinanderfolgender Pixel sind die Pixel der neuen Ebenen und werden in 5 Dual Port RAMs (DPR) abgespeichert.

Die DPR dienen auch als Schnittstelle zwischen zwei Takten. Der Pixeltakt der Kamera beträgt 28 MHz. Da durch das Verfahren nahezu die doppelte Datenmenge zur Verarbeitung erzeugt wird, erfolgt die Korrelation mit 56 MHz, um die Echtzeitfähigkeit zu gewährleisten..

Die DPRs können mit unterschiedlichen Geschwindigkeiten gelesen und geschrieben werden, es darf jedoch nicht der gleiche Speicherbereich von beiden Schnittstellen gleichzeitig angesprochen werden. Um dies zu verhindern, sind die DPRs mit doppelter Speichertiefe realisiert und werden für jede Zeile umgeschaltet.

Die Ansteuerung zum Schreiben wird durch einen gemeinsamen Zähler realisiert, von dem jeweils nur die oberen Bits für den entsprechenden Speicher als Adresse verwendet werden. Das Most Signifikant Bit (MSB) wird durch das Least Significant Bit (LSB) der Bildnummer bestimmt und schaltet so zwischen dem geschriebenen und dem gelesenen Bild um.

Zum Auslesen der Speicher läuft ein Zähler nacheinander von den jeweiligen Maximalwerten der Speicher auf Null. Dadurch wird die Zeile rückwärts ausgelesen. Die Hardware kann nur in Ausleserichtung der Speicher suchen, da jedoch das linke Bild das Referenzbild ist, muss die Suchrichtung nach links laufen. Dies ergibt sich aus der Epipolargeometrie und ist in Abschnitt 2.3.3 beschrieben.

Die ausgelesenen Daten werden durch einen Multiplexer (MUX) dem Korrelator zugeführt.

### 5.1.5.2 Gleitender Mittelwert und Korrelation der Bildbereiche

Nach dem Auslesen einer Zeile aus den DPRs erscheint die Bildpyramide des Stereobildes wie zwei doppelt lange Bildzeilen. Diese werden in Blöcke geteilt, die miteinander korreliert werden. Hierbei wechselt der Referenzblock alle 16 Pixel und der Suchblock bei jedem Pixel. An den Übergängen zwischen den Ebenen wird dabei jeweils der letzte Referenzblock der niederen mit den ersten 16 Suchblöcken der höheren Ebene korreliert. Aus dieser Berechnung können keine sinnvollen Werte ermittelt werden. Da jedoch durch die Berechnung Shiftregister für die Erstellung der mittelwertfreien Pixel und die Pipeline gefüllt werden, wird die Berechnung nicht abgebrochen, sondern die fehlerhaften Werte später ausgeblendet. Hierzu wird ein Signal verwendet, das beim Auslesen der DPRs erstellt wird. Durch diesen Schritt wird die Verarbeitung sprungfrei gehalten.

Für das hierarchische Verfahren wurde die QNKKFMF (siehe Abschnitt 2.4.3) als Korrelati-

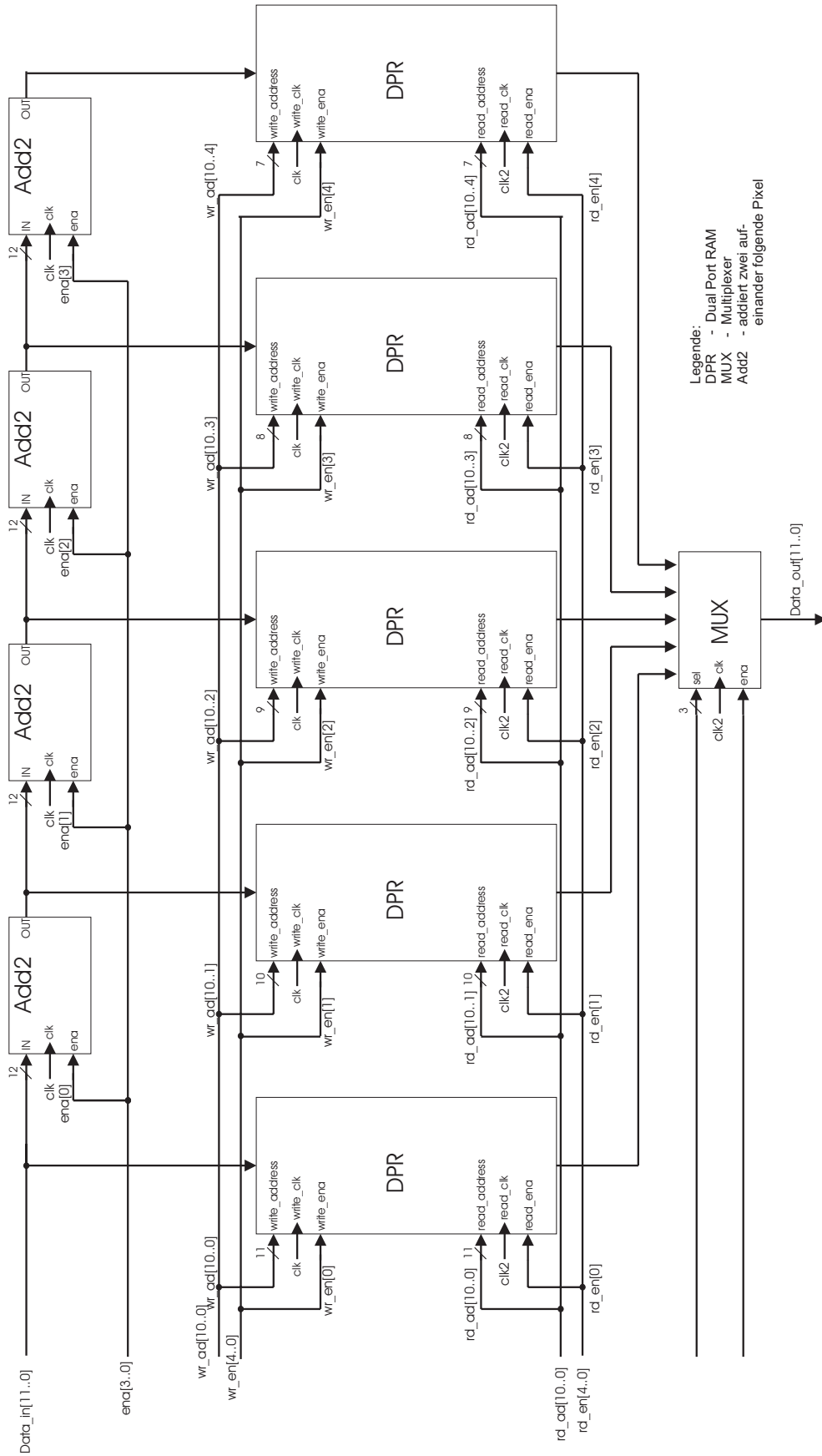


Abbildung 5.6: Struktur zur Generierung der Ebenen in Echtzeit

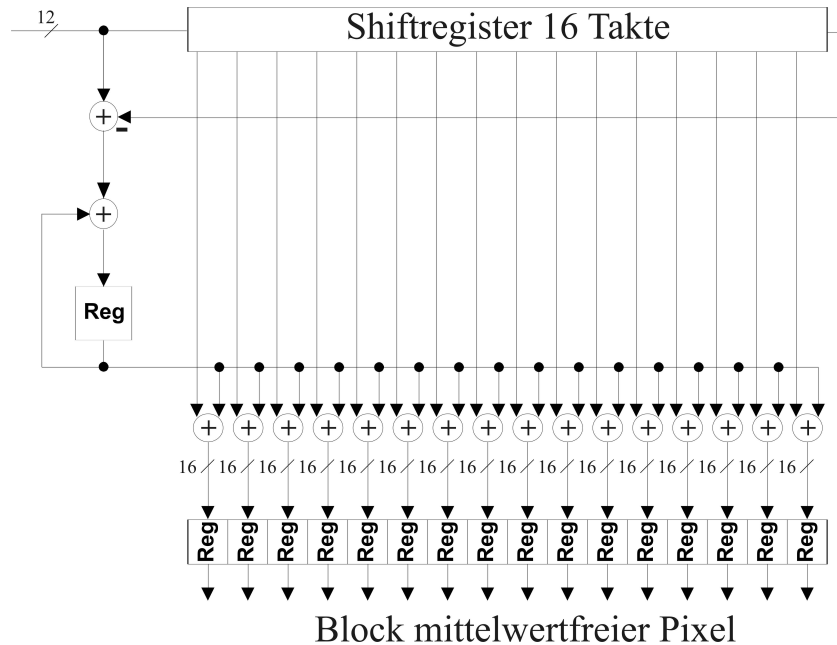


Abbildung 5.7: Zuführung der mittelwertfreien Daten mittels des gleitenden Mittelwertes

onskriterium ausgewählt. Sie wird in voll paralleler Implementierung realisiert. Um die mittelwertfreien Blöcke möglichst effektiv zu generieren, wird der gleitende Mittelwert verwendet. Wie in Abbildung 5.7 dargestellt, werden die Pixelwerte in einem Shiftregister gespeichert. Zum Beginn ist das gesamte Shiftregister auf Null gesetzt. Beim Einschieben der Daten wird aus dem neuesten und dem ältesten Pixel die Differenz gebildet und so die Änderung des Mittelwertes akkumuliert. Der akkumulierte Wert wird durch die Fensterbreite dividiert. Die Fensterbreite wurde auf 16 festgelegt, da so die Division wieder durch eine Shiftoperation realisiert werden konnte. Der Wert wird auf zwei Stellen nach dem Komma gerundet. Von den Pixelwerten im Shiftregister wird nun der Mittelwert abgezogen und das Ergebnis in Registern zwischengespeichert. Der Referenzblock wird alle 16 Pixel gespeichert. Die Struktur dieser Funktion in Hardware ist in Abbildung 4.5 dargestellt.

Anschließend kann die Berechnung der QNKKFMF als paralleles Design mit 108 Pipelinestufen implementiert werden, wie sie in Abschnitt 4.3.2 beschrieben ist. Mit jeder steigenden Flanke des doppelten Pixeltaktes wird ein Korrelationswert berechnet und direkt im Anschluss das Maximum der zu einem Referenzblock gehörenden Korrelationsergebnisse ermittelt.

### 5.1.5.3 Subpixelgenaue Auswertung der lokalen Extrema

Nach der Korrelation erfolgt die lokale ebeneninterne Auswertung ebenfalls mit doppeltem Pixeltakt. Hierbei wird jeweils das Maximum der Korrelationsfunktion gesucht. Daraus werden Datensätze mit Bildnummer, Blocknummer, Zeilennummer und der Disparität (siehe Abbildung 5.8) erstellt.

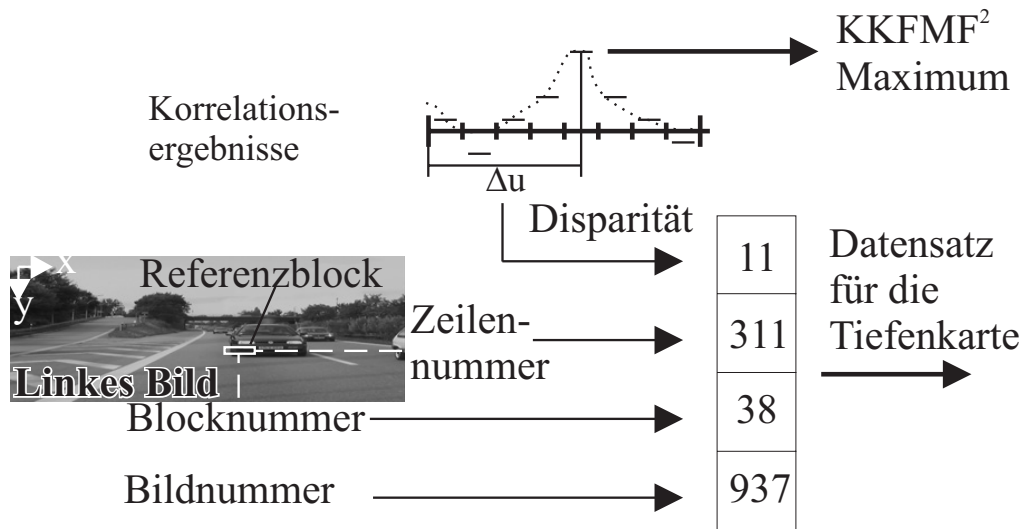


Abbildung 5.8: Erstellung des Datensatzes aus Blockposition, Disparität und Korrelationswerten

Durch Zähler wird die Position des Blockes innerhalb des Bildes ermittelt. Zeitgleich werden die Korrelationsergebnisse mit Hilfe der Schaltung *Dispsearch* aus Abbildung 5.9 pixelweise nach dem Maximum durchsucht. Dazu werden für jeden Wert, der größer ist als seine Vorgänger, der Korrelationswert und die Position innerhalb des Suchbereiches, welche der Disparität entspricht, gespeichert. Für die anschließende Subpixelinterpolation werden weiterhin die Korrelationswerte des Vorgängers und des Nachfolgers abgespeichert.

Am Anfang der Folge ist der Korrelationswert des Maximums noch unbekannt. Er wird auf den ersten Wert der Folge von Korrelationskriterien gesetzt. Sollte der neue Korrelationswert größer sein, so ersetzt er das bisherige Maximum. Mit den gespeicherten Werten des vorausgehenden und des nachfolgenden Korrelationswertes wird eine Subpixelinterpolation mit einer Genauigkeit von  $1/8$  Pixel auf Basis des im Abschnitt 2.3.4 beschriebenen Verfahrens durchgeführt.

Anschließend wird aus der Blockposition und der subpixelgenauen Disparität der Datensatz für den Referenzblock erstellt. Die Datensätze werden für jede Ebene in getrennten Dual-Port-RAMs (DPR) abgespeichert. Die Korrelationswerte stehen in einem Extraspeicher.

#### 5.1.5.4 Generierung der Tiefenkarte aus allen Ebenen

Anschließend werden die Datensätze zur Zusammenführung der Ebenen gleichzeitig aus allen DPRs gelesen. Dabei werden die in den Ebenen übereinanderliegenden Blöcke miteinander verglichen und ausgewertet. Die Zuordnung der Blöcke zueinander ist aus Abbildung 5.3 ersichtlich. Die Daten werden mit dem einfachen Pixeltakt gelesen. Der Speicher dient wieder als saubere Trennung zwischen den beiden Takten.

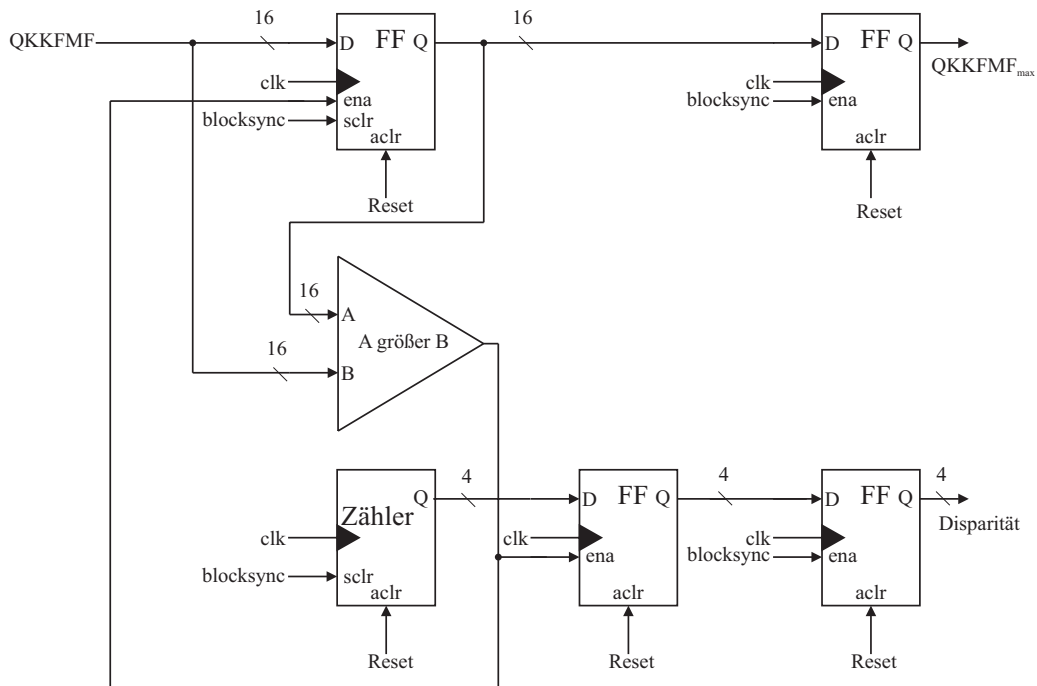


Abbildung 5.9: Pixelgenaue Ermittlung der Maxima mit der VHDL-Funktion *Disparität*

Dazu wurden im Abschnitt 5.1.4 zwei Vorgehensweisen vorgestellt, der Vergleich der Korrelationsmaxima über alle Ebenen und die Verifizierung eines Maximums in einer Ebene durch ein Maximum in einer der höheren Ebenen.

**Auswertung über das Maximum der Korrelationswerte** Es werden zeitgleich aus allen Speichern die Korrelationswerte der zugeordneten Blöcke ausgelesen und das Maximum der 5 Ebenen bestimmt. Der Gewinner des Vergleichs wird als 3D-Punkt in die Tiefenkarte übernommen. Er wird die Blockposition einnehmen, die der zugeordnete Wert in der Ebene 0 inne hatte.

Zum synchronen Vergleich der Korrelationswerte wird, wie aus Abbildung 5.10 ersichtlich, das Auslesen der Speicher parallel realisiert. Anschließend erfolgt während des Vergleichs der nächsten Korrelationswerte das Auslesen und die Ausgabe des Datensatzes des 3D-Punktes.

Der Vergleich über die Korrelationswerte kann nur bei normierten Korrelationskriterien erfolgen. Somit ist für diese Anwendung die QNKKFMF (siehe Abschnitt 4.3.2) der Favorit.

**Auswertung über die lokalen Maxima** Die Auswertung über die lokalen Maxima verschiedener Ebenen erfolgt in einer ähnlichen Struktur. Jedoch ist der Vergleich über die Positionen der Maxima in Hardware schwerer zu realisieren. Gesucht wird der Block der niedrigsten Ebene, dessen Maximum durch mindestens eine darüberliegende Ebene verifiziert wird.

Dazu werden die Blöcke wieder parallel ausgelesen und ein Vergleich der Positionen der Maxima



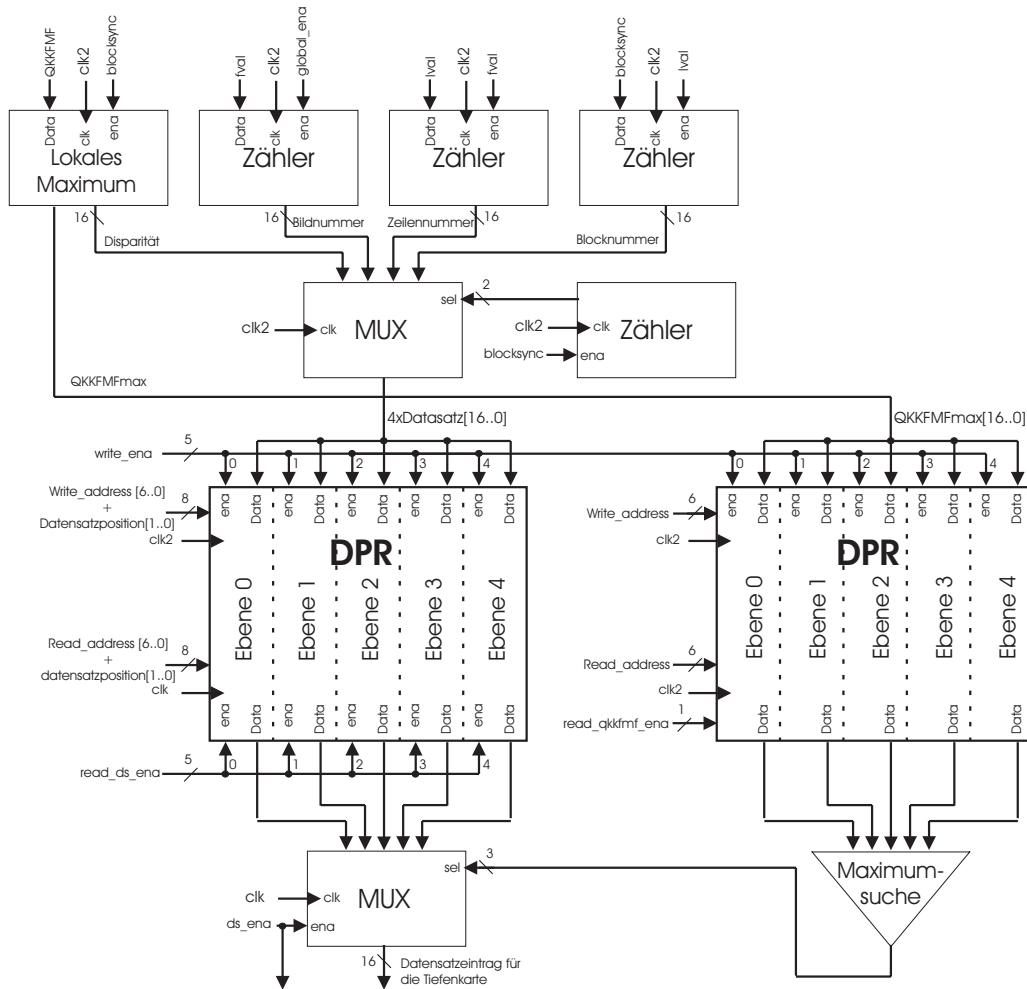


Abbildung 5.10: Ermittlung der dominierenden Maxima in den überlappenden Blöcken der Ebene mit Hilfe der Korrelationswerte

durchgeführt. Hierbei muss in beiden Ebenen je ein Maximum an der durch die Gleichung 5.3 bestimmbaren Position der Ebene 0 liegen. Durch die Abdeckung teilweise recht großer Disparitätsbereiche muss auf die entsprechende niedrigere Ebene sinnvoll gerundet werden.

Diese Methode der Auswertung ist für alle Korrelationskriterien geeignet. Sie lässt sich jedoch einfacher in Software als in Hardware implementieren, da in Software die zeitlichen Randbedingungen nicht durch ein starres Raster bestimmt werden.

**Bereitstellung der Tiefenkarte** Der Datensatz wird während des Vergleiches der Werte des nächsten Blockes aus dem Speicher gelesen und ausgegeben. Ein Datensatz besteht dabei aus vier 16-Bit-Worten. Alle 16 Takte des Pixeltaktes wird ein neuer Datensatz geschrieben. Wurde der Block als ungültig markiert, z.B. weil er an einer Ebenengrenze liegt oder weil der Informationsgehalt des Referenzblockes nicht ausreichend ist, wird die Disparität auf Null gesetzt, der Datensatz jedoch trotzdem übertragen.

## 5.2 Datenflussorientierung durch Anpassung der Implementierung eines linearen blockbasierten Korrelationsverfahrens

Das hierarchische Verfahren reduziert die Auflösung sowohl in X- als auch in Z-Richtung. Für Anwendungsfelder, die sich auf den Nahbereich konzentrieren, sind teilweise höhere örtliche Auflösungen [39] in Z-Richtung und X-Richtung erforderlich. Hierfür wird auf die einfache blockbasierte Korrelation zurückgegriffen. Das Verfahren wird hierbei nicht verändert, aber die Implementierung so angepasst, dass sie datenflussorientiert und parallelisierbar ist. Hierzu wurde eine Korrelatorkaskade entwickelt, bei der das Problem durch Parallelisierung und geschickte Datenzuführung mit minimalem Speicherbedarf und Hardwarekosten gelöst wird. Der allgemein höhere Aufwand der einfachen blockbasierten Korrelation gegenüber dem hierarchischen Verfahren führt dazu, dass die QNKKFMF auf der gegenwärtigen Implementierungsplattform nicht verwendet werden kann. Daher wurde für die Korrelatorkaskade die SAD-Funktion gewählt.

In den folgenden Abschnitten wird das Verfahren unter Verwendung der mittelwertfreien MAD-Funktion beschrieben.

### 5.2.1 Diskussion der Implementierungsprobleme eines einfachen Blockmatchingalgorithmus

Bei der Untersuchung des Blockmatchingalgorithmus auf Implementierbarkeit in Hardware ergibt sich ein Problem beim Zugriff auf die Primärdaten. Prinzipiell ist es kein Problem, eine Korrespondenzsuche über 256 Pixel zu implementieren und diese z.B. 16-fach in einen Hardware-Schaltkreis zu implementieren. Aufgrund des deutlich geringeren Rechenaufwandes kann die SAD-Funktion bis zu 30-mal in der aktuellen Hardware implementiert werden, während bei der NKKFMF bereits die vierfache Implementierung an die Grenzen der Hardwareplattform stößt.

Während der Abarbeitung einer linearen blockbasierten Korrelation ist es erforderlich, häufig auf die gleichen Daten beider Bilder zuzugreifen. Durch die Struktur der Korrelatorkaskade ergibt sich die Möglichkeit, dies auch bei einer starken Parallelisierung zu realisieren. Während die Daten des Referenzbildes mit einem vertretbaren Aufwand verteilt werden können, müssen die Daten des Suchbildes bei z.B. 16 Implementierungen teilweise bis zu 16-mal je Block verwendet werden. Da der Algorithmus bekannt ist, kann die Notwendigkeit der Verteilung der Daten im Vorfeld bestimmt werden. Damit wäre das Problem unter Verwendung einer komplexen Speicherverwaltung direkt lösbar. Aufgrund der Komplexität der erforderlichen Speicherverwaltung wäre eine exklusive Speicherung der betreffenden Zeile des Suchbildes für jeden Korrelator effektiver, würde aber den zur Verfügung stehenden Speicherplatz stark belasten.

## 5.2.2 Datenflussorientierte Implementierung durch eine Korrelator-kaskade

Die Lösung für das Speicherzugriffsproblem liegt in der pipelineartigen Abarbeitung der Korrelation. Dabei wird von einer Instanz jeweils eine bestimmte Anzahl von Korrelationswerten (z.B. 16) für einen Referenzblock berechnet. Die weiteren Korrelationswerte werden in den folgenden Stufen der Korrelator-kaskade berechnet. Damit werden in jeder Stufe 16 Korrelationswerte berechnet. Der Referenzblock wandert dabei einmal durch alle Stufen, während die Daten des Suchbereiches an alle Kaskadenstufen gleichzeitig übertragen werden. Dadurch können theoretisch beliebige Disparitäten durch Hinzufügen neuer Instanzen berechnet werden. Gleichzeitig muss jedoch auch die Auswertung der Korrelationswerte und Ermittlung des Extremums parallelisiert werden, da das Ergebnis der Korrelator-kaskade ein paralleler Datenstrom von Korrelationswerten ist. Bei der Auswertung wird das jeweilige Zwischenergebnis an die nächste Stufe der Auswertung weitergegeben, wodurch der parallele Datenstrom wieder vereinigt wird. Abbildung 5.11 zeigt eine schematische Darstellung der Korrelator-kaskade.

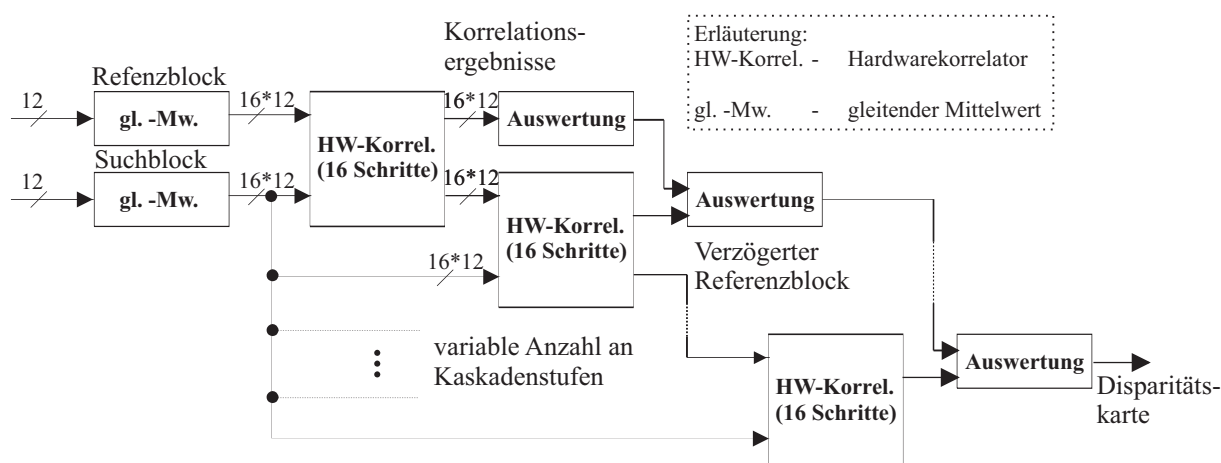


Abbildung 5.11: Schematische Darstellung der Korrelator-kaskade

Weil alle Korrelatoren die gleichen Daten verarbeiten, müssen die mittelwertfreien Daten nur einmalig erstellt werden. Zur Berechnung der mittelwertfreien Pixel wird ebenfalls der gleitende Mittelwert, wie er in Abbildung 5.7 dargestellt ist, verwendet. Der Rechenaufwand bzw. Ressourcenbedarf für diese Operation ist jeweils für Referenz- und Suchblock dem der SAD-Funktion ähnlich.

Die Anzahl der Korrelatoren richtet sich nach der Größe des Suchbereiches. Im vorliegenden Fall bearbeitet jeder Korrelator 16 Suchschritte. Dann werden die Daten weitergeleitet und vom nächsten Korrelator weiterverarbeitet. Um eine Suche über 256 Pixel zu realisieren, sind 16 Korrelatoren erforderlich und damit der 16-fache Hardwareaufwand der SAD-Funktion und der Funktion *Dispsearch*. Prinzipiell kann mit dem Konzept der Korrelator-kaskade eine kontinuierliche Suche für jedes Pixel eines Bildes realisiert werden, wobei der erforderliche Hardwareaufwand sich auf einen Korrelator je Pixel im Suchbereich erhöhen würde.

### 5.2.3 Beschreibung eines Kaskadenelementes

Ein Element der Kaskade besteht, wie bereits im vorhergehenden Abschnitt beschrieben, zum einen aus dem entsprechenden Korrelator, der *SAD-Funktion*, und zum anderen aus der dazugehörigen Auswertung und Ermittlung des Extremums (für die SAD-Funktion ein Minimum) für den betreffenden Abschnitt, in diesem Fall mit *Dispsearch* bezeichnet.

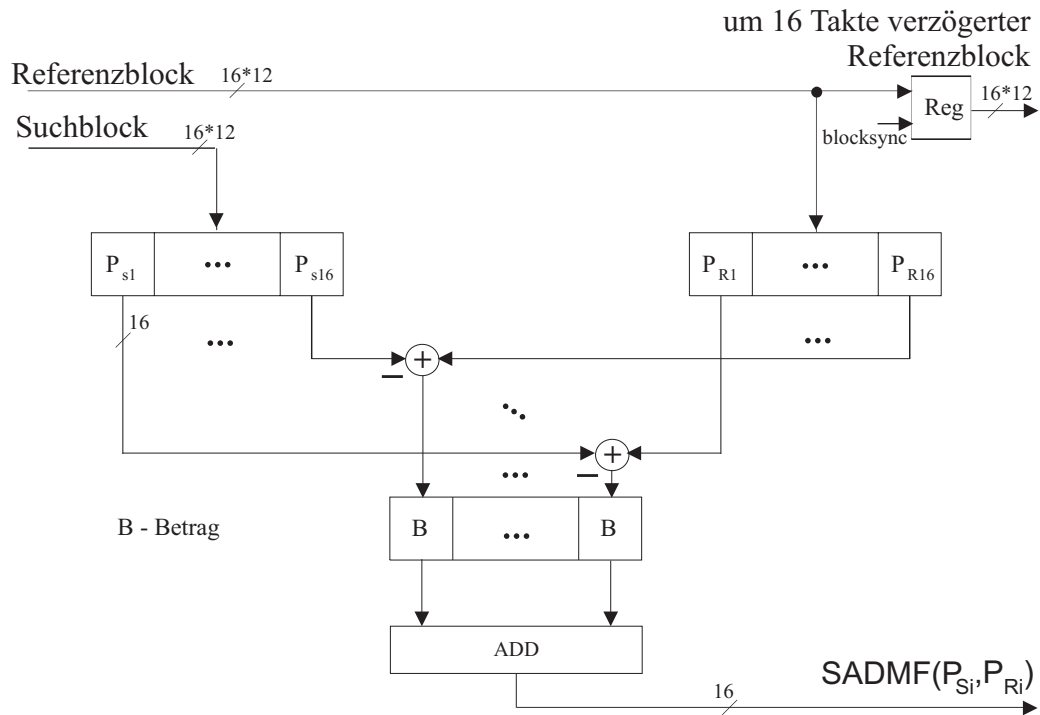


Abbildung 5.12: Blockschaltbild der kaskadierbaren SAD-Funktion

Die Eingangsdaten des Korrelators sind die parallel zugeführten Pixel des Referenzblockes, in diesem Fall  $16 \times 12$  Leitungen, welche von dem vorhergehenden Korrelator übergeben werden, und die parallel zugeführten Pixel des Suchblockes ( $16 \times 12$  Leitungen), welche an alle Korrelatoren gleichzeitig übertragen werden. Sowohl der Referenz-, als auch der Suchblock sind bereits mittelwertfrei. Während der nächsten Takte wird mit jedem Takt ein neuer Suchblock anliegen, während der Referenzblock konstant bleibt. Die Berechnung der SAD-Funktion erfolgt mit der parallelen Implementierung der SAD-Funktion, die im Abschnitt 4.3.2 beschrieben ist. Dieser Vorgang ist prinzipiell unabhängig von der Position in der Korrelatorkaskade.

Als Ergebnis entsteht eine Folge von Korrelationswerten, welche direkt an die VHDL<sup>2</sup>-Funktion *Dispsearch* weitergeleitet wird. Weitere Ausgangswerte sind, wie bereits erwähnt, z.B. der um 16 Takte verzögerte Referenzblock, damit dieser korrekt von der nächsten Stufe der Kaskade weiterverwendet werden kann, und diverse Steuersignale.

Zur Auswertung wird eine Kaskade der Funktion *Dispsearch* aufgebaut. Sie erhält ebenfalls mit den Korrelationswerten der aktuellen Stufe und dem Maximum der vorhergehenden Stufe zwei

<sup>2</sup>VHDL - Very High Speed Integrated Circuit Hardware Description Language

Eingangswerte. Weiterhin wird von der vorhergehenden Funktion noch der Wert des Startpixels des betreffenden Abschnittes der Korrelation festgelegt. In dieser Funktion wird das übergebene Maximum kontinuierlich mit den neuen Korrelationswerten verglichen. Ergibt sich dabei ein neues Maximum, wird es zusammen mit seiner Position sowie dem Korrelationswert des Vorgängers und Nachfolgers für den Vergleich mit den kommenden Korrelationswerten gespeichert. Ergibt sich kein neues Maximum, wird das Ergebnis an die nachfolgende Stufe von *Dispsearch* weitergegeben. Die Subpixelinterpolation erfolgt nur einmalig in der letzten Stufe der Kaskade.

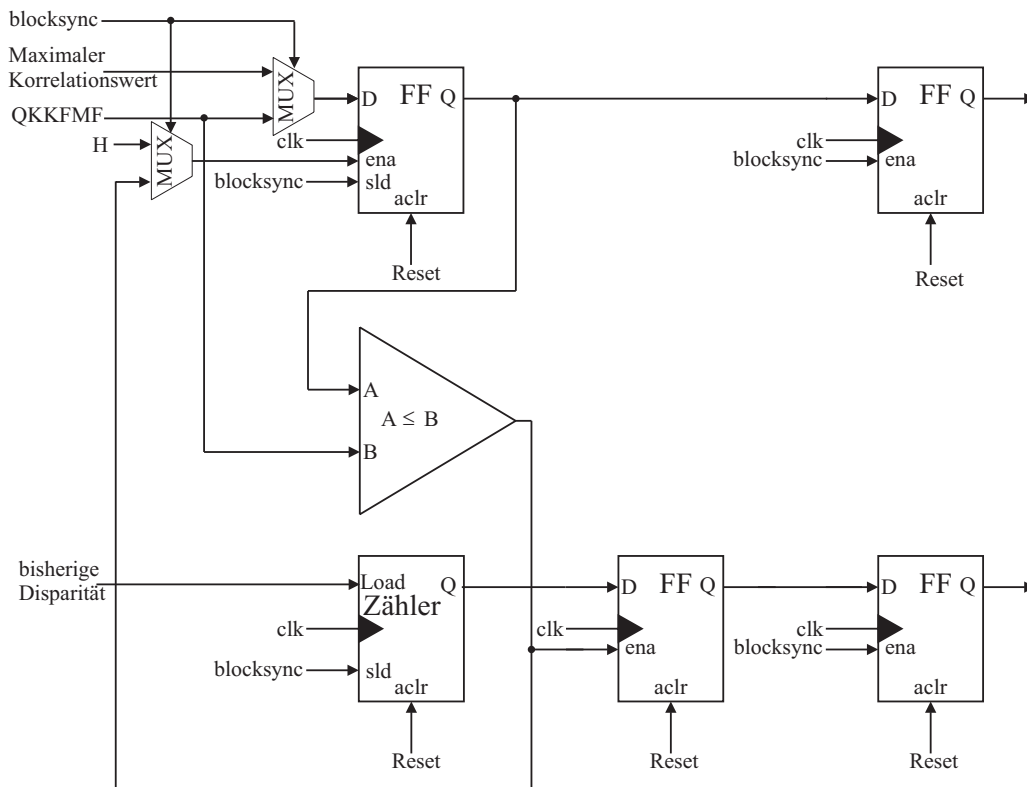


Abbildung 5.13: Blockschaltbild der kaskadierbare VHDL-Funktion "Dispsearch"

Die Weitergabe erfolgt jeweils zum Zeitpunkt des Blockwechsels und wird mit dem Signal *blocksync* ausgelöst. Es wird zentral in der Steuerung erstellt und in der aktuellen Implementierung immer nach 16 Takten aktiviert.

Durch die zeitgleich mitlaufenden Zähler für Bildnummer, Zeilennummer und Blocknummer wird der Datensatz für den 3D-Punkt erzeugt und direkt ausgegeben. Das Format des Datensatzes ist identisch zu dem des hierarchischen Verfahrens.

#### 5.2.4 Laufzeitproblematik für die Korrelatorkaskade bei verschiedenen Korrelationskriterien

Als Korrelator kann in dieser Kaskade nahezu jedes Korrelationskriterium eingesetzt werden. Jedoch wirkt sich ein sehr hoher Rechenaufwand nachteilig aus. Daher ist die SAD-Funktion

oder die SSD-Funktion sowie deren mittelwertfreie Varianten prinzipiell besser geeignet als aufwändige Korrelationsverfahren, wie z.B. die NKKFMF. Neben den vielen Multiplikationen sind bei der NKKFMF vor allem die Berechnung der Wurzel und die Division für eine Implementierung in Hardware problematisch. Da jedoch nur die Position des Maximums relevant ist, kann die Wurzel durch Quadrierung der NKKFMF zur QNKKFMF eliminiert werden.

$$Q = QNKKFMF(\xi, \eta) = \frac{\left( \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right) \right)^2}{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2} \quad (5.4)$$

Die Implementierung der Division für einen hohen Datendurchsatz ist aufgrund ihrer komplexen Pipelinestruktur sehr aufwändig. Da nicht der Absolutwert, sondern nur die Relation bei der Suche nach dem Maximum des Korrelationswertes relevant ist, kann die Division prinzipiell durch zwei Multiplikationen ersetzt werden. Dies wurde im Anhang [B.2.0.7](#) bereits beschrieben. Für eine Auswertung in Echtzeit ist hierbei die durch die Laufzeit der Multiplikationen entstehende lange Laufzeit ein Problem. Sollen direkt aufeinanderfolgende Werte miteinander verglichen werden, ist eine Erhöhung der Taktrate der Multiplikationen eine praktikable Lösung.

Somit sind nicht normierte Korrelationskriterien einfacher zu handhaben und insbesondere für die Implementierung einer Korrelatorkaskade besser geeignet.

### 5.3 Vergleich der linearen und der hierarchischen blockbasierten Korrelation bei einer Softwareimplementierung

Auch bei einer Softwareimplementierung ist die Einführung des hierarchischen Verfahrens günstig für eine Berechnung von Stereobildern mit großen Disparitätsbereichen. Untersuchungen [\[47\]](#) bei der Implementierung des hierarchischen Verfahrens haben ergeben, dass auch für eine Softwareimplementierung eine erhebliche Ersparnis an Rechenaufwand realisiert werden kann.

In Software ist es für eine schnelle Verarbeitung unerlässlich, nicht zwingend benötigte Rechnungen, z.B. wenn ein Block nicht relevant ist, zu unterlassen und vorher abzufangen. Für eine Hardwareimplementierung sind andere Faktoren wichtig. Vor allem müssen unvorhergesehene Sprünge vermieden werden. So kann es in Hardware günstiger sein, nicht benötigte Werte zu berechnen und später auszublenden, weil dadurch der Ladevorgang eines Shiftregisters unterstützt wird.

In [Tabelle 5.2](#) ist der Vergleich des Rechenaufwandes und die erreichbaren Bildfrequenzen für das lineare und das hierarchische Verfahren dargestellt.

Auf dem Festkomma-DSP BF533 kann mit Hilfe des hierarchischen Verfahrens der Aufwand auf

Bildauffösung [Pixel]		Festkomma-DSP BF533 (600 MHz)			Gleitkomma TS201S 500MHz		
		1024x1024 [Bilder/s]	Zyklen je Zeile	512x384 [Bilder/s]	1024x1024 [Bilder/s]	Zyklen je Zeilen	512x384 [Bilder/s]
Disparitätsbereich [Pixel]		256		128	256		128
lin .	SADMF	0,5	98192	4-5	1,2	375691	10
	SADMF DSP-opt.	1	497341	10	-	-	-
	QNKKFMF	-	-	-	0,25	937755	2
hier.	SADMF	4	141279	24	10	45691	40
	SADMF DSP-opt.	6	97341	48	-	-	-
	QNKKFMF	0,6	973195	4	2	248755	16

Tabelle 5.2: Vergleich der erreichbaren Bildfrequenzen [Bilder/s] und des *Rechenaufwandes* [Zyklen] von Korrelationskriterien beim linearen und beim hierarchischen Verfahren in Software

ca. 1/6 gegenüber dem linearen Verfahren reduziert werden. Dadurch erhöht sich die erreichbare Bildfrequenz für 1024x1024 Pixel a 256 Pixel Disparität bei der SADMF ohne DSP-Optimierung auf 4 Bilder/s und mit DSP-Optimierung auf 6 Bilder/s. Bei der kleineren Bildauflösung von 512x384 Pixel mit 128 Pixel Disparität können damit bis zu 48 Bilder/s erreicht werden.

Durch die besseren Parallelisierungseigenschaften des Gleitkomma-DSPs TS201S kann für die SADMF eine Vervierfachung, für die KKFMF sogar eine Verachtfachung der Leistung erreicht werden.

Der Vergleich verdeutlicht, dass lineare Korrelationsverfahren bei Auflösungen und Disparitätsbereichen dieser Größenordnung für den Einsatz in Prozessoren in eingebetteten Systemen nur bedingt geeignet sind.

## 5.4 Diskussion der Eigenschaften der Implementierungsvarianten für die Stereobildauswertung

In den vorangegangenen Abschnitten wurden zwei Implementierungsvarianten für die Ermittlung der Disparität vorgestellt und ihre Implementierung diskutiert. Unter Berücksichtigung der Korrelationskriterien KKFMF und SADMF ergeben sich somit 4 Varianten (siehe Tabelle 5.3) für den Vergleich der Eigenschaften auf der FPGA-Plattform Stratix EP1S60. Die Eigenschaften der Plattform wurden in den Abschnitten 3.3.2 und 7.3 konkreter beschrieben.

Im Vergleich zum hierarchischen Algorithmus sind andere Stereobildauswertungsverfahren mittels Bildpyramiden stark kontrollflussorientiert. So wird in einer niedrig aufgelösten Bildebene



ein Initialisierungswert für die Disparität gesucht. Dieser wird als Startwert für die Korrelation auf der nächst höher aufgelösten Ebene verwendet. In jeder höher aufgelösten Ebene wird der Messwert dem wahren Wert angenähert und somit die Disparität genauer ermittelt. Dadurch ist vor Beendigung der Korrelation der ersten Ebene keine Information über die Korrelation in der nächsten Ebene bekannt. Die Daten können nicht vorsortiert werden. Zur Parallelisierung kann entweder die Berechnung innerhalb der Ebene parallelisiert oder ein Ansatz ähnlich der Korrelatorkaskade verwendet werden, in dem die Berechnung jeder Ebene als Pipelinestufe betrachtet wird. Problematisch ist hierbei die voraussichtlich unterschiedliche Ausführungsdauer der einzelnen Pipelinestufen.

Durch das regelmäßige Muster der Verarbeitung des in dieser Arbeit vorgestellten hierarchischen Verfahrens ist die Reihenfolge der Eingangsdaten bekannt, wodurch ohne komplexe Sprungvoraussage der Korrelator mittels Pipelining realisiert werden kann.

Ebenso ist das Prinzip der Korrelatorkaskade organisiert. Auch hier ist die Folge der Eingangsdaten und ihre Beziehung zueinander bekannt. So können dieselben Eingangsdaten von vielen Instanzen der SAD-Funktion gleichzeitig verwendet werden.

Beide Verfahren sind auf einen kontinuierlichen Datenstrom ausgerichtet. Durch effektive Ausnutzung der Parallelität sind beide Algorithmen gut in Hardware implementierbar.

Aus den zwei favorisierten Korrelationsfunktionen und den zwei Rechenverfahren ergeben sich vier mögliche Varianten, die in der Tabelle 5.3 miteinander verglichen werden. Die lineare blockbasierte Korrelation wurde auf Basis der SADMF-Funktion mit Hilfe der Korrelatorkaskade implementiert (Variante 1.). Für die 4. Variante wurde das hierarchische Verfahren mit der QNKKFMF implementiert. Beide Varianten wurden auf dem FPGA umgesetzt, getestet und sind für unterschiedliche Anwendungen im Einsatz.

Die 2. Variante beschreibt die lineare Implementierung der QNKKFMF ebenfalls auf Basis der Korrelatorkaskade. Sie wäre in den aktuell größten FPGAs implementierbar. Diese wurde aber nur theoretisch als Vergleichsvariante betrachtet. Im Gegensatz dazu wurde für die 3. Variante das hierarchische Verfahren mittels der SADMF implementiert.

Für jede der vier Varianten wurden Werte für eine Schrittweite bei den Referenzblöcken von 16 Pixeln und von 8 Pixeln ermittelt. Soll der Takt minimal gehalten werden, sind zwei Instanzen der gesamten Verarbeitungskette erforderlich, denen die Daten parallel zugeführt werden. Daher steigen die erforderlichen Ressourcen auf das Doppelte. Wird der minimale Ressourcenverbrauch angestrebt, muss der Takt entsprechend erhöht werden, da die Daten zweimal durch eine Instanz des Korrelators verarbeitet werden müssen. Während beim hierarchischen Verfahren bereits eine Speicherung der Zeile für die Generierung der Ebene implementiert wurde (2048 Takte Verzögerung), muss bei der linearen Implementierung der blockbasierten Korrelation dieser Zwischenspeicher extra hinzugefügt werden.

Betrachtet man die Verarbeitungsgeschwindigkeit aller Varianten, dargestellt in Tabelle 5.3, fällt der deutliche Geschwindigkeitsvorteil der SADMF gegenüber der QNKKFMF auf. Dieser resultiert vor allem aus dem Aufwand der Normierung der QNKKFMF. Die bei der QNKKFMF



	Korrelatorkaskade		hierarchisches Verfahren	
	1. SADMF	2. QNKKFMF	3. SADMF	4. QNKKFMF
Allgemein: Schrittweite: 16 Pixel; Pixeltakt 28 MHz				
Performance max.	152 MHz	–	163 MHz	114 MHz
+ erforderlich	28 MHz	28(56) MHz	56 MHz	56 MHz
Logikaufwand <sup>3</sup>	18 352 LE	ca. 480 000 LE	8390 LE	33 497 LE
Entwurfsaufwand <sup>4</sup>	6 MM	–	4. + 3 MM	24 MM
Skalierbarkeit	fein		grob	
+ Disparität	linear		exponentiell gestaffelt	
+ Tiefenkartenaufkl.	linear		logarithmisch	
+ Genauigkeit	linear mit Z		logarithmisch	
Pipelineaufbau	146 Takte	270 Takte (theoretisch)	50 + 2048 Takte	124 + 2048 Takte
Schrittweite: 8 Pixel; Pixeltakt: 28 MHz minimaler Takt				
Takt	28 MHz	28(56) MHz	56 MHz	56 MHz
Logikaufwand	38 704 LE	ca. 960 000 LE	16 810 LE	67 512 LE
Schrittweite: 8 Pixel; Pixeltakt: 28 MHz minimale Ressourcen				
Takt	56 MHz	56(112) MHz	112 MHz	112 MHz
Logikaufwand	20 651 LE	ca. 480 000 LE	8482 LE	33 591 LE

Tabelle 5.3: Vergleich der hardwareimplementierten Algorithmen zur Stereobildauswertung

erforderliche Division ist nur sehr aufwändig mittels eines 36-stufigen Pipelinings in Hardware zu realisieren und erreicht auf dem Stratix EP1S60 eine maximale Taktrate von 114 MHz. Die für den Dividierer eingesetzten Hardwareressourcen entsprechen mit ca. 6000 Logikelementen (LE) mehr als einem Fünftel des Gesamtbedarfs der QNKKFMF.

Die SADMF kann aufgrund des einfachen Aufbaus und der überwiegenden Verwendung von Additionen deutlich schneller und mit erheblich geringerem Hardwareaufwand realisiert werden. Die Berechnung der mittelwertfreien Pixel und die SAD-Funktion werden getrennt implementiert und somit je nach Implementierung unterschiedlich betrachtet. Je Instanz der SAD ergeben sich 2173 LE. Zusätzlich sind einmalig 1721 LE für die Generierung der mittelwertfreien Pixel erforderlich. Durch den Aufbau der Korrelatorkaskade mittels einer Mehrfachinstanziierung von 7 SAD-Funktionen ergeben sich somit insgesamt 18352 LE. Während eine einzelne SAD-Funktion mit ca. 210 MHz auf einem Stratix realisiert werden kann, ist beim Verbund in der Korrelatorkaskade nur eine Geschwindigkeit von 152 MHz möglich.

Würde die SAD-Funktion jedoch ebenfalls normiert werden, würde der Ressourcenbedarf um ca. 6000 LE steigen. Dadurch ergäbe sich für die Implementierung der Korrelatorkaskade mit 16 Pixel Schrittweite ein Bedarf von ca. 112 000 Logikelementen. Dies entspräche dem Doppelten der verfügbaren Logikelemente auf der verwendeten Plattform.

Die Skalierbarkeit der Korrelatorkaskade ist durch die vollständig parametrisierte Implemen-

tierung gut realisierbar. Durch eine Anpassung, z.B. eine Vergrößerung des Messbereichs in  $Z$ , ändert sich der Ressourcenverbrauch nur linear. Beim hierarchischen Verfahren können weitere Ebenen hinzugefügt werden, ohne dass sich der Rechenaufwand erhöht. Dadurch können sehr große Disparitätsräume vermessen werden. Begrenzt ist der Disparitätsbereich jedoch durch die Auflösung der Kameras. Die am niedrigsten aufgelöste Ebene kann minimal die Breite des Suchbereiches einer Ebene erreichen. Für kleinere Disparitätsräume ist der Aufwand für das hierarchische Verfahren zu hoch.

Ein deutlicher Vorteil des hierarchischen Verfahrens ergibt sich für die nachfolgenden Algorithmen durch die Vereinfachung der Suchmuster. Während sich bei der herkömmlichen Stereobildauswertung die Größe eines Objektes sowohl in der Disparitätskarte als auch in der Tiefenkarte stark verändert, schwankt beim hierarchischen Verfahren die Größe der Abbildung des Objektes immer in einem bekannten Größenbereich, der durch die Bildpyramide vorgegeben ist. Weiterhin ist es durch die Effektivität des Verfahrens möglich, hier die QNKKFMF einzusetzen. Verwendet man die SADMf statt der QNKKFMF, wird das Verfahren sehr kompakt und kann bereits auf kleinen FPGAs implementiert werden.

### 5.5 Fazit

Beide in diesem Kapitel vorgestellten Verfahren nutzen die Parallelität der Operatoren und der Beziehungen zwischen den Daten, um eine Implementierung für eine kontinuierliche Verarbeitung von Kameradaten mit einem hohen Datendurchsatz in Hardware zu ermöglichen. Eine Umsetzung der Verfahren in einem FPGA der Familie Stratix EP1S60 kann realisiert werden.

Während das hierarchische Verfahren auf spezielle Messbereiche, die vorwiegend in der Stereophotogrammetrie zu finden sind, angewendet wird, ist die Implementierung der SAD-Funktion in der Korrelatorkaskade für eine Vielzahl von Aufgaben der Bildverarbeitung verwendbar. Aufgrund der guten Skalierbarkeit kann das Verfahren schnell an neue Gegebenheiten angepasst werden. Die resultierende Implementierung ermöglicht die schnelle Auswertung von Signalen auf einem geeigneten Schaltkreis. Bei der Optimierung einer Schaltung bezüglich Geschwindigkeit und Ressourcen verhalten sich beide Größen im Allgemeinen indirekt proportional zueinander. Durch die aktuelle technische Entwicklung bei den FPGAs stehen auch im Low-Cost-Bereich Schaltkreise mit ausreichender Rechenleistung für eine Bildverarbeitung mit mehr als 25 Bildern/s zur Verfügung.

Bei dem hierarchischen Verfahren wird die Auflösung des Bildes angepasst, sodass das Berechnungsverfahren über das gesamte Bild identisch bleiben kann und dennoch die Objekte in der für sie erforderlichen Auflösung bearbeitet werden können. Dadurch ergibt sich eine deutliche Reduktion des Rechenaufwandes auf allen Implementierungsplattformen. Wird das Verfahren mit begrenzter Schrittweite und begrenzter Blockgröße angewendet, so kann es für Messaufgaben mit einem 3D-Messbereich, bei dem eine Dimension deutlich größer ist als die anderen, gut an neue Problemstellungen angepasst werden. Verwendet man eine Kombination aus der Korrelatorkaskade und dem hierarchischen Verfahren, kann für jedes Pixel eines Bildes

ein 3D-Wert effektiv und schnell berechnet werden. Wendet man weiterhin die Auswertung der Ebenen über die Position des Maximums an, so kann das hierarchische Verfahren als datenflussorientierte Implementierung klassischer Pyramiden-Verfahren verwendet werden. Es kann dabei theoretisch eine deutlich höhere Auflösung erreichen. Der dafür erforderliche Hardwareaufwand (ca. 300 000 LE) wird vom derzeitigen Versuchsaufbau nicht abgedeckt. FPGAs der aktuellen Generation, wie der Stratix IV, bieten jedoch die erforderliche Leistungsfähigkeit.

Die Ergebnisse dieses Kapitels bestätigen die Annahme, dass eine datenflussorientierte Implementierung in Hardware sehr vorteilhaft ist. Es konnten beide Verfahren mit starker Parallelisierung umgesetzt werden. Auf dem in Kapitel 7 beschriebenen Versuchsaufbau wurden mehrere Implementierungen getestet. Das hierarchische Verfahren mit der QNKKFMF konnte für Bildfrequenzen von 50 Bildern/s realisiert werden. Während das hierarchische Verfahren mit der SADMf Stereobilder mit ca. 75 Bilder/s verarbeiten kann. Mit der linearen blockbasierten Korrelation mittels Korrelatorkaskade kann mit der SADMf eine Bildfrequenz bis zu 150 Hz erreicht werden. Der Unterschied zwischen den beiden Verfahren resultiert aus der für die Verarbeitung der Bildpyramide erforderlichen doppelten Taktfrequenz.

Der Aufwand an Ressourcen beim hierarchischen Verfahren konnte gegenüber dem linearen blockbasierten Verfahren um  $2/3$  reduziert werden.



# Kapitel 6

## Praktische Erprobung und Vergleich der Algorithmen

In den vorangegangenen Kapiteln standen vorwiegend der Rechenaufwand und die Stereobildauswertung in Echtzeit im Mittelpunkt. In diesem Kapitel werden die Verfahren der linearen blockbasierten Korrelation mittels der Korrelatorkaskade und die hierarchischen Verfahren bezüglich ihrer Bildverarbeitungseigenschaften miteinander verglichen.

Für den Vergleich werden die gleichen Kombinationen von Verfahren und Korrelationsfunktionen verwendet, die im Abschnitt 5.4 benutzt wurden.

1. Korrelatorkaskade mit SADMF (Diagrammbeschriftung: SADMF)
2. Korrelatorkaskade mit QNKKFMF (Diagrammbeschriftung: QNKKFMF)
3. Hierarchisches Verfahren mit SADMF (Diagrammbeschriftung: hier. SADMF)
4. Hierarchisches Verfahren mit QNKKFMF (Diagrammbeschriftung: hier. QNKKFMF)

Zum Vergleich werden jeweils alle 4 Verfahren auf Stereobilder im Außen- wie im Innenbereich angewendet und dabei die beiden in Kapitel 4 vorgestellten Szenarien Spurwechselassistent und Hinderniserkennung für fahrerlose Transportfahrzeuge berücksichtigt.

Die Berechnungen werden mit einer Hardwareemulation in Software durchgeführt, da in Hardware die Ausgabe vieler Zwischenergebnisse nicht möglich ist. Die Berechnungen erfolgen in identischer Weise zu der Verarbeitung auf der Hardware. Hierbei sind insbesondere die begrenzten Bitbreiten relevant. Die Hardwareemulation wurde in Matlab realisiert. Der Rechenaufwand wurde hierbei nicht betrachtet, so konnte auch die Korrelatorkaskade mit der linearen blockbasierten Korrelation auf ihre Bildverarbeitungseigenschaften getestet werden.

Damit kann jeder Messbereich unabhängig von verfügbaren Ressourcen realisiert werden. Auf die realen Einschränkungen bei der Realisierung wurde bereits in Kapitel 5 eingegangen. Bei der Auswahl für eine konkrete Anwendung sind die dortigen Aussagen weitgehend begrenzend.

Die Untersuchungen in diesem Kapitel erhalten mehrformale Aussagen für die Anwendung der Algorithmen bei gleichen Messbereichen in Z-Richtung.

Es wurden die Eigenschaften der Verfahren bezüglich der Vermessung in Z-Richtung untersucht. Dazu wurden Aufnahmen im Fernbereich (10 - 150 m) und Aufnahmen im Nahbereich (1 - 20 m) untersucht. Vereinzelt starke Ausreißer wurden nicht berücksichtigt und mit Hilfe eines Filters ausgeschlossen. Durch das im Kapitel 7 beschriebene statistische Clusterverfahren erfolgt eine ähnliche Prozedur im realen Einsatz. Für die Untersuchung des Messfehlers in Z-Richtung wurde das Ergebnis der Disparität durch eine Subpixelinterpolation auf 1/8 Pixel verbessert.

## 6.1 Vermessung im Fernbereich

Zur Vermessung des Fernbereichs wurde ein Kamerasystem im Normalfall der Stereophotogrammetrie aus zwei CMOS-Grauwert-Kameras MV-D1024 der Firma Photonfocus verwendet. Die Auflösung der Kameras beträgt 1024x1024 Pixel a 10 Bit. Es wurden jedoch nur 8 Bit für die Verarbeitung in der Hardwareemulation verwendet, da die Bilder als Bitmaps gespeichert wurden. Die Basisbreite  $b$  des Systems beträgt 70 cm und die Brennweite der Objektive  $f=25$  mm.

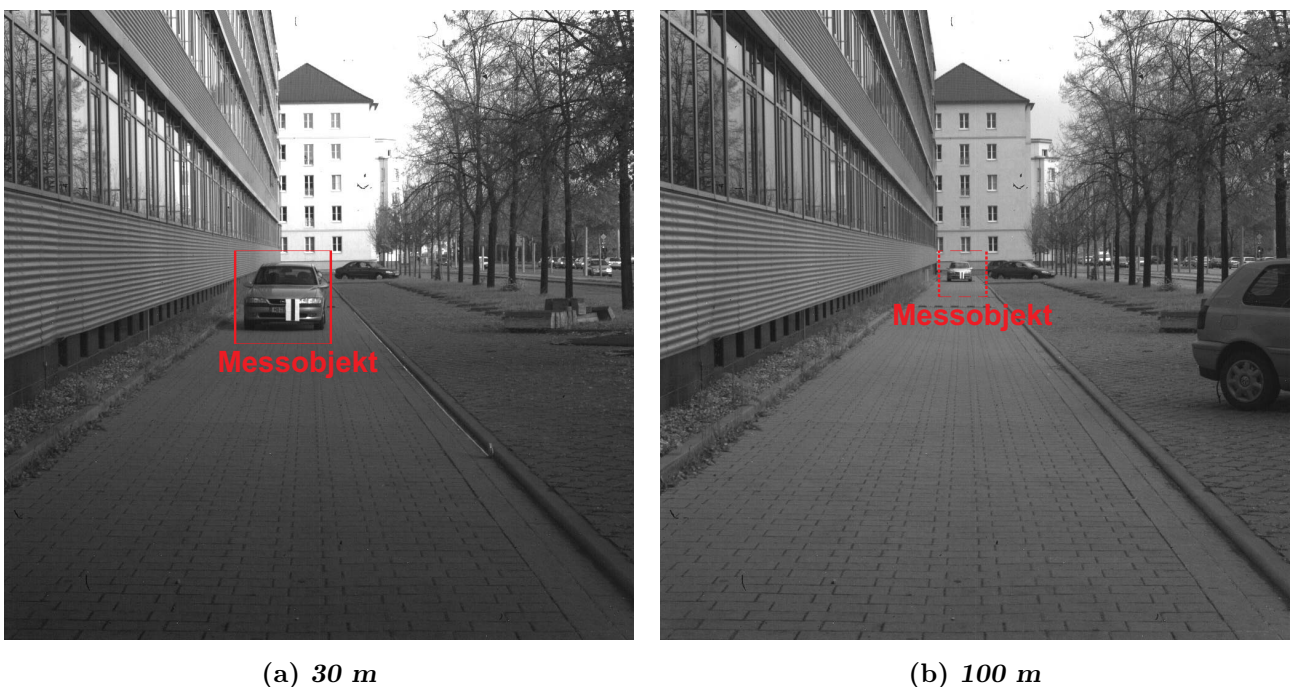


Abbildung 6.1: Linkes Kamerabild bei der Vermessung des Fahrzeugs in zwei Entfernungen

Es wurden mehrere Serien von Aufnahmen durchgeführt. Die erste Serie wurde mit einem Fahrzeug durchgeführt, welches sich im Bereich von 150 m - 30 m den Kameras in Schritten von einem Meter und von 30 - 10 m in Schritten von einem halben Meter nähert. Da sich ein Fahrzeug jedoch über einen gewissen Tiefenbereich erstreckt, können davon keine exakten

Messungen abgeleitet werden. Daher wurde eine zweite Serie mit einer Messtafel mit einem senkrechten schwarzen Streifen durchgeführt.

### 6.1.1 Vermessung der Entfernung des Fahrzeugs

In Abbildung 6.1 ist das Bild der linken Kamera mit dem Fahrzeug in einer Entfernung von 30 m und in einer Entfernung von 100 m dargestellt. Die Größe des Objektes verhält sich indirekt proportional zur Messentfernung. In 150 m Entfernung beträgt die Breite der Abbildung 20 Pixel, in 100 m Entfernung 39 Pixel, in 30 m Entfernung 132 Pixel und in 10 m Entfernung 385 Pixel.

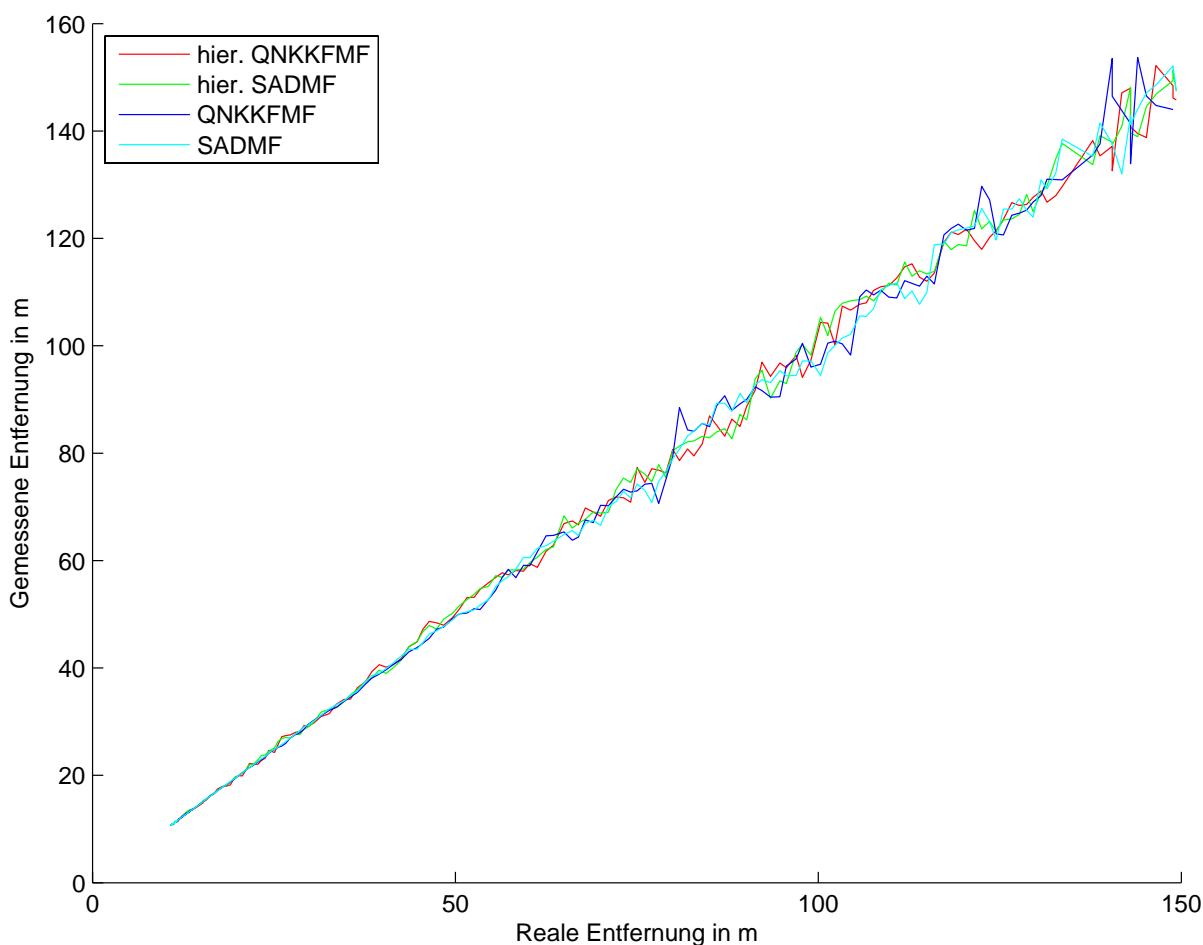


Abbildung 6.2: Verlauf der Entfernungsbestimmung eines Fahrzeugs von 10 - 150 m mit dem hierarchischen und dem linearen Verfahren

Das Diagramm in Abbildung 6.2 zeigt den Verlauf des Mittelwertes der Messergebnisse auf dem gesamten Fahrzeug über der "realen Entfernung". Die reale Entfernung des Fahrzeugs wurde mit Hilfe eines Laserentfernungsmessgerätes Lasertape FG21-HA [75] ermittelt. Für das Gerät wird ein Fehler in von  $\pm 5 \text{ cm}$  in der Betriebsart Distanzmesser angegeben. Die korrekte Ausrichtung des Messgerätes auf das Ziel in Entfernungen über 120 m war sehr problematisch. Daher wurde

ein weißes Feld zum Anvisieren, welches zur Überprüfung der Messergebnisse der Kameras mit einem schwarzen Streifen versehen wurde, an der Front des Fahrzeugs angebracht (siehe 6.1).

Insgesamt folgen die Messergebnisse der Sollentfernung über den gesamten Messbereich sehr gut. Bis in eine Entfernung von 75 m liegen die Kurven der untersuchten Verfahren sehr dicht beieinander. Ab dieser Entfernung steigt der Fehler der Messung vor allem bei den linearen Verfahren an. Die Erhöhung des Fehlers ist auch bei der Vermessung mit den Varianten auf Basis des hierarchischen Verfahrens sichtbar, wenn auch nicht in so großem Umfang.

In den Diagrammen in Abbildung 6.3 ist der absolute Fehler über dem Sollwert dargestellt. Für den Vergleich der beiden Verfahren wird zusätzlich der relative Fehler gegenüber dem Sollwert in Abbildung 6.3 dargestellt. Zur Übersichtlichkeit wurden die Kurven der linearen und die der hierarchischen Verfahren getrennt abgebildet. In Abbildung 6.3(a) ist das leichte Ansteigen des absoluten Fehlers über den gesamten Messbereich bei dem hierarchischen Verfahren erkennbar. Dies wird durch die annähernde Konstanz des relativen Fehlers über den gesamten Messbereich unterstützt (siehe Abbildung 6.4(a)).

In Abbildung 6.5 ist der theoretische Verlauf des relativen Fehlers für das hierarchische Verfahren zur Entfernungsmessung dargestellt. Im Vergleich zum Fehler im realen Fall in Abbildung 6.4(a) ist eine markante Ähnlichkeit zu erkennen. Der relative Fehler der Messung liegt in einem nahezu konstanten Fehlerschlauch über den gesamten Messbereich.

Der absolute Fehler des linearen Verfahrens ist, wie in Abbildung 6.3(b) ersichtlich, bis zu einer Entfernung von 50 m sehr niedrig, danach schwankt er stark und erreicht bei 75 m sein vorläufiges Maximum. Bei Betrachtung des relativen Fehlers wird das kontinuierliche Ansteigen des Fehlers über den gesamten Messbereich und auch die deutliche Steigerung des Fehlers ab einer Entfernung von 50 m deutlich.

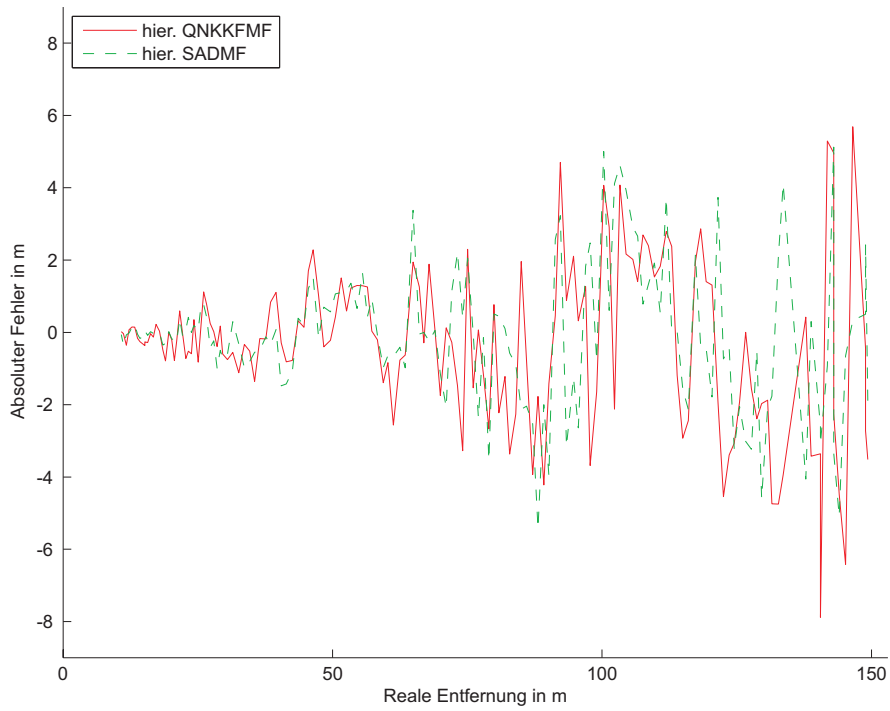
Eine wesentliche Auswirkung des verwendeten Korrelationskriteriums konnte nicht festgestellt werden. Die Anzahl der validen Punkte war bei der QNKKFMF geringfügig höher und der Fehler im Mittel geringfügig darunter. Aschwanden stellt eine Empfindlichkeit der SADMF insbesondere für verrauschte Bilder und Signale fest. Daher wäre eine Untersuchung über die Auswirkungen von Nebel bezüglich der Vermessung mit den Verfahren interessant. Diese wurde aber im Rahmen der Arbeit aus Aufwandsgründen nicht durchgeführt.

### 6.1.2 Vermessung der Entfernung der Messtafel

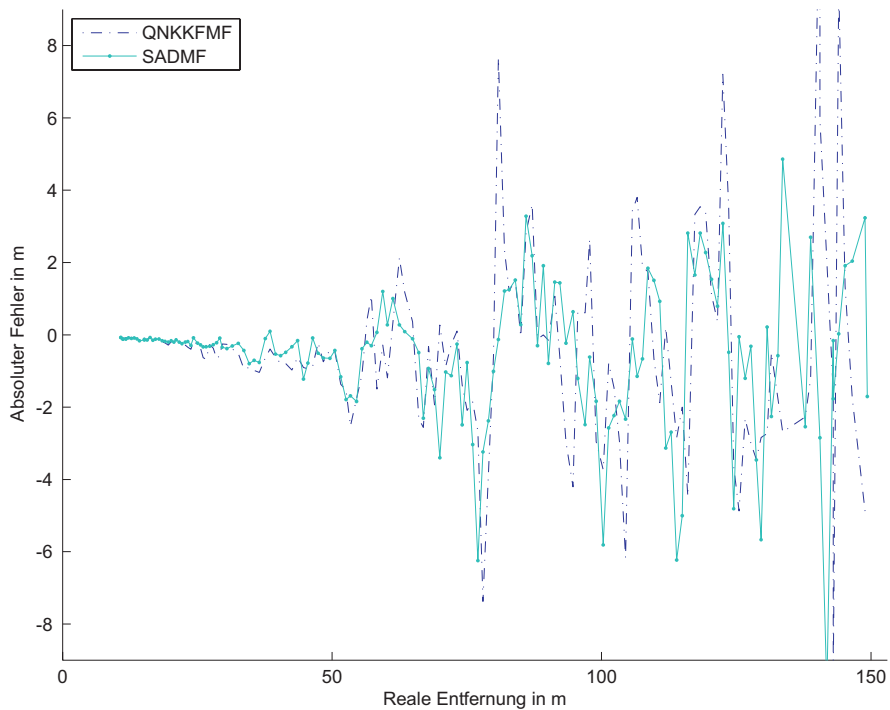
Als Messtafel wurde eine weiße Gipskartonplatte (100 cm x 150 cm) mit einem 50 cm breiten schwarzen Streifen verwendet. Die Tafel war etwas schmaler als das Fahrzeug (10 m - Breite 198 Pixel; 30 m - Breite 68 Pixel; 100 m - Breite 21 Pixel; 144 m - Breite 15 Pixel). Die beiden Kanten der Messtafel können wie in Abbildung 6.6 über den gesamten Messbereich gut wahrgenommen werden.

Am Aufnahmeort, ein Parkplatz der Universität, variierten an diesem Tag die Lichtverhältnisse sehr stark. Dadurch ist die Versuchsreihe auch ein Test für die Wirkung verschiedener



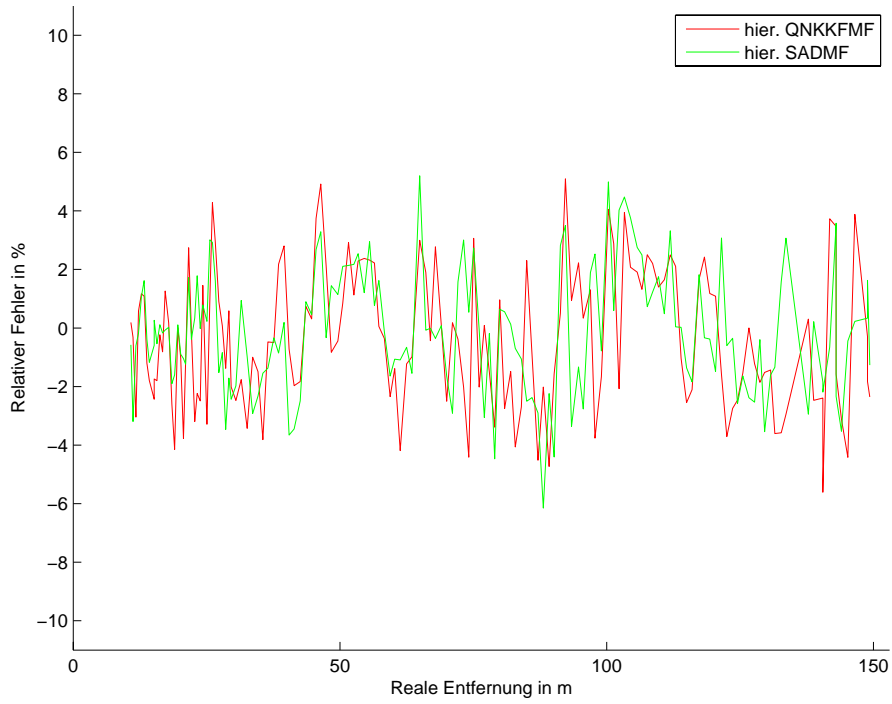


(a) *Absoluter Fehler: hierarchische Verfahren*

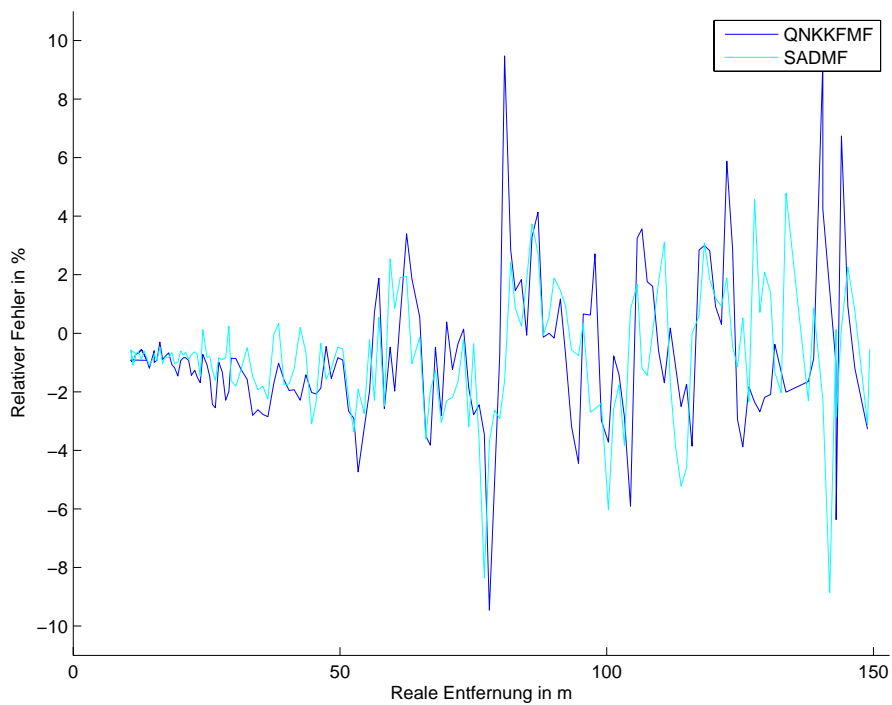


(b) *Absoluter Fehler: lineare Verfahren*

Abbildung 6.3: Absoluter Fehler beim Vermessen der Position eines Fahrzeugs im Bereich von 10 - 150 m



(a) *Relativer Fehler: hierarchische Verfahren*



(b) *Relativer Fehler: lineare Verfahren*

Abbildung 6.4: Relativer Fehler beim Vermessen der Position eines Fahrzeugs im Bereich von 10 - 150 m

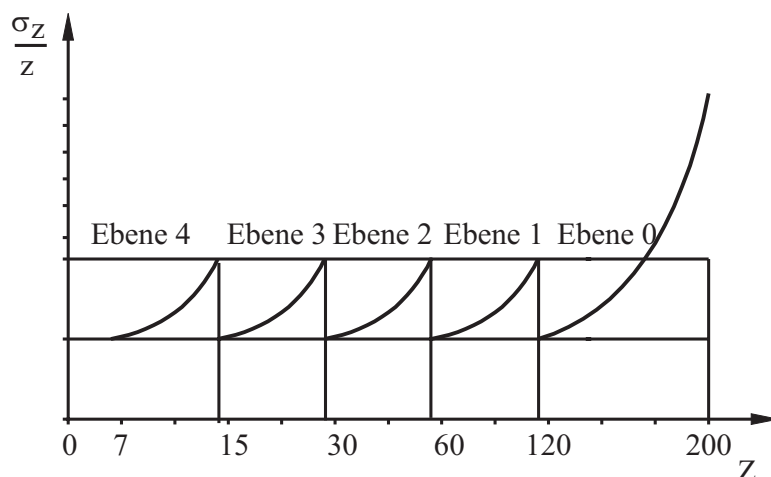


Abbildung 6.5: Theoretischer Verlauf des relativer Fehler beim Vermessen mit dem hierarchischen Verfahren im Fernbereich

Helligkeiten auf die Entfernungsvermessung.

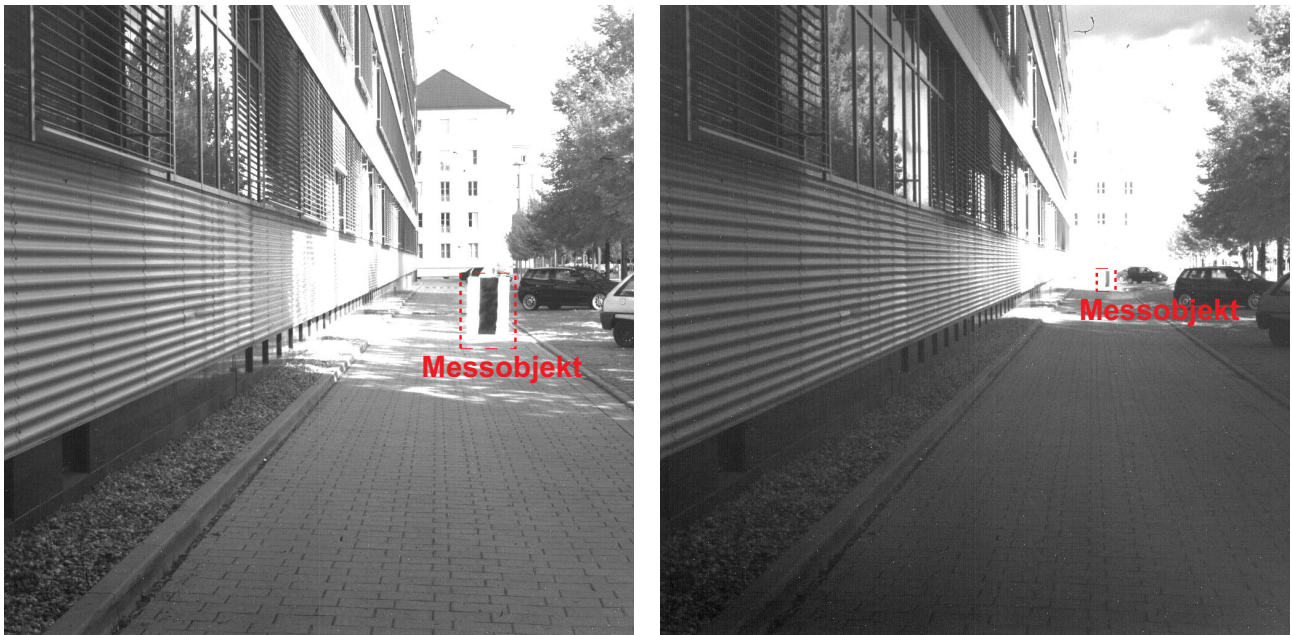
Aufgrund der manuellen Platzierung der Tafel konnte der Sollwert der Entfernung mit Hilfe eines Maßbandes ausreichend genau ermittelt werden. Der Messfehler in der Handhabung sinkt dadurch im Vergleich zum Laserentfernungsmessgerät insbesondere in Entfernungen oberhalb von 100 m, da ein stabiles Anvisieren in dieser Entfernung selbst mit Stativ nur schwer möglich war.

Anschließend wurden nach der Berechnung der Tiefenkarten die Spalten der Tiefenkarte, in denen die Messtafel abgebildet ist, gegen die Referenzentfernung bewertet (siehe Abbildung 6.7(a)). Das synthetische Objekt überdeckte im Nahbereich mehr als 1000 3D-Punkte in allen Spalten, während am Ende des Messbereiches (bei 150 m) gerade noch 40 3D-Punkte ermittelt wurden.

In Abbildung 6.7(a) zeigt der Verlauf der Entfernungsmessung ebenfalls, dass alle Verfahren die Entfernung sicher ermitteln können. Jedoch ist ab einer Entfernung von ca. 60 m ein veränderter Verlauf der hierarchischen Verfahren erkennbar. Dies resultiert aus einer leichten Dejustierung der Kameras. Es wurde kein Kalibrierverfahren angewendet. Die beiden Verfahren reagieren unterschiedlich auf diese Dejustierung. Aufgrund des geringeren Disparitätsbereiches in den einzelnen Ebenen reagiert das hierarchische Verfahren zunächst empfindlicher.

Betrachtet man den absoluten Fehler (Abbildung 6.8<sup>1</sup>) der Verfahren im Vergleich, so fällt auf, dass der Fehlerverlauf verhältnismäßig unabhängig vom Korrelationskriterium ist, jedoch stark von dem verwendeten Rechenverfahren abhängt. Sowohl beim hierarchischen Verfahren als auch bei dem linearen Verfahren ist deutlich die quadratische Abhängigkeit des Fehlers von der Entfernung ersichtlich (siehe Abschnitt 2.3.4). Ohne Kalibrierung ist die Auswertung des relativen Fehlers hier nicht sinnvoll.

<sup>1</sup>In diesem Kapitel sind nur die absolut erforderlichen Diagramme dargestellt. Alle Serien der Diagramme der diskutierten Messungen befinden sich im Anhang C.



(a) 30 m

(b) 100 m

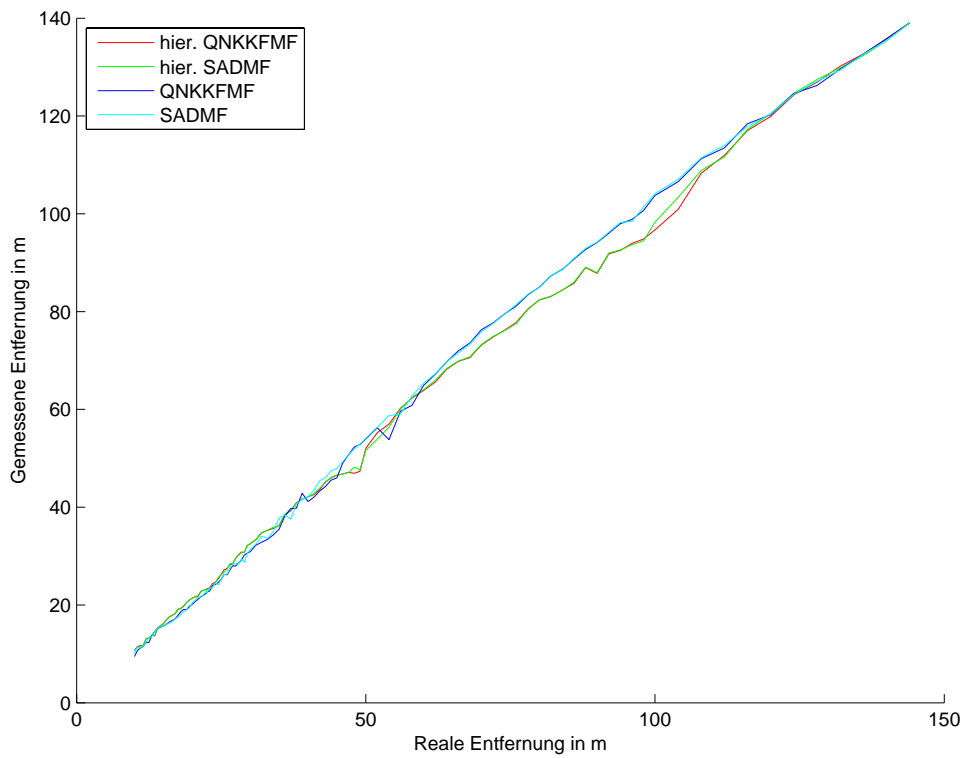
Abbildung 6.6: Linkes Bild der Messtafel in 30 und 100 m Entfernung

### 6.1.3 Auswirkung der vereinfachten Kalibrierung

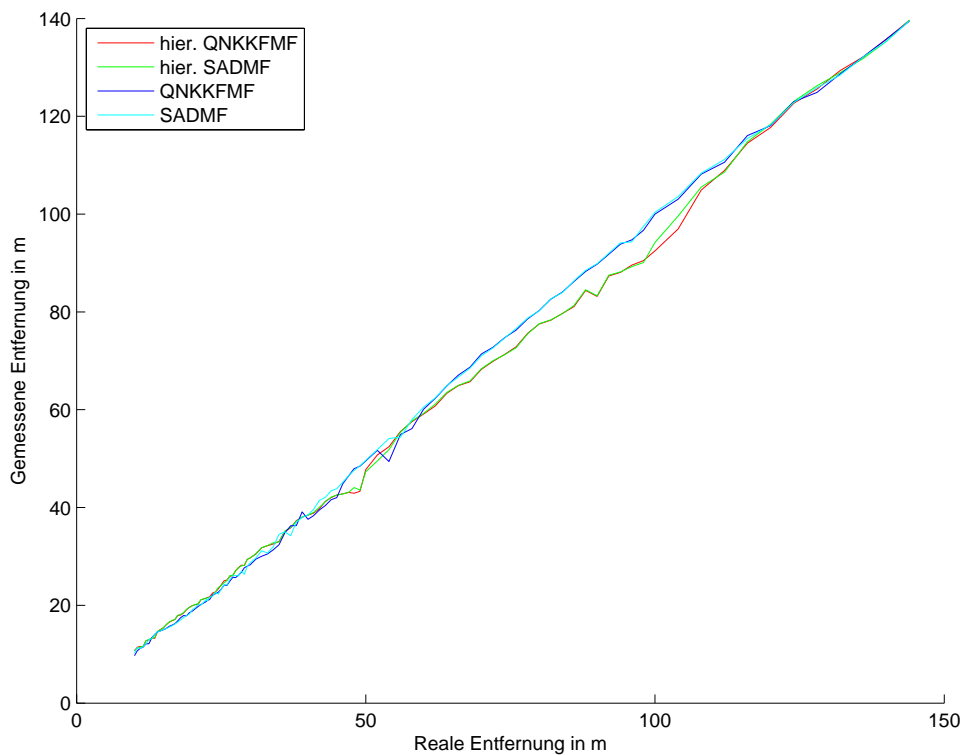
Zum Ausgleich der Dejustierung der Kameras wurde das vereinfachte Kalibrierverfahren aus Anhang D verwendet. Hierbei werden an 3 Punkten die Soll- und Ist-Werte ermittelt. Aus diesen Werten werden 3 Koeffizienten für eine quadratische Korrektur der Messergebnisse ermittelt. Diese Koeffizienten beinhalten bereits die Kamerakonstante und die Basisbreite, sodass eine separate Bestimmung dieser Größen entfällt [48]. Anschließend wird mit Hilfe dieser Koeffizienten eine Korrektur der Messergebnisse in Abhängigkeit von der Entfernung berechnet.

In Abbildung 6.7(b) ist der kalibrierte Verlauf der Entfernung dargestellt. Im Vergleich zu den unkalibrierten Messergebnissen folgt der Verlauf der Mittelwerte über der Messtafel dem Sollwert deutlich besser. Die Einschnitte in den Kurvenverlauf durch starke Ausreißer werden zwar reduziert, können aber nicht ausgeglichen werden. Sie resultieren aus den ungünstigen Beleuchtungsbedingungen in einigen Bildern. Dies verdeutlicht die Notwendigkeit einer Helligkeitskorrektur oder die Verwendung von HDR-Kameras zur Abdeckung eines großen Dynamikumfangs.

Der absolute Fehler (Abbildung 6.8) konnte im Vergleich zu den unkalibrierten Messungen deutlich reduziert werden. Er liegt abgesehen von einigen Ausreißern bei allen Verfahren unterhalb von  $\pm 2$  m. Der relative Fehler ist in Abbildung C.7 abgebildet. Es zeigt sich, dass der Fehler abgesehen vom unmittelbaren Nahbereich (bis 25 m) bei den hierarchischen Verfahren stark reduziert werden konnte. Die linearen Verfahren zeigen bis in eine Entfernung von 50 m einen ungünstigeren Verlauf des relativen Fehlers.

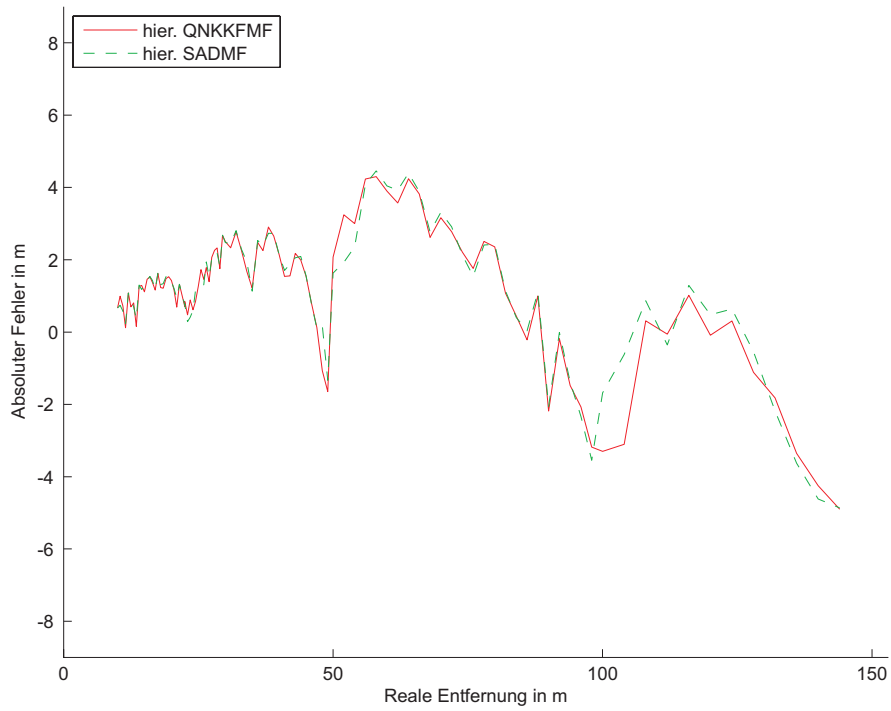


(a) *unkalibriert*

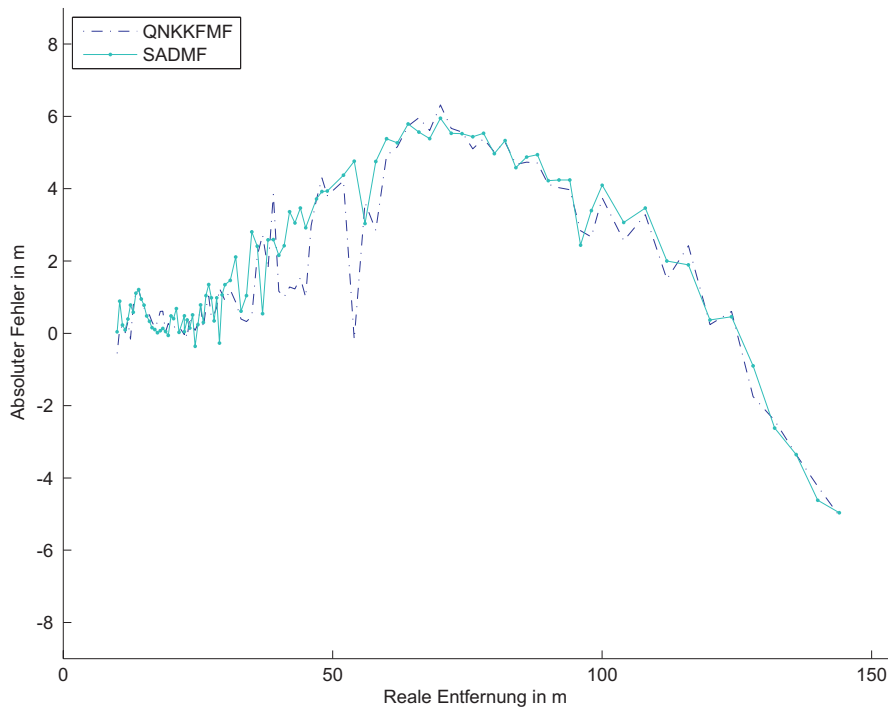


(b) *kalibriert*

Abbildung 6.7: Vergleich der Entfernungvermessung der Messtafel im kalibrierten und nicht kalibrierten Fall

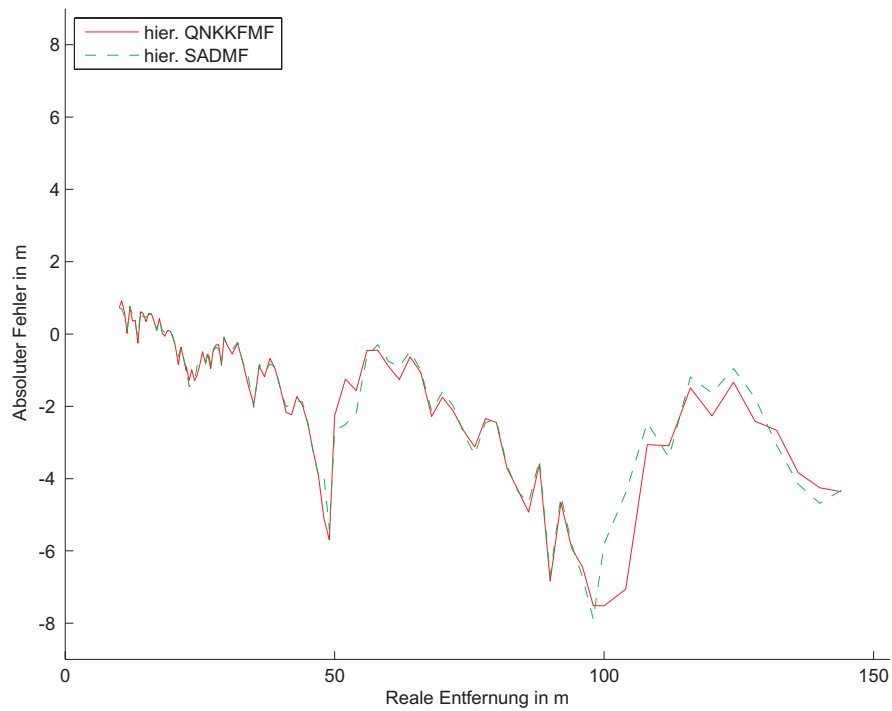


(a) *Absoluter Fehler: hierarchische Verfahren*

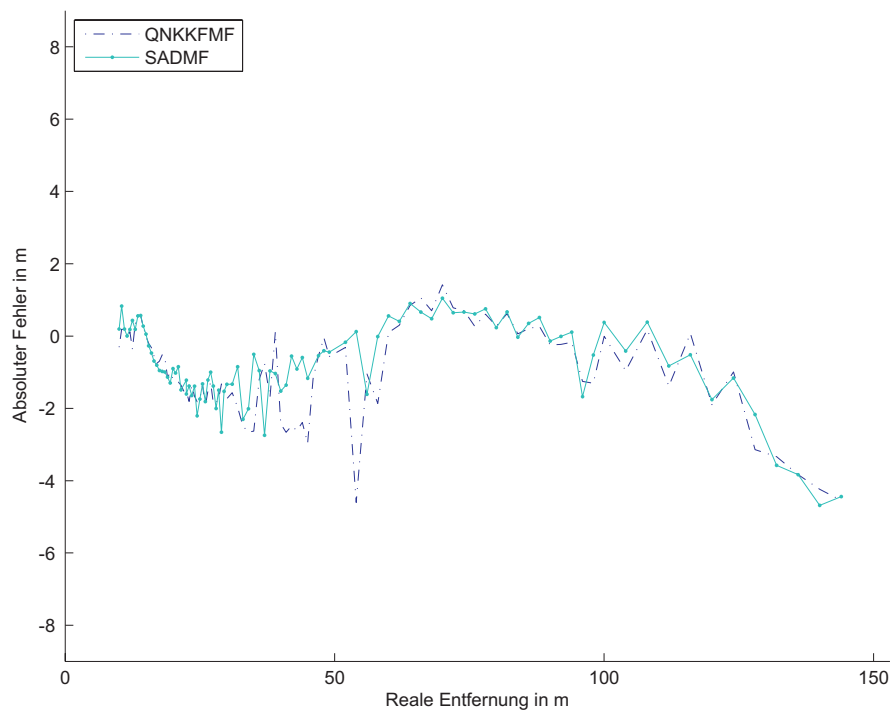


(b) *Absoluter Fehler: lineare Verfahren*

Abbildung 6.8: Absoluter Fehler des nicht kalibrierten Falles der Vermessung der Position der Messtafel im Bereich von 10 - 150 m



(a) *Absoluter Fehler: hierarchische Verfahren (kalibriert)*



(b) *Absoluter Fehler: lineare Verfahren (kalibriert)*

Abbildung 6.9: Absoluter Fehler des kalibrierten Falles der Vermessung der Position der Messtafel im Bereich von 10 - 150 m



(a) *Kamerasystem 1: Photonfocus*



(b) *Kamerasystem 2: Bumblebee 2*

Abbildung 6.10: verwendete Kamerasysteme

## 6.2 Vermessung im Nahbereich

Im Nahbereich ist der zu überwachende Öffnungswinkel des Messsystems mit  $\geq 90^\circ$  deutlich größer als im Fernbereich mit  $\leq 30^\circ$ . Durch die größeren Öffnungswinkel des Messsystems sind entsprechend kleinere Brennweiten erforderlich, bei denen deutlich stärkere Verzerrungen auftreten.

Im Folgenden werden zur Untersuchung der Auswirkung der Brennweite auf die einzelnen Verfahren folgende Kamerasysteme zur Aufnahme verwendet:

1. **Kamerasystem 1** bestehend aus 2 Photonfocus-Kameras MV-D1024, angesteuert über Kameralink mit einer Basisbreite von  $b = 6,5$  cm und dem Objektiv C21211KP der Firma Cosmicar mit einer Brennweite von  $f = 12,5$  mm, montiert auf zwei Goniometern und einem Hebe- und einem Drehtisch (siehe Abbildung 6.10(a))
2. **Kamerasystem 2** Bumblebee 2 [74] aufgebaut als starres System mit festmontierten Linsen mit einer Basisbreite von  $b = 12$  cm und einer Brennweite von  $f = 6$  mm; enthält Schaltkreis zur Rektifizierung in Echtzeit (siehe Abbildung 6.10(b))

In Abbildung 6.11 ist die Wirkung der unterschiedlichen Öffnungswinkel der Kameras gut zu erkennen. Die starren Objektive des Kamerasystems 2 haben mit ihrer kleineren Brennweite von 6 mm einen wesentlich größeren Öffnungswinkel als das Objektiv mit 12,5 mm. Aufgrund der

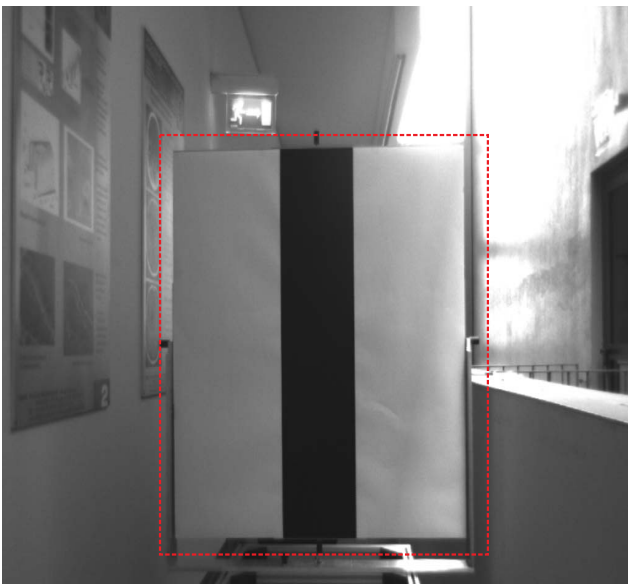




(a) *Photonfocus: Abstand 2,1 m*



(b) *Photonfocus: Abstand 10 m*



(c) *Bumblebee 2: Abstand 2,1 m*



(d) *Bumblebee 2: Abstand 10 m*

Abbildung 6.11: Vergleich der Auswirkungen von Basisbreite und Brennweite auf die Messung anhand von zwei Kamerasystemen im Nahbereich: 2 Photonfocus  $b = 6,5$  cm,  $f = 12,5$  mm (a) (b); Bumblebee 2 (rektifiziert)  $b = 12$  cm,  $f = 6,3$  mm (c) (d)

deutlich geringeren Sensorgröße von 2,2 mm im Vergleich zu ca. 11 mm bei den Photonfocus-Kameras fällt der für die Messung entscheidende Formatwinkel für das Kamerasystem 2 mit  $43^\circ$  deutlich geringer aus als der Formatwinkel von  $94,23^\circ$  des Kamerasystems 1. Der Formatwinkel gibt den Sichtbereich einer Kamera an. Er wird durch die Sensorgröße und die Brennweite bestimmt (siehe Abschnitt [A.1](#) und Abbildung [A.1](#)).

Im Folgenden werden zunächst mit beiden Kamerasystemen Bildfolgen in einem Messbereich von 1 - 10 m aufgenommen und mit den am Anfang des Kapitels beschriebenen 4 Verfahren ausgewertet. Das Stereokamerasystem Bumblebee 2 ist primär darauf ausgelegt, die eigenen Stereobilder direkt unter Verwendung einer eingebauten Hardwareunterstützung in eine Tiefenkarte umzurechnen. Zuerst werden die Bilder mit den vier eigenen Verfahren ausgewertet. Anschließend erfolgt ein Vergleich mit den Ergebnissen der im Bumblebee 2 integrierten Auswertelgorithmen.

### 6.2.1 Auswertung von Aufnahmen verschiedener Kamerasysteme mit den Verfahren 1 - 4

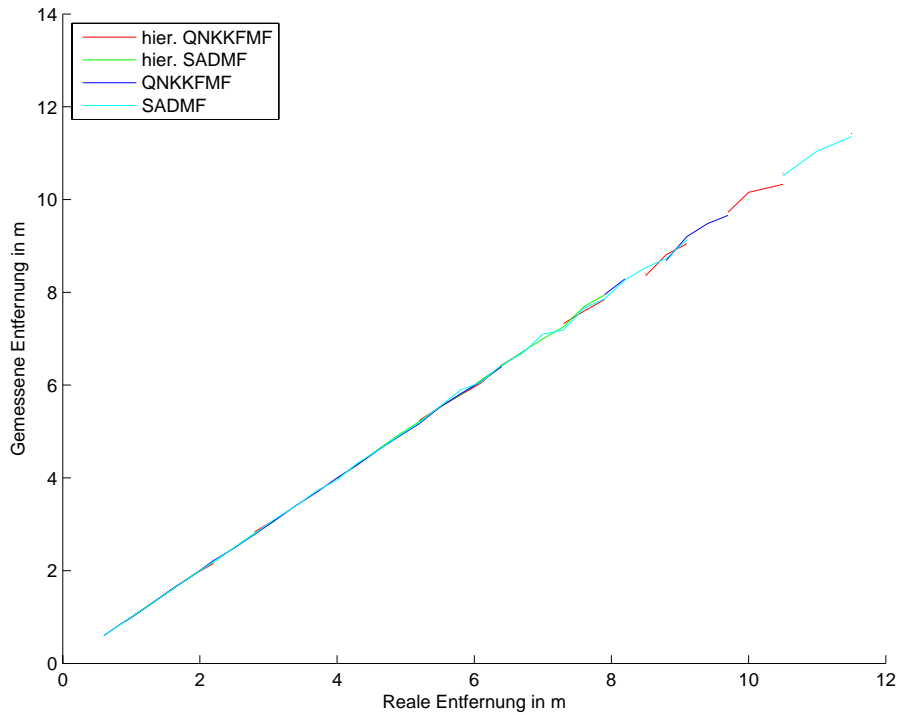
Die Auswertung der Vermessung mit dem Kamerasystem 1 in Abbildung [6.12\(a\)](#) zeigt, dass insbesondere die beiden linearen Verfahren sehr gute Ergebnisse im Nahbereich liefern. Die Abweichung der Resultate der hierarchischen Verfahren, die im Fernbereich sehr gut abgeschnitten haben, zeigt deutliche Schwächen in der Genauigkeit, sodass einige Ergebnisse von der Weiterverarbeitung ausgeschlossen wurden.

In Abbildung [6.12](#) ist jeweils die Gemessene Entfernung über der Realen Entfernung dargestellt. Es zeigt sich, dass die Verläufe bis zu einer Entfernung von 5 m gut mit der realen Entfernung übereinstimmen. Die Messung auf dem Kamerasystem 1 folgt dem Sollwert weiterhin sehr gut, während bei der Messung mit dem Kamerasystem 2 der Messfehler deutlich ansteigt. Ein detaillierterer Vergleich der Messfehler wird im Anhang [C](#) gegeben.

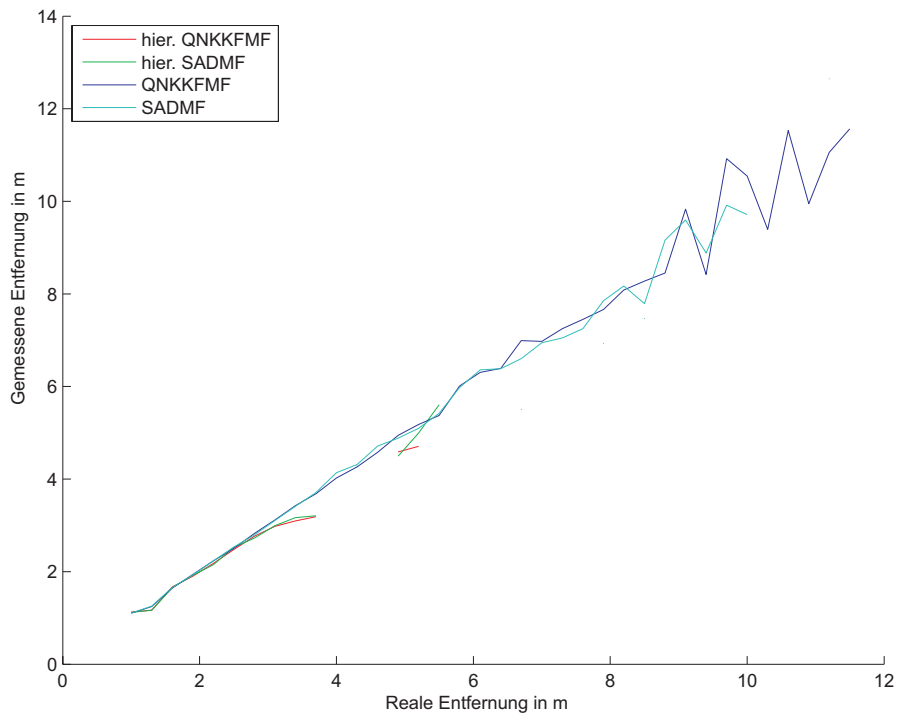
Bei der Auswertung der Bilder des Bumblebeesystems wurden die vom System rektifizierten Bilder verwendet, da eine zuverlässige Entfernungsvermessung bei einer Brennweite von  $f=6$  mm ohne Berücksichtigung der radiometrischen Verzeichnung nicht mehr möglich ist. Dies wird im Abschnitt [4.2.2.1](#) detaillierter betrachtet.

Die Messwerte der Entfernung, die mit dem linearen Verfahren ermittelt worden sind, folgen dem Sollwert in den ersten 5 m noch recht gut, danach erhöht sich der Fehler zunehmend. Dies resultiert aus der Rektifizierung, die immer noch deutliche Fehler in der Größenordnung von mehreren Pixeln enthält. Ab einer Entfernung von 5 m liegen diese Fehler im Bereich von 2 - 3 Prozent der Disparität, wodurch sich deutlich schlechtere Messergebnisse gegenüber dem Kamerasystem 1 ergeben.

Die Auswertung der mit dem Bumblebee 2 aufgenommenen Bilder mit dem hierarchischen Verfahren zeigt bereits am Anfang des Messbereichs einen Fehler im Bereich von 10 Prozent. Da jedoch die beiden Stereoaufnahmen im Bereich von 1 und 1,3 m fehlerhaft sind, können



(a) *Kamerasystem 1: Photonfocus*



(b) *Kamerasystem 2: Bumblebee 2*

Abbildung 6.12: Mittelwert der Entfernung im Nahbereich bei der Vermessung mit dem Kamerasystem 1 (Messbereich: 0,6 - 10 m)

diese als Fehlmessung ausgeschlossen werden. Bereits oberhalb von vier Metern können keine zuverlässigen Werte mehr ermittelt werden, da die Fehler der Rektifizierung auch hier in einem entsprechenden Messfehler resultieren.

### 6.2.2 Aufnahme und Auswertung mit dem kommerziellen Sensorsystem "Bumblebee 2" zur Stereovermessung im Nahbereich

Zur Wertung des technischen Standes soll ein Vergleich mit den Algorithmen des kommerziellen Produktes Bumblebee 2 erfolgen. Bei der Verarbeitung der Stereobildpaare direkt mit der zugehörigen Auswertung werden die Originalbilder in ihrer Größe reduziert (z.B. auf die halbe Originalauflösung), danach rektifiziert und dann die Tiefenkarte berechnet.

Die resultierende Tiefenkarte hat die Auflösung des Bildpaares mit der reduzierten Bildauflösung. Die Bilder mit der hohen Bildauflösung werden vorwiegend zur Überlagerung einer 3D-Darstellung der Tiefenkarte mit der Originaltextur verwendet.

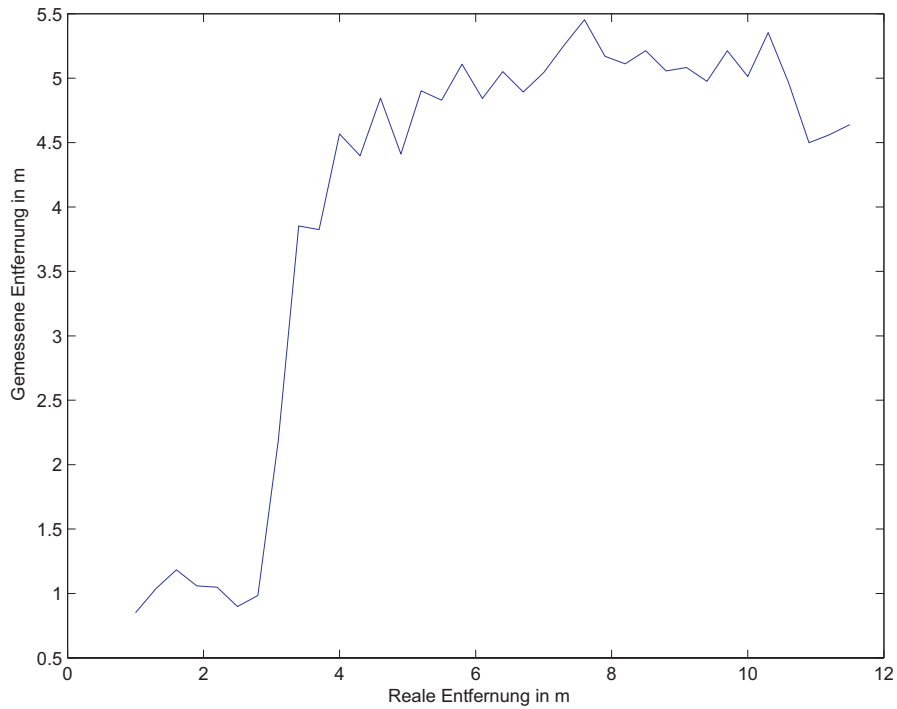
Bei der Auswertung aller auf dem Objekt liegenden 3D-Punkte erkennt man einen sehr ungünstigen Kurvenverlauf (siehe Abbildung 6.13(a)). Der Mittelwert der Entfernung stimmt am Anfang mit dem Sollwert überein, bleibt dann zunächst auf der initialen Entfernung und folgt erst ab 2 m wieder der Sollentfernung. Der Anstieg ist jedoch wesentlich zu steil. Bei einer Entfernung von 4 m erzeugt die Messung einen Wert von 4,9 m. Dieses Verhalten resultiert aus dem Ansatz, möglichst dichte Tiefenkarten zu generieren. Eine große einfarbige Fläche ist diesbezüglich jedoch ein sehr ungünstiger Fall. In dieser speziellen Messsituation führen diese geschätzten 3D-Punkte, mit denen die gleichfarbige Fläche aufgefüllt wird, zu einem erheblichen Fehler.

Um einen realen Vergleich der Qualität der Messwerte zu erhalten, wurden alle Punkte, die in Regionen mit geringer Textur liegen, von der Auswertung ausgeschlossen. Dies wurde in ähnlicher Weise durch die Berechnung der Varianz des Referenzblockes auch bei den im Rahmen dieser Arbeit entwickelten Verfahren realisiert.

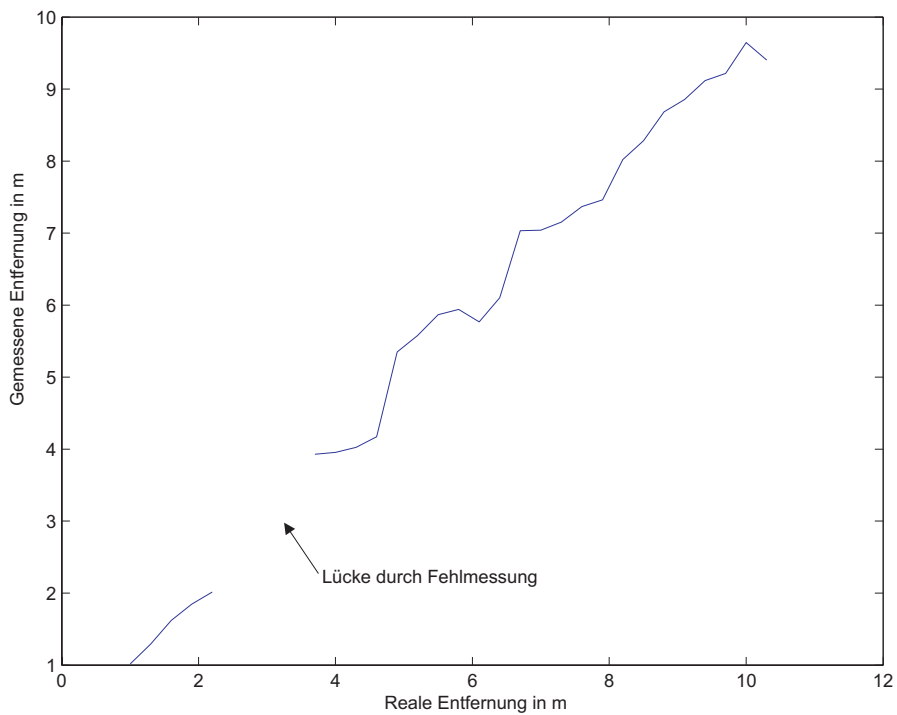
Nach der Reduktion der Messwerte auf die wesentlichen 3D-Punkte (siehe Abbildung 6.13(b)) folgt die gemessene Entfernung dem Sollwert verhältnismäßig gut bis zu einer Entfernung von ca. 10 m. Danach sinkt der Wert wieder ab.

Mit den gleichen Bildern wurden mit dem einfachen linearen blockbasierten Verfahren bis zu einer Entfernung von 7 m sehr zuverlässige Werte ermittelt und der Verlauf der Messkurve zeigt nicht diese starken Fehler (siehe Abbildung 6.12(b)). Daher müssen die Gründe dafür in den Algorithmen des Bumblebee 2 liegen. Der relative Fehler liegt entsprechend dem realen Fehler im Bereich bis 10 %.

Die wesentlichen Einsatzszenarien des Bumblebee 2 Kamerasystems liegen nicht in der Umfelderkennung, sondern im stationären Einsatz an einem PC zur Vermessung von texturierten Oberflächen. Es wird zum Beispiel in verschiedenen Projekten bei der Vermessung von Gesichtern und der Gestikererkennung erfolgreich am IESK eingesetzt. Die Messentfernungen sind in



(a) *Mittelwert aller Punkte*



(b) *Mittelwert der Punkte in relevanten Regionen*

Abbildung 6.13: Mittelwert der Entfernung bei der Auswertung mit den Algorithmen des Bumblebee2-Systems: Messbereich 1 - 10 m

diesen Fällen selten größer als 2 m.

### 6.3 Diskussion der Eigenschaften der in dieser Arbeit entwickelten Verfahren zur Stereobildauswertung

Die Messungen in diesem Kapitel haben verdeutlicht, dass die Verwendung des hierarchischen Auswerteverfahrens günstig für Messbereiche ist, die in der Z-Richtung eine erheblich größere Ausdehnung haben als in den anderen Richtungen. Dabei wurde im Wesentlichen auf die für die Umfeld erfassung relevanten Eigenschaften eingegangen, da in dieser Arbeit die Echtzeitfähigkeit und der Ressourcenverbrauch im Vordergrund stehen. Aschwanden in [9] führt einen komplexen Vergleich und eine spezifische Bewertung für eine große Auswahl an Korrelationskriterien durch. Dies ist für die Auswahl eines Kriteriums für eine spezielle Anwendung sehr hilfreich.

In diesem Kapitel wurden nur Untersuchungen für die in dieser Arbeit relevanten Verfahren bezüglich ihrer Eigenschaften in Implementierungen mit speziellen Randbedingungen, wie Bitbreite und Festkommaarithmetik für Bildsequenzen, aufgenommen im Nah- und im Fernbereich durchgeführt.

Zum Bewertung des technischen Standes der in der Arbeit behandelten Verfahren und ihrer Implementierungen zur Stereobildauswertung wurde ein Vergleich mit einem kommerziellen Produkt durchgeführt. Bei der Messung im Nahbereich mit dem kommerziellen Sensorsystem Bumblebee 2 der Firma Point Grey Research wurden sowohl die Aufnahmebedingungen mit kurzen Brennweiten mit und ohne Rektifizierung als auch die Auswertung mit den im System enthaltenen Algorithmen untersucht. Das System erzeugt hierbei eine dichte Tiefenkarte, welche bei stark texturierten Szenen die Realität gut widerspiegelt. In Szenen mit großen einfarbigen Flächen, in denen keine guten Korrespondenzen ermittelt werden können, erzeugen diese Interpolationen stark fehlerhafte Werte, was zu Fehlmessungen führt. Ein Vergleich mit den Ergebnissen der Verfahren dieser Arbeit lässt darauf schließen, dass das Bumblebee 2 nicht vorwiegend für die Hinderniserkennung konzipiert wurde. Es wurde ausgewählt, da es mit verhältnismäßig geringem technischen Aufwand eine Stereobildaufnahme und Auswertung in Echtzeit realisiert.

In für die Umfeld erfassung typischen Situationen konnten mit Hilfe der in dieser Arbeit behandelten Verfahren stabilere und genauere Messwerte ermittelt werden. Sie sind daher für den Anwendungsfall der Umfeld erfassung bei Fahrzeugen gut geeignet. Der Messfehler liegt in einer vergleichbaren Größenordnung wie der von herkömmlichen Messverfahren. Der Vorteil des hierarchischen Verfahrens besteht hierbei nicht in einer Reduktion des Fehlers, sondern vorwiegend in einer starken Reduktion des Rechenaufwandes.

Insbesondere mit Hilfe des hierarchischen Verfahrens kann eine echtzeitfähige Verarbeitung bei signifikanten Einsparungen an Ressourcen erreicht werden. Wie in Kapitel 5 dargestellt wurde, benötigt es nur ca. ein Drittel der Ressourcen, die für das lineare blockbasierte Verfahren für eine Abdeckung des gleichen Messbereiches erforderlich sind. Mit dem hierarchischen Verfahren

kann die Stereobildauswertung in eingebetteten Systemen ressourceneffizient mit vergleichbarer Genauigkeit umgesetzt werden.

Eine Implementierung des linearen blockbasierten Verfahrens unter Verwendung der Korrelatorkaskade für den gleichen Messbereich erfordert erheblich mehr Ressourcen. Die Genauigkeit in Z-Richtung ist, wie die Messungen verdeutlichen, weitgehend identisch. Im Gegensatz zum hierarchischem Verfahren kann bei einer echtzeitfähigen Implementierung hiermit für den Nahbereich eine höhere Auflösung in X-Richtung erreicht werden. Dies ist insbesondere bei Anwendungen zur Hindernisdetektion relevant, wie in Kapitel 7 detaillierter ausgeführt wird.

Der Vergleich der unterschiedlichen Korrelationskriterien in beiden Verfahren verdeutlicht, dass für ein weites Anwendungsfeld mit der QNKKFMF und der SADMF sehr ähnliche Ergebnisse erzielt werden können.





# Kapitel 7

## Umsetzung einer Hindernisdetektion für fahrerlose Transportfahrzeuge

Dieses Kapitel behandelt die Umsetzung einer Hindernisdetektion für fahrerlose Transportfahrzeuge als Anwendung für die Umfelderkennung auf Basis der Stereophotogrammetrie. Es soll die Erhöhung der Sicherheit und der Geschwindigkeit der fahrerlosen Fahrzeuge im Außenbereich erreicht werden.

Ziel ist die Detektion aller erhabenen Objekte (insbesondere Personen) im Fahrkorridor des fahrerlosen Fahrzeugs. Je nach Entfernung des Hindernisses soll das Fahrzeug langsam abgebremst oder sofort gestoppt werden. Für diese Aufgaben wurde das lineare blockbasierte Korrelationsverfahren ausgewählt und gemäß Abschnitt 5.2 mit Hilfe einer Korrelatorkaskade von SADM-Funktionen realisiert, da dieses Verfahren im Nahbereich (1 - 20 m) eine höhere Auflösung in X-Richtung als das hierarchische Verfahren liefert (siehe Abschnitt 6.2). Ebenso relevant für die Auswahl war der geringere Ressourcenaufwand gegenüber dem hierarchischen Verfahren mit der QNKKFMF. Als vollständige Implementierungen für die Realisierung eines Sensorsystems standen, wie in Kapitel 5 beschrieben, nur die Korrelatorkaskade mit der SADM und das hierarchische Verfahren mit der QNKKFMF zur Auswahl.

Die reduzierte Auflösung des hierarchischen Verfahrens ist für Sensorsysteme mit einem großen Messbereich in Z-Richtung optimiert. Diese Systeme sind in der Regel für Objektive mit einem kleinen Öffnungswinkel ausgelegt, wodurch eine hohe Auflösung in X-Richtung nicht erforderlich ist. Durch eine höhere Auflösung entsteht ein Mehraufwand.

Die Daten der Tiefenkarte werden mit Hilfe eines einfachen statistischen Verfahrens geclustert. Während der Clusterung werden die Position, die Ausdehnung und auch der Bewegungsvektor relevanter Objekte ermittelt. Aufgrund der Untersuchungen in Abschnitt 4.3 wurde eine FPGA-basierte Plattform ausgewählt. Die Bildaufnahme, Datenverarbeitung und Kommunikation mit dem Fahrzeug wurde vollständig mittels dieser Plattform realisiert.

Im diesem Kapitel werden zunächst die Algorithmen zur Clusterung beschrieben und anschließend die Komponenten und Funktionen der Hardwareplattform erläutert.

## 7.1 Anforderungen an das System zur Hindernisdetektion

Im Abschnitt 4.2.2 wurden die Randbedingungen auf Basis der Anforderungen eines Projektes "Fahrerloses Fahrzeug mit maschinellem Sehen" mit der Götting KG definiert. Die Fahrzeuge bewegen sich gegenwärtig mit 3 - 5 km/h und sollen zukünftig mit kamerabasierten Sicherheitssystemen Geschwindigkeiten von 10 - 15 km/h erreichen können. Der Messbereich wurde aufgrund der Geschwindigkeit auf einen Bereich von 1 - 20 m festgelegt und die zeitliche erforderliche Abtastung mit 10 Bilder/s ermittelt.

Zur Bildaufnahme wird ein Stereokamerasystem im Normalfall der Stereophotogrammetrie mit folgenden Parametern eingesetzt:

### Kamerasystem

Basisbreite	:	5,5 cm
Ausrichtung	:	Normalfall

### Kameras

Fabrikat	:	Photonfocus MV-D1024-28-CL-10
Sensorauflösung	:	1024x1024 Pixel
Grauwertaauflösung	:	10 Bit
Pixeltakt	:	28 MHz

### Objektive

Brennweite	:	12,5 mm
Öffnungswinkel	:	92,52°

### Tiefenkarte

Auflösung (x,y, $\Delta u$ )	:	64x1024x128
relativer Fehler Z	:	5 %

**Warnzonen** Bei der Detektion sollten die Hindernisse in 3 Warnzonen (0, 1, 2) eingeordnet werden. Die Warnzone 0 liegt direkt vor dem Fahrzeug. Ein Hindernis in dieser Warnzone 0 löst einen Notstopp aus. Die Warnzonen 1 und 2 schließen sich gestaffelt an und erfordern beim Fahrzeug eine an die Entfernung angepasste allmähliche Bremsreaktion. Die Warnzonen sind so angelegt, dass immer alle im Sichtbereich befindlichen Objekte berücksichtigt werden. Dadurch wird eine höhere Sicherheit und Stabilität beim Wechsel eines Objektes in eine kritischere Warnzone erreicht. Da keine Navigation mit den Daten der Stereobildanalyse realisiert werden soll, ist ein relativer Fehler von 5% bei der Messung ausreichend.

**Fahrkorridor** Zur Vermeidung von Fehlalarmen aufgrund von Objekten neben der eigentlichen Strecke des FTF<sup>1</sup> wird ein Fahrkorridor definiert, der dynamisch vom Fahrzeug an den Sensor übermittelt wird. Er beschreibt die zukünftige Fahrstrecke in den nächsten Sekunden anhand von 8 Punkten. Die 8 Punkte geben je 4 Punkte direkt links und rechts neben dem

---

<sup>1</sup>fahrerloses Transportfahrzeug

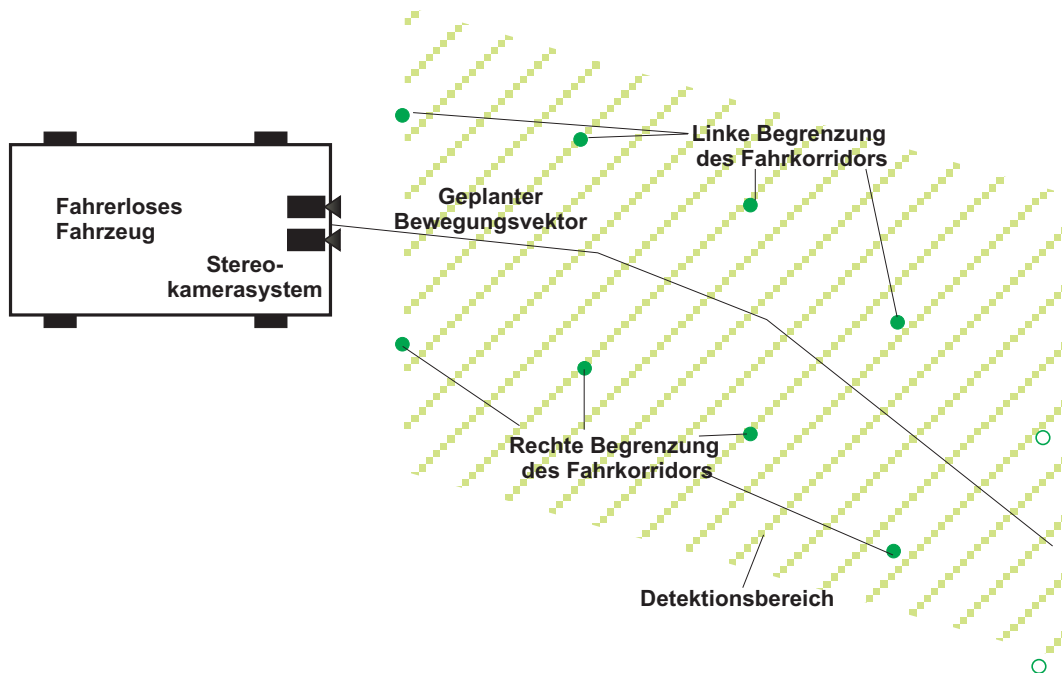


Abbildung 7.1: Anpassung des Detektionsbereiches an den Fahrkorridor

zukünftigen Spurverlauf des Fahrzeuges an. Diese Daten werden regelmäßig per CAN<sup>2</sup> an das Kamerasystem übertragen.

## 7.2 Hindernisdetektion mit Hilfe eines statistischen Verfahrens zur Auswertung der Tiefenkarte

Als Basis für die Hindernisdetektion dient eine Tiefenkarte (siehe Kapitel 2). Mit einer vorhandenen Tiefenkarte kann die Informationsmenge gegenüber einem Bild bereits deutlich gesenkt werden. Für eine Hinderniserkennung oder eine Navigation sind die Zusammenhänge der Daten nicht eindeutig genug. Daher sollen durch eine Clustering die Zusammenhänge der 3D-Punkte aufgezeigt und so die Daten weiter auf das Wesentliche reduziert werden.

Im Folgenden wird nach einer Diskussion von statistischen und objektbezogenen Clusterverfahren ein statistisches Verfahren für die Auswertung der Stereobilder verwendet.

<sup>2</sup>Controller Area Network - Feldbus, der vorwiegend in Fahrzeugen zur Reduzierung der Kabelbäume eingesetzt wird

### 7.2.1 Statistische Clusterverfahren im Vergleich zu anderen Clusterverfahren

Für die Rekonstruktion von Objekten aus einer Tiefenkarte gibt es viele unterschiedliche Verfahren. Je nach dem verwendeten Verfahren sind a priori Informationen über das oder die zu erkennenden Objekte erforderlich.

Verfahren, die die konkrete Form zur Clusterung und Identifikation eines Objektes benötigen, sind nur für eine begrenzte Anzahl von Objekten realisierbar. Diese Verfahren arbeiten nur im Nahbereich [44], da sie sehr dichte Tiefenkarten benötigen. Sie ermöglichen die Klassifizierung [77] der Objekte. Für die Umfelderkennung ist eine Klassifizierung in den meisten Fällen nicht notwendig. Daher sind formorientierte Clusterverfahren für diese Problemstellung nur bedingt erforderlich.

Wird die Form eines Objektes abstrahiert, kann eine passende Box um ein Objekt gelegt werden. Dabei wird zum Beispiel durch ein Optimierungsverfahren eine zu den Objektklassen passende Box so um Häufungen von 3D-Punkten angeordnet, dass der Abstand zu allen Punkten bzw. zum Schwerpunkt der Wolke minimal ist [43]. Hierbei müssen je nach Implementierung unter Umständen alle Punkte mehrfach überprüft werden. Prinzipiell kann dieses Verfahren auch ohne Kenntnis der Boxgröße angewendet werden. Dieses Vorgehen führt jedoch zu einer deutlich höheren Anzahl an Iterationen. Aufgrund der mehrfachen Überprüfung jedes Punktes ist das Verfahren nur bedingt für eine Echtzeimplementierung mit begrenzter Rechenleistung geeignet. Aus diesem Grund wurde für die Hinderniserkennung in diesem Projekt ein statistisches Merkmal zur Clusterung ausgewählt [81]. Hierbei wird die Anzahl der Punkte in ein räumlich verteiltes Histogramm eingetragen und anschließend durch Schwellwerte die Existenz von erhabenen Objekten bestimmt. Für eine Verarbeitung in einem eingebetteten System können statistische Verfahren so realisiert werden, dass sie in mehreren Stufen zunächst erhabene Objekte und anschließend bewegte Objekte ermitteln [39]. Im Folgenden wird das im Rahmen dieser Arbeit verwendete Verfahren beschrieben.

Ein ähnlicher Ansatz wird von Nguyen [67] in Form einer gridbasierten Datenfusion verfolgt.

### 7.2.2 Beschreibung des räumlichen statistischen Clusterverfahrens

Für die Auswertung der Tiefenkarte in Echtzeit wurde in diesem Fall ein statistisches Auswerteverfahren ausgewählt und erweitert. Dazu wird der Raum in Zellen eingeteilt und ein räumliches Histogramm der 3D-Punkte für jedes Stereobildpaar erstellt (Abbildung 7.2). Die Verteilung der Punkte im Raum beschreibt die Verteilung der Wahrscheinlichkeit für die Existenz eines Hindernisses. Die Belegung der Zellen mit Objekten wird mittels eines Schwellwertes im Histogramm ermittelt. Diese Belegungskarte wird dann zur Clusterung herangezogen. Anschließend erfolgt eine Verifizierung und Verfolgung der Objekte über mehrere Bilder.

Arbeitsschritte des Verfahrens:

- Bildung eines räumlichen Histogramms über der Tiefenkarte
- Detektion von Zellen, die Teil eines erhabenen Objektes sind, und Markierung in einer Belegungskarte
- Clusterung der Zellen von erhabenen Objekten aufgrund ihrer Nachbarschaftsbeziehungen in der Belegungskarte
- Verifizierung der Objekte über die vorangegangenen Zustände in der History der Belegungskarte

Nachfolgend werden die einzelnen Schritte erläutert

### 7.2.2.1 Erzeugung eines räumlichen Histogramms aus der Tiefenkarte

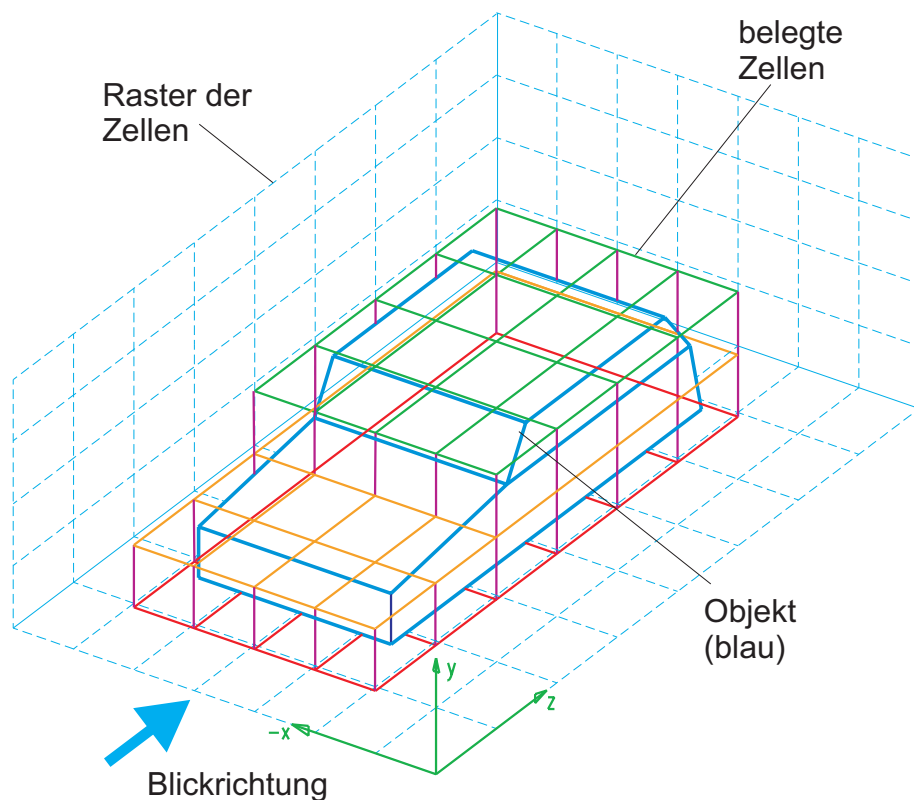


Abbildung 7.2: Clusterung im räumlichen Histogramm

Für dieses Projekt wurde das Raster an der Disparität orientiert. Die Größe der Zellen nimmt mit der Entfernung zu, da bei der linearen blockbasierten Korrelation mit voller Auflösung in der Nähe deutlich mehr Punkte verfügbar sind als an der oberen Grenze des Messbereiches.

Es wurde ein Raster mit der einfachen Blockbreite ( $\hat{=}16$  Pixel) in x-Richtung und einer Schrittweite von vier Pixeln bei der Disparität gewählt. In Simulationen wurde die Gesamtmesshöhe in vier Zellen unterteilt, um so Objekte auf dem Boden, Objekte in mittlerer Höhe und Objekte über dem Fahrzeug unterscheiden zu können. Dadurch können Objekte oberhalb der Durchfahrthöhe ausgeschlossen werden. Ein Beispiel ist in Abbildung 7.2 dargestellt. Dabei wird ein Objekt (blau) vollständig von belegten Histogrammzellen eingeschlossen.

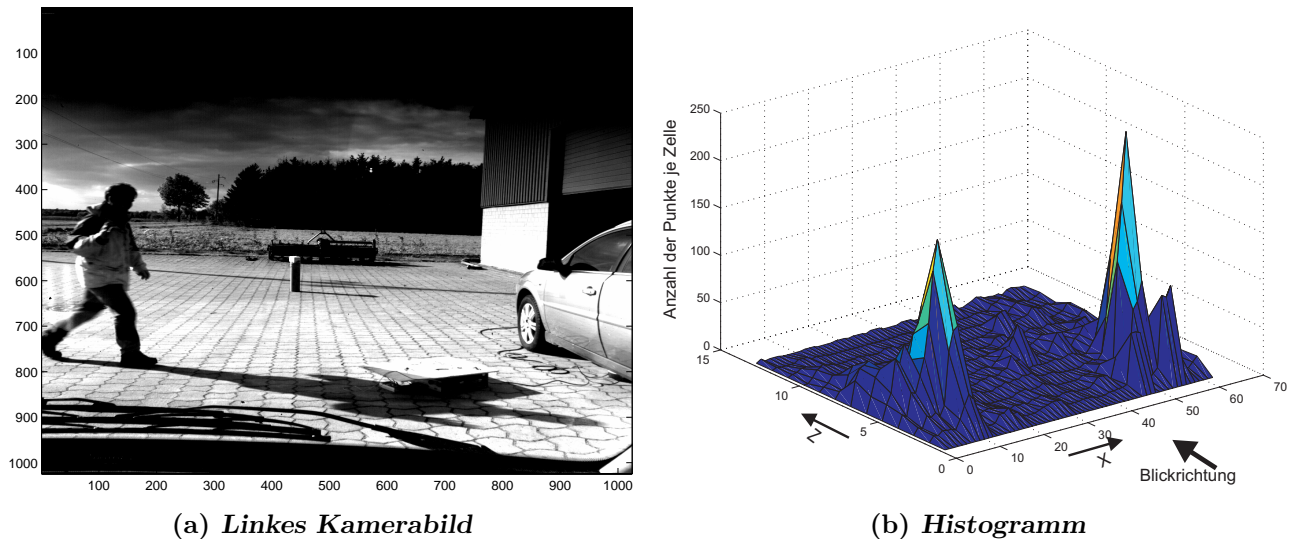


Abbildung 7.3: Reduktion auf ein 2D-Histogramm zur Clusterung

Aus Implementierungsgründen erstrecken sich die Zellen der Implementierung für das eingebettete System zum gegenwärtigen Zeitpunkt über die gesamte Höhe des Messbereiches. Dadurch entsteht ein zweidimensionales Histogramm mit  $64 \times 32$  Zellen (siehe Abbildung 7.5 (links)). Es ergeben sich 2048 Zellen für das 2D-Histogramm. Alle 3D-Punkte der Tiefenkarte können räumlich jeweils einer Zelle zugeordnet werden. So ergibt sich das räumliche Histogramm. Für jede Zelle wird zusätzlich der Mittelwert der Disparität aller ihr hinzugefügten 3D-Punkte gebildet.

Der Ablauf der gesamten Hindernisdetektion ist in Abbildung 7.4 dargestellt. Nach der Erstellung der Tiefenkarte erfolgt das Tiefenhistogramm. Im nächsten Abschnitt wird die Clusterung anhand des räumlichen Histogramms beschrieben.

### 7.2.2.2 Objektdetektion, Weiterverfolgung und Prädiktion der Bewegungsrichtung von Clustern

Zur Clusterung von Objekten muss zunächst festgestellt werden, welche Zellen des Histogramms ein erhabenes Objekt repräsentieren. Hierfür wird ein geeigneter Schwellwert ermittelt, der sich relativ zu den Parametern für die Erstellung der Tiefenkarte verändert.

Durch Bewegungen kann eine zwischenzeitliche Verteilung der Punkte eines Objektes auf mehrere Zellen des Histogramms erfolgen. Dadurch kann eine feste Schwelle unterschritten werden.

Die Objektdetektion muss entweder durch eine Modifikation des Histogramms oder eine übergreifende Betrachtung von mehreren Zellen stabilisiert werden. Bei der Modifikation des Histogramms werden doppelt so viele Zellen erzeugt, die sich jeweils zu 50% überlappen, und alle Punkte werden zweimal ausgewertet. Dadurch befinden sich bei der Bewegung eines erhabenen Objektes im Bild ausreichend Punkte in einer Zelle, um das Objekt sicher zu detektieren. Dieser Schritt erzeugt eine scharfe Belegungskarte, aber auch den doppelten Speicherbedarf für das Histogramm für jedes Bild.

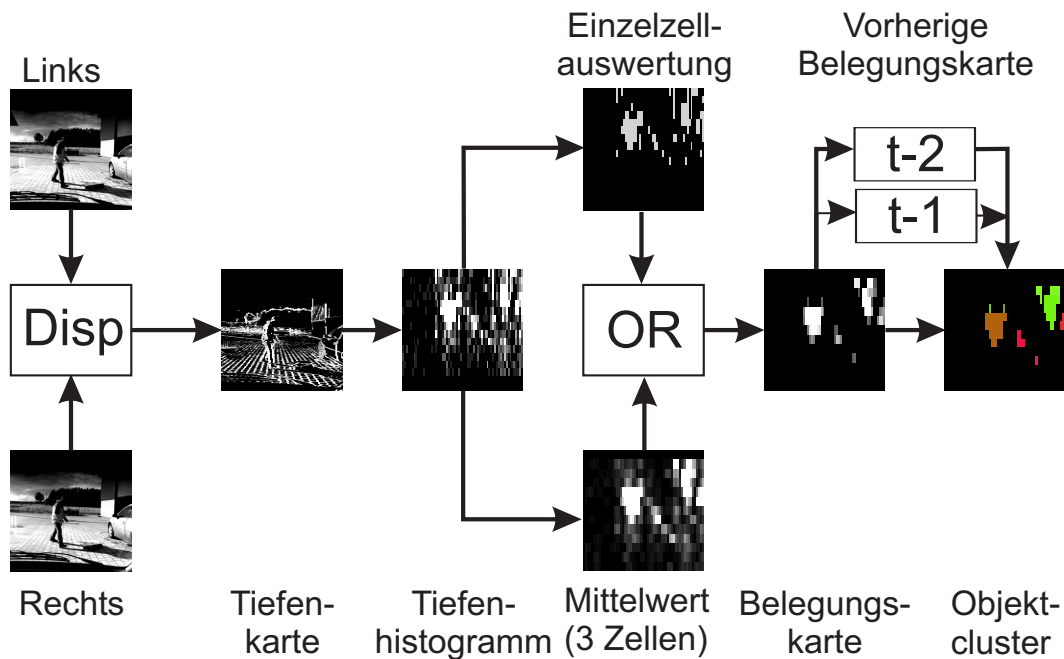


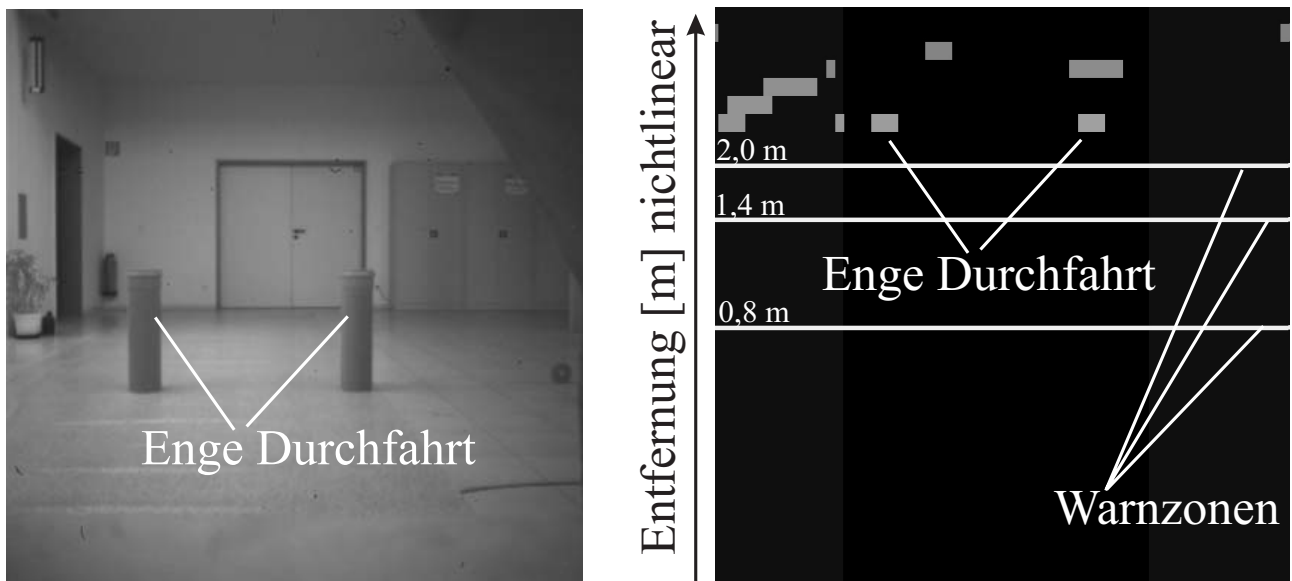
Abbildung 7.4: Ablauf des Clusterverfahrens

Die zweite Variante beruht auf einer Bildung des Mittelwertes über 3 Zellen des Histogramms in x-Richtung. Eine Mittelwertbildung erfordert eine Division. Um diese Division zu vermeiden, erfolgt die Auswertung über einen separaten Schwellwert. Das Tiefenhistogramm wird mit Hilfe der Einzelzellauswertung und der Mittelung über 3 Zellen parallel ausgewertet. Ist eine der beiden Schwellen überschritten, ist ein erhabenes Objekt vorhanden und die Zelle wird in der Belegungskarte markiert, dies ist durch das "OR" in Abbildung 7.4 dargestellt.

Zur Clusterung werden die markierten Zellen auf Nachbarschaft überprüft. Beginnend mit der linken hinteren (beim räumlichen Histogramm der unteren) Zelle in der Nachbarschaft überprüft ob weitere belegte Zellen existieren. Dadurch werden zusammenhängende Areale ermittelt. Dies wird dann durch einen zweiten Lauf von vorn rechts (räumlich: oben) wiederholt, um mögliche separierte Verästelungen einzubinden. Die Zellen, die zusammenhängende Areale bilden, sind die Cluster. Jeder Cluster bekommt zur Verdeutlichung eine eigene Farbe zugewiesen.

Anschließend wird die Existenz der Cluster den Ergebnissen der Clusterung der zwei vorangegangenen Bilder gegenübergestellt, um die Existenz eines Objektes zu verifizieren und die entsprechende Clusternummer zu übernehmen. Dies wird in Abbildung 7.4 durch das "t-1" und das "t-2" dargestellt. Für jede zur Verifizierung abgeglichene Belegungskarte der Vergangen-





(a) *Linkes Kamerabild*

(b) *Ausgabe: Clusterung*

Abbildung 7.5: Ausgabe der Ergebnisse der Clusterung durch das Messsystem

heit wird der Suchraum um eine Zelle erweitert, sodass auch Bewegungen des Objektes im Bild berücksichtigt werden. Die Betrachtung der zwei vorangegangenen Belegungskarten ist im Allgemeinen ausreichend. Cluster ohne entsprechende Abbildungen in der Vergangenheit sind neue Cluster und erhalten eine neue Clusternummer. Für jeden Cluster wird aus der aktuellen Repräsentation die Position des Schwerpunktes, die Ausdehnung und zusammen mit den vorhergehenden Belegungskarten die Bewegung des Schwerpunktes ermittelt.

Auf der Basis der Bewegungen der Schwerpunkte kann mit Hilfe von Prädiktionsfiltern die Bewegung prädiziert werden. Eine Prädiktion ist jedoch insbesondere bei Lebewesen sehr problematisch, da ihr Bewegungsvektor sich im Gegensatz zu Fahrzeugen sehr schnell ändern kann. Aus diesem Grund ist eine Prädiktion für den vorgesehenen Einsatzfall gegenwärtig nicht sinnvoll.

Die Abbildung 7.5 zeigt die Ausgabe der Ergebnisse für die Clusterung während einer aktiven Messung durch das echtzeitfähige Messsystem. Links ist das Bild der linken Kamera dargestellt. Rechts hingegen sind die geclusterten Objekte in Graustufen dargestellt. Die drei waagerechten Striche zeigen die Positionen der Warnzonen.

### 7.3 Hardwareplattform für die Implementierung der Hindernisdetektion

Für ein echtzeitfähiges Sensorsystem zur Stereobildverarbeitung ist eine leistungsfähige Hardwareplattform, ein Kamerasystem zur Bildaufnahme ebenso wie die Kommunikation mit dem



Fahrzeug erforderlich. Im folgenden Abschnitt werden die wesentlichen Parameter des Entwicklungssystems beschrieben.

### 7.3.1 Eingliederung in das Gesamtsystem

Die Implementierung des Demonstrators erfolgt auf einer bereits bestehenden Hardwareplattform des IESK<sup>3</sup> (Abbildung 7.6). Diese Plattform besteht aus zwei FPGAs [6] EP1S60 von Altera und verfügt über zusätzlichen Speicher und vier CameraLink-Interfaces (zwei Eingänge und zwei Ausgänge). Diese FPGA-basierte Plattform und Teile des FPGA-Designs wurden im Rahmen von verschiedenen Projekten im IESK aufgebaut und erweitert. Die vorhandenen Teile wurden zur Realisierung der Hindernisdetektion für fahrerlose Transportfahrzeuge angepasst.

Von der Hardwareplattform werden Signale zur Ansteuerung der Kameras gesendet. Weiterhin nimmt sie die Kameradaten in Echtzeit auf und verarbeitet diese gleichzeitig. Die Ergebnisse werden anschließend an das FTF mittels eines CAN-Interfaces übermittelt.

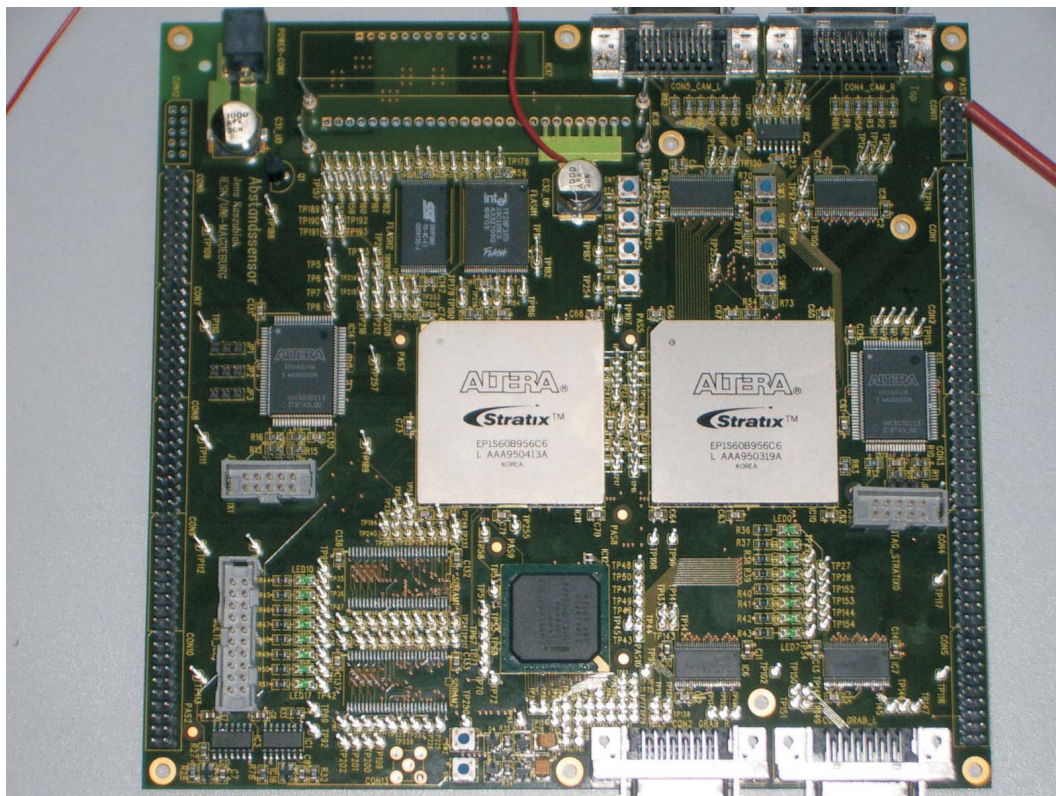


Abbildung 7.6: Hardwareplattform

<sup>3</sup>Institut für Elektronik, Signalverarbeitung und Kommunikationstechnik der Otto-von-Guericke Universität Magdeburg

### 7.3.2 Hardware-Software-Co-Design zur Realisierung einer Echtzeitverarbeitung

Um die Hardwarevoraussetzungen für die Implementierung möglichst gering zu halten, ist eine Aufteilung des Algorithmus auf Hardware und Software vorgesehen, damit sowohl die datenflussorientierten als auch die kontrollflussorientierten Teile des Algorithmus gemäß der Echtzeitbedingung effektiv implementiert werden können.

Zur Datenverarbeitung sind auf dem Demonstrator zwei FPGAs vom Typ EP1S60 der Firma Altera verbaut. Sie haben jeweils ca. 60 000 Logikzellen und ca. 800 frei programmierbare Pins.

Altera hat für seine FPGAs [5] speziell den NIOS-Prozessor entwickelt, wodurch sich ein geringer Ressourcenbedarf ergibt. Der NIOS-Prozessor ist eine 32-Bit-RISC-Maschine, welche sowohl durch einen mathematischen Co-Prozessor, als auch durch im FPGA codierbare und somit extrem effektive Spezialoperationen erweitert werden kann.

Somit kann in einem FPGA ein Hardware-Software-Co-Design effektiv implementiert werden.

#### 7.3.2.1 Umsetzung der Algorithmen in ein Hardware-Software-Co-Design

Die Algorithmen für den Hindernissensor sind sehr unterschiedlich. Während der Algorithmus zur Berechnung einer Tiefenkarte aus einem Stereobild prinzipiell ein datenflussorientierter Algorithmus ist, stellt insbesondere die Kommunikation mit dem übergeordneten System eher kontrollflussorientierte Algorithmen dar.

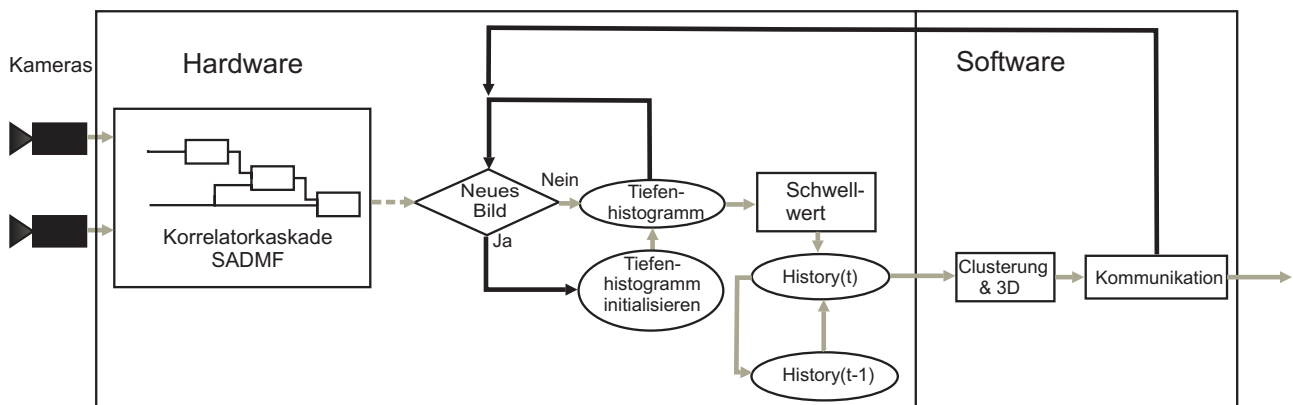


Abbildung 7.7: Aufteilung des Algorithmus auf Hard- und Software

Die Algorithmen zum Aufbau der Belegungskarte sind gleichermaßen datenflussorientiert und kontrollflussorientiert. Daraus resultiert, dass die Art der Implementierung in diesem Fall im Wesentlichen von der geforderten Geschwindigkeit bzw. den zur Verfügung stehenden Hardwareressourcen abhängt.

Für die Kommunikation mit dem Fahrzeug wurde das CAN-Protokoll und zur Kommunikation mit dem Konfigurationsrechner das USB-Protokoll ausgewählt. Beide Protokolle lassen sich besser in Software abbilden als in Hardware.

Die Algorithmen zur Stereobildverarbeitung und zur Clusterung können weitestgehend datenflussorientiert in Hardware implementiert werden. Die Kommunikation mit dem FTF erfolgt aufgrund der Auswertung des Kommunikationsprotokolls des CAN-Busses vorwiegend kontrollflussorientiert.

Die Partitionierung der Algorithmen in Hard- und Softwareimplementierung ist in Abbildung 7.7 dargestellt.

### 7.3.3 Bildaufnahme und Bildausgabe mittels Kameralink

Für eine Auswertung der Bilddaten in Echtzeit müssen die Bilddaten in Echtzeit aufgenommen und zeitgleich verarbeitet werden. Im FPGA steht nicht genügend Speicher zur Verfügung, um ein komplettes Bild zu speichern. Jedes der beiden Bilder erfordert etwa 2 MB Speicherplatz, der unabhängig von der Berechnung benötigt wird. Weiterhin ist ein zentralisierter Speicher für eine räumlich verteilte und parallelisierte Berechnung ungünstig. Aus diesem Grund muss das System die Bilder synchron zur Bildaufnahme ohne Zwischenspeicherung verarbeiten.

Die Verarbeitung der Bilder online zur Aufnahme hat deutliche Geschwindigkeitsvorteile. Dies erfordert aber, dass die beiden Kameras und die Auswertehardware ein zeitlich synchrones System bilden müssen. Somit ist eine Vorverarbeitung der Bilder nur möglich, wenn auch sie online zur Bildaufnahme realisiert werden kann. Eine Rektifizierung ist in diesem Fall nur sehr aufwändig realisierbar. Somit können hier nur Kameras verwendet werden, deren Objektive die Bilder kaum verzerren.

Weiterhin müssen zur Realisierung dieser Funktionalität einige Randbedingungen von der Platine und den Kameras erfüllt werden.

#### 7.3.3.1 Eigenschaften der Kameras

**Externer Takteingang** Üblicherweise laufen Kameras mit ihrem internen Takt. Nur ein Teil der auf dem Markt verfügbaren Kameras kann durch einen Fremdtakt angesteuert werden. Mit diesem Takt läuft dann die gesamte innere Elektronik der Kameras von der Aufnahme der Bilder bis hin zur Übertragung an das FPGA.

**Externer Synchronisationseingang und globaler Shutter** Nur Kameras, bei denen über einen externen Synchronisationseingang der Zeitpunkt der Bildaufnahme bestimmt werden kann, sind für stereophotogrammetrische Aufnahmen von sich verändernden Szenen verwendbar. Problematisch hierbei ist, dass auch die Bildaufnahme vollständig synchron erfolgen muss. Dies kann nur durch einen globalen Shutter realisiert werden. Zu Beginn des Projektes war

dies insbesondere bei CMOS-Kameras vorwiegend nicht der Fall [15]. Mittlerweile verfügt die Mehrzahl der CMOS-Kameras über einen globalen Shutter, bei dem die Bildaufnahme mit einer elektronischen Blende gesteuert wird. Dadurch empfangen alle CMOS-Elemente im Sensor zur gleichen Zeit die Information. Bei CCD-Kameras ist dies vom technischen Prinzip her erforderlich, da Ladungen in den Charged Coupled Devices aufgenommen werden und anschließend bei “geschlossenem” Shutter seriell durch Ausschleichen ausgelesen werden. Bei CMOS-Kameras kann theoretisch jedes Element selbstständig eine “Bildaufnahme” realisieren. Dadurch kann die Integrationszeit als Verzögerungsfaktor minimiert werden. Diese Aufnahmetechnik ist jedoch für Stereobildaufnahmen von bewegten Szenen nicht verwendbar. Es muss zwingend bei beiden Kameras zur gleichen Zeit die Belichtung erfolgen.

**Freie Definition der Bild- und Zeilenpausen** Um prinzipielle Laufzeit- und Reaktionsunterschiede der beiden Kameras auszugleichen, müssen ggf. die Bild- und Zeilen-Pausenzeiten angepasst werden. Es hat sich gezeigt, dass auch bei Kameras eines Modells, die nicht aus der gleichen Serie stammen, zeitlich unterschiedliches Verhalten bezüglich ihrer Bildaufnahme vorhanden ist. Dies kann durch Variation der Pause zu Beginn der Aufnahme ausgeglichen werden. Prinzipiell kann dieser Unterschied auch auf der Platine ausgeglichen werden, da er immer konstant bleibt.

### 7.3.3.2 Eigenschaften der Platine

**Parallele Aufnahme zweier Kamerastreams** Zur zeitgleichen Verarbeitung von Stereokamerabildern müssen diese parallel zueinander aufgenommen werden. Somit muss die gesamte Verarbeitungstrecke zur Bildaufnahme vollständig parallel ausgeführt sein.

**Senden eines spezifizierbaren Kamerataktes** Kameras arbeiten jeweils nur in einem speziellen Taktbereich, auf den sie optimiert sind. Für die vollständig synchrone Verarbeitung ist es erforderlich, dass das FPGA die Kameras mit einem Takt speist. Da aktuelle FPGAs über leistungsfähige PLLs verfügen, kann aus einer über einen Quarz zur Verfügung gestellten Frequenz ein breites Spektrum an Frequenzen erzeugt werden. Hierbei ist zu beachten, dass die PLLs bis zu 5 verschiedene Takte aus einem Eingangstakt generieren können, solange sie alle aus einer höheren Frequenz generiert werden. Diese Frequenzen sind dann synchron zueinander.

**Synchrone Bildaufnahme mittels eines spezifischen Synchronisationssignals** Abhängig von den Kameras muss ein spezifisches zeitlich exaktes Synchronisationssignal generiert werden. Da aufgrund des identischen Taktes eine gleiche Zeitbasis zur Verfügung steht, kann dieses Signal anhand des Datenblattes der Kameras erstellt werden. Die Bildaufnahme sollte jedoch unabhängig davon auf die Steuersignale der Kameras reagieren, um Störungen aufgrund von Verzögerungen der Datenübertragung zu minimieren.

Die Übertragung des Synchronisationssignals und die Datenübertragung benötigen je nach Kabellänge unterschiedlich viel Zeit. Aus diesem Grund sollten mit Hilfe von Verzögerungselementen in den IO-Blöcken von FPGAs und einer anschließenden taktsynchronen Übernahme die Daten in das Zeitraster des FPGAs überführt werden.

**Ausgleich minimaler zeitlicher Verschiebungen der Bilder** Um technische Unterschiede verschiedener Exemplare eines Kameramodells auszugleichen, sind Ausgleichseinheiten erforderlich. Diese können durch einen Ringpuffer mit unterschiedlicher Laufzeit realisiert werden. Dadurch kann zur Laufzeit die zeitliche Verzögerung zwischen den beiden Kamerasignalen ausgeglichen werden.

### 7.3.3.3 Kommunikation mit dem übergeordneten System

Um dem Fahrzeug Hindernisse mitzuteilen, muss das Sensorsystem mit dem Fahrzeug kommunizieren. Da ein Großteil der Kommunikation des Fahrzeugs auf dem CAN-Bus basiert, wurde zu Beginn des Projektes entschieden, Hindernisinformationen auch über dieses Protokoll zu übertragen. Zur Anpassung und Konfiguration des Sensors wurde weiterhin eine Schnittstelle zu einem Notebook konzipiert, um den Sensor zur Laufzeit konfigurieren zu können. Die Konfiguration über CAN und somit über das Fahrzeugbussystem zu realisieren, wäre zu dem gegebenen Zeitpunkt sehr aufwändig gewesen. Die Verbindung zum Konfigurationsrechner wurde deshalb mittels USB hergestellt.

#### Kommunikation vom Fahrzeug zum Sensor

**Fahrkorridor** Damit der Sensor entscheiden kann, ob sich ein Hindernis im Fahrkorridor des Fahrzeugs befindet und somit die Gefahr einer Kollision besteht, muss dieser Korridor an den Sensor übertragen werden. Dazu wurde von der Götting KG ein Protokoll mit Hilfe von 8 Punkten vor dem Fahrzeug entwickelt. Die 8 Punkte geben je 4 Punkte direkt links und rechts neben dem zukünftigen Spurverlauf des Fahrzeugs an. Diese Daten werden regelmäßig über den CAN-Bus aktualisiert.

In [Abbildung 7.1](#) ist der Fahrkorridor und seine Auswirkung auf den Detektionsbereich qualitativ dargestellt.

**Geschwindigkeit des Fahrzeugs** Die Geschwindigkeit hat auf die Reaktion des Fahrzeugs, insbesondere auf das Stoppverhalten, entscheidende Auswirkungen. Dementsprechend müssen auch die Warnbereiche angepasst und relativ zur Fahrzeuggeschwindigkeit festgelegt werden. Würde dies der Sensor übernehmen, müsste bei Umsetzung für ein neues Fahrzeug das entsprechende Fahrverhalten im Sensor implementiert werden. Da dies nicht sinnvoll ist, bietet sich die doppelte Verwendung des Fahrkorridors an. Hierbei wird der Abstand zwischen den Punkten



des Fahrkorridors in Fahrtrichtung derart variiert, dass dadurch die Schwellen für die unterschiedlichen Warnbereiche definiert werden. Somit stellen sich die Warnbereiche als Flächen in Form von Vierecken vor dem Fahrzeug dar.

**Kommunikation vom Sensor zum Fahrzeug** Bei der Übertragung der Hindernisdaten stellen sich zwei prinzipielle Szenarien:

1. Reine Hindernisdetektion der Sensor meldet nur das Vorhandensein eines Hindernisses in den Warnzonen
2. Komplexe Umfelderkennung zur Planung von Ausweichrouten, der Sensor meldet die Existenz, die Position und Ausdehnung jedes Hindernisses im Sichtbereich

Im Projekt wurde zunächst der erste Fall realisiert, da dies der minimale Betriebsfall ist. Die Kommunikation für den zweiten Fall wurde aus zeitlichen Gründen nicht implementiert. Außerdem ist zum gegenwärtigen Zeitpunkt die Planung einer Ausweichroute aufgrund fester Fahrtrouten für automatisierte Fahrzeuge auf Betriebsgeländen nicht vorgesehen. Die Algorithmen wurden so gestaltet, dass die Information über die Position der Objekte ermittelt werden kann.

**Kommunikation mit dem Konfigurationsrechner** Zur Anpassung der Parameter des Sensorsystems wurde eine Kommunikation über USB mit einem Notebook realisiert. Auf der Sensorplatine werden diese Konfigurationsdaten mittels eines I<sup>2</sup>C-Busses übertragen. Mit Hilfe dieses Übertragungsweges werden die Parameter in einen Speicher geschrieben, aus dem sie dann zeitgleich und kontinuierlich im FPGA abgegriffen werden können. Um dies zu gewährleisten, besteht der Speicher aus unabhängigen Registern, welche über ein Interface mit Speicheradressen versehen wurden. Somit ist ein paralleler Zugriff auf alle Konfigurationsdaten, wie er für eine Hardwareimplementierung erforderlich ist, gewährleistet. Zur Ansteuerung wurde ein Windows-Programm mit einer übersichtlichen Oberfläche (Abbildung 7.8) entworfen, welches den direkten Zugriff sowohl auf die RAM-Zellen als auch auf spezielle Steuerelemente für häufig verwendete Funktionen zur Verfügung stellt.

## 7.4 Testfahrt mit Hindernisdetektion auf einem automatisierten Fahrzeug

Im Verlauf des Projektes<sup>4</sup> mit der Götting KG erfolgten mehrere Testfahrten. Sie dienten vornehmlich der Gewinnung von Daten zur Entwicklung der Algorithmen für die Hindernisdetektion. Diese wurden sowohl mit einem PKW "Smart" im Außenbereich und mit einem von der Götting KG zur Verfügung gestellten Elektrofahrzeug im Innenraum durchgeführt. Bei den

---

<sup>4</sup>AIF-Projekt: Fahrerloses Transportfahrzeug mit maschinellem Sehen

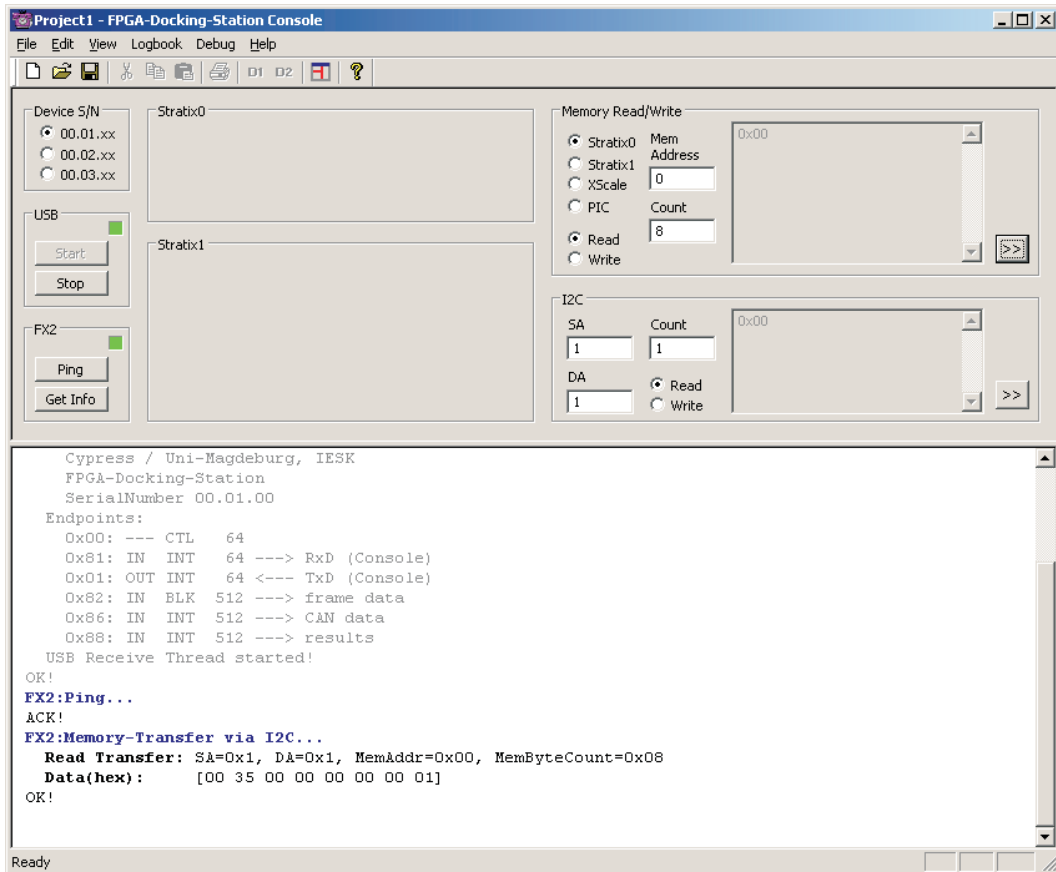


Abbildung 7.8: Oberfläche des Programmes "FPGA-Docking-Station" zur Anpassung von Parametern zur Laufzeit

Fahrten mit dem Smart wurden über eine CAN-Schnittstelle verschiedene Fahrzeugparameter, unter anderem Position, Bewegungsrichtung und Geschwindigkeit, mitprotokolliert. Die gewonnenen Daten wurden mit der Softwareemulation der hardwareorientierten Algorithmen verarbeitet und die Funktion anhand der gewonnenen Daten überprüft.

Abschließend erfolgten Testfahrten mit einem auf Basis der entwickelten Algorithmen erstellten eingebetteten echtzeitfähigen System. Diese Testfahrten wurden wieder auf dem Elektrofahrzeug bei der Götting KG durchgeführt. Durch die Götting KG wurde das Areal für die Testfahrt vermessen und so mögliche Hindernisse in Form von Schränken, Türen und technischen Anlagen auf dem Kurs im Sichtbereich des Fahrzeugs bestimmt. Die Auswirkungen der teilweise engen und umbauten Durchfahrten in Kurven zeigen sich in Abb. 7.9. Durch Objekte, wie Wände, Schränke und Maschinen, die sehr nahe am Fahrweg des Fahrzeugs liegen und so in den Warnzonen erscheinen, wird das Fahrzeug abgebremst und teilweise gestoppt, obwohl sie kein Hindernis auf dem Fahrweg darstellen.

In Abbildung 7.9 sind insbesondere im oberen Bereich des Kurses kurzzeitige Messfehler erkennbar. Diese resultieren aus zufälligen Häufungen von 3D-Punkten im Histogramm aufgrund von Schatten und 3D-Punkten durch falsche Korrespondenzen, da kein reales Objekt existiert, das diese Häufung erzeugt haben könnte. Sie treten jeweils nur für ein Bild auf. Während der

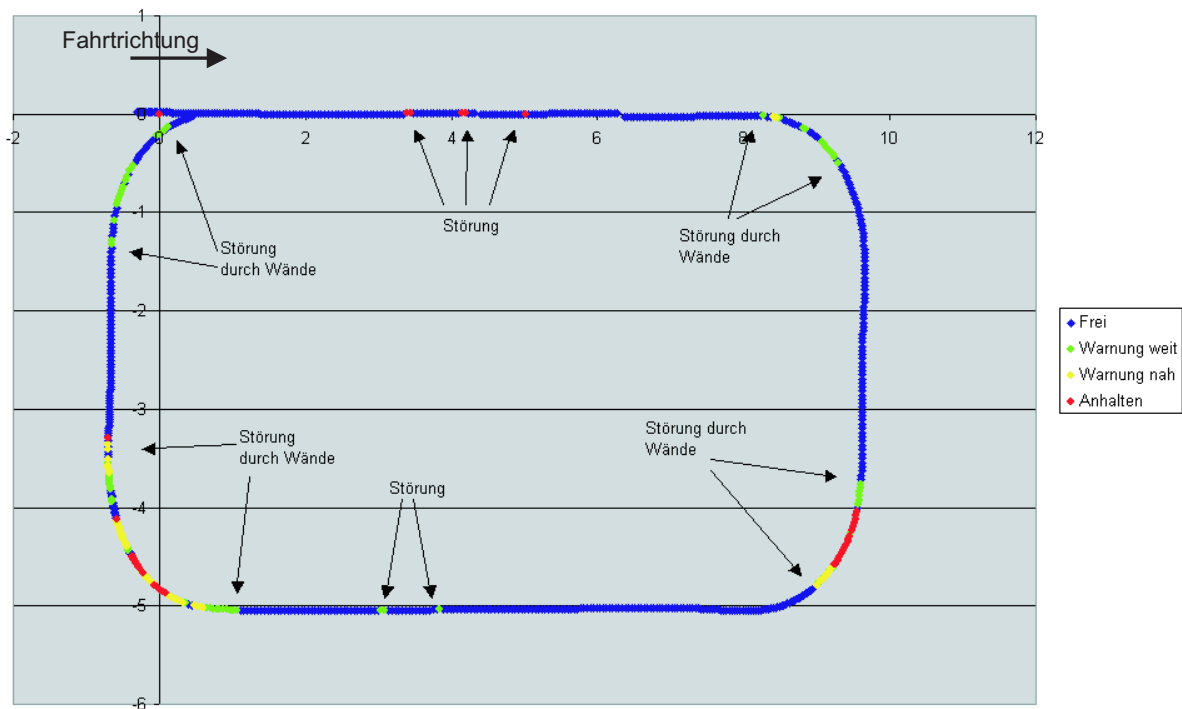


Abbildung 7.9: Testfahrt ohne Hindernis in der Fahrstrecke

Testfahrt wurde keine Verifizierung über die vorangegangenen Belegungskarten durchgeführt, da die Implementierung der Verifizierung in das eingebettete System zum Zeitpunkt der Testfahrt noch nicht abgeschlossen war. Durch die Verifizierung über die Vergangenheit erhöht sich die Sicherheit und die Stabilität der erfassten Hindernisse deutlich. Bei Detektion eines Objektes im Haltebereich muss definitiv eine Bremsung durchgeführt werden. Für einen sicheren Betrieb müssen die Warnbereiche dynamisch an das Fahrverhalten angepasst werden. Hierbei muss insbesondere der Bremsweg des Fahrzeugs berücksichtigt werden.

Das Diagramm in Abbildung 7.9 zeigt in den Kurvenfahrten Störungen und Haltesituationen. Diese resultieren aus engen Kurvendurchfahrten, sodass die Umgebung des Kurses als Hindernis detektiert wird. Durch eine präzise Prädiktion des Kurses mit Hilfe der Punkte des Fahrkorridors können diese Effekte ausgeschlossen werden. Für die Tests wurden die Warnzonen durch ein einfaches Rechteck abstrahiert. Eine Variation der abstrahierten Warnzone verbesserte die Trennung zwischen Objekten innerhalb und außerhalb der Fahrstrecke deutlich. Einige der in Abbildung 7.9 dargestellten Störungen konnten mit den als Rechteck abstrahierten Warnzonen nicht vermieden werden. Dies verdeutlicht die Notwendigkeit des Fahrkorridors.

Die Abbildung 7.10 zeigt eine Testfahrt mit einem Hindernis, bei der das Sensorsystem das Hindernis erkennt, das Fahrzeug allmählich abbremst und zum vollständigen Stillstand bringt. Beim Wegnehmen des Hindernisses entsteht aufgrund des umgehenden Losfahrens des Fahrzeugs eine neue Haltesituation. Für ein Sicherheitssystem muss eine kurze Wartezeit von 2 -



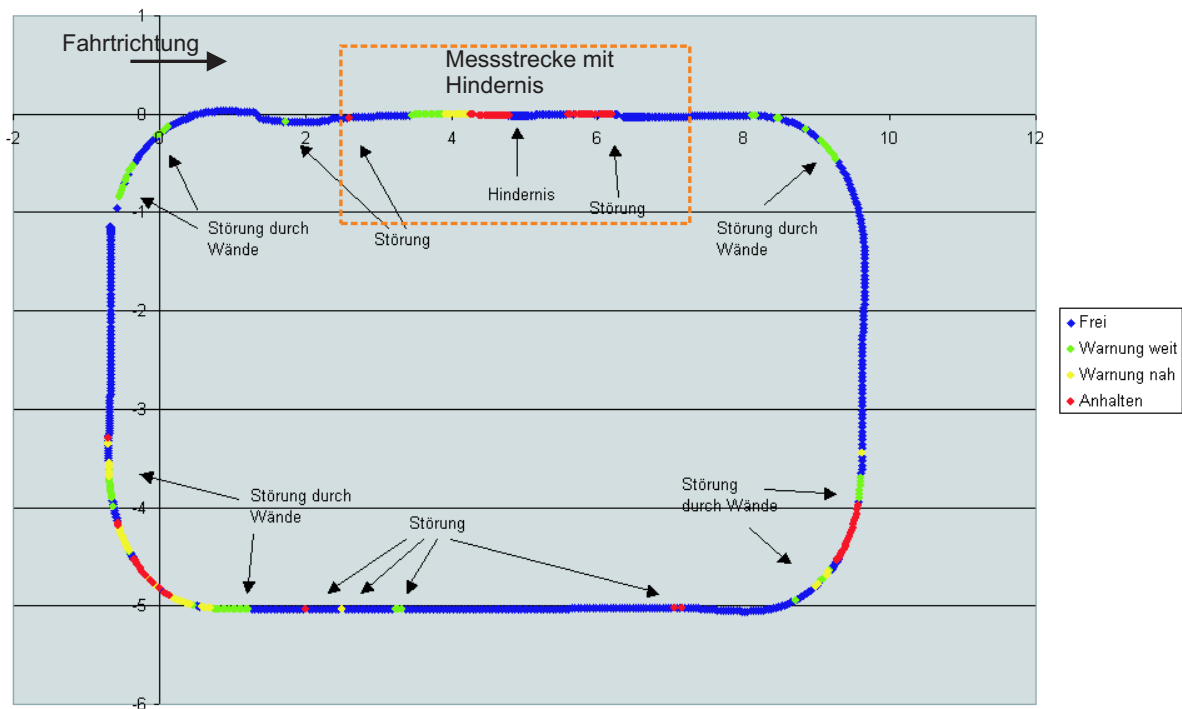


Abbildung 7.10: Testfahrt mit Hindernis in der Fahrstrecke

3 Sekunden nach der Freigabe des Weges nach einem Haltevorgang eingerichtet werden. Anschließend setzt das Fahrzeug die Runde selbstständig fort.

Durch Berücksichtigung der Fahrtrichtung bei der Hindernisdetektion konnten insbesondere in Kurven enge Durchfahrten realisiert werden.

## 7.5 Ergebnisse der Entwicklung einer Hindernisdetektion für fahrerlose Transportfahrzeuge

Die Hindernisdetektion wurde im Verlauf der Arbeit aufbauend auf die echtzeitfähige Stereobildauswertung entwickelt, implementiert und auf einem fahrerlosen Elektrofahrzeug erfolgreich getestet. Als Hardwareplattform wurde eine Platine mit zwei FPGAs und vier Kameralink-Anschlüssen verwendet.

Zusätzlich zur Stereobildauswertung auf Basis der Korrelatorkaskade wurde im FPGA die stochastische Clusterungsmethode zur Hindernisdetektion implementiert. Hierbei wurde der Großteil der Verarbeitung in Hardware realisiert und nur die Berechnung der 3D-Daten sowie die Kommunikation mit dem Fahrzeug für die Abarbeitung in Software vorgesehen.

Mit dieser Struktur konnte eine Verarbeitung der Bilddaten synchron zur Bildaufnahme mit 25 Bildern/s realisiert werden. Für die Testfahrt wurde aufgrund der niedrigen Helligkeit inner-

halb des Gebäudes eine höhere Integrationszeit verwendet und somit kamerabedingt nur eine Bildverarbeitung mit 10 Bildern/s realisiert.

Dabei wurde der Detektionsbereich in X-Richtung in Abhängigkeit von den Lenkbewegungen des Fahrzeugs angepasst, sodass enge Durchfahrten auch in Kurven selbstständig vom Fahrzeug durchgeführt werden konnten. Da die Kameras starr am Fahrzeug angebracht sind, erscheinen relevante Objekte in Kurvenfahrten am jeweiligen seitlichen Rand des Messbereichs, während in seinem Zentrum Objekte detektiert werden, die nicht auf dem Fahrweg des Fahrzeugs liegen. Bei den Testfahrten wurden alle Hindernisse erkannt. Eine optimale Abstimmung von Kamerasystem und Fahrzeug ist für einen Betrieb als Sicherheitssystem zwingend erforderlich.

Bei der Bearbeitung des Themengebietes wurde ebenfalls ein Spurwechselassistent mit der Stereobildauswertung in einem eingebetteten System auf Basis des hierarchischen Verfahrens mit der QNKKFMF entwickelt. Die Objektdetektion wurde ebenfalls durch ein statistisches Verfahren realisiert und am IESK entwickelt. Sie wird in Kaszubiak [41] beschrieben. Im Rahmen dieser Arbeiten wurden Messungen im dynamischen Fall auf Autobahnen [40] durchgeführt, die zeigen, dass die Resultate aus Kapitel 6 weitestgehend übertragen werden können. Weiterhin wurde das Verhalten bei Nachtfahrten [89] untersucht. Es zeigte sich, dass auf Autobahnen mit CMOS-Kameras statt der Fahrzeuge die Lampen der Fahrzeuge vermessen wurden. Somit kann das passive stereophotogrammetrische Messsystem für den Spurwechselassistenten mit Einschränkungen auch in der Dämmerung und in der Nacht eingesetzt werden. Es wäre jedoch der Einsatz von HDR-Kameras empfehlenswert.

Mit diesen Arbeiten wurden die in den vorangegangenen Kapiteln entwickelten und vorgestellten Verfahren (primär die Korrelatorkaskade mit der SADMF) in eine praktische Anwendung überführt und die vollständige Funktion innerhalb eines eingebetteten Systems erfolgreich getestet.

Damit stehen für die Umfelderkennung für verschiedene Anwendungsbereiche echtzeitfähige Implementierungen für die Stereobildauswertung zur Verfügung. Mit der in dieser Arbeit verwendeten Hardwareplattform (2\* Stratix EP1S60) sind, wie in Kapitel 5 beschrieben, 50 Bilder/s bei Verwendung des hierarchischen Verfahrens mit der QNKKFMF und 75 Bilder/s mit der SADMF für eine zur Bildaufnahme synchrone Verarbeitung möglich, während mit der Korrelatorkaskade auf Basis der SADMF Bildfrequenzen bis zu 150 Bilder/s erreicht werden können.

Unter Berücksichtigung der aktuellen Entwicklungen bei den FPGAs steht somit ein wesentlicher Bestandteil für echtzeitfähige Bildverarbeitungssysteme in einem weiten Anwendungsfeld zur Verfügung. Dies wird in der Zusammenfassung und dem Ausblick näher ausgeführt.

# Kapitel 8

## Zusammenfassung und Ausblick

Diese Arbeit befasst sich mit der Detektion und der Vermessung der Position von Objekten im Umfeld von Fahrzeugen auf Basis der Stereophotogrammetrie. Dazu wurden verschiedene Algorithmen und Plattformen (DSP, FPGA) miteinander verglichen und zwei Verfahren zur Echtzeitstereobildauswertung in einen FPGA implementiert.

Ausgehend von zwei Szenarien, einem Spurwechselassistenten und einem fahrerlosen Transportfahrzeug, wurden der Messbereich, die Konfiguration des Stereokamerasystems und die Echtzeitanforderung abgeleitet. Die Verarbeitung der Stereobilder erfolgt auf einem eingebetteten System mit Hilfe von FPGAs in Echtzeit.

Die Verwendung eines Stereokamerasystems zur 3D-Vermessung ermöglicht die sichere und zuverlässige Ermittlung der 3D-Informationen über einen großen Entfernungsbereich. Ein solches Messsystem kann vollständig passiv realisiert werden, wodurch die Beeinträchtigung der Umwelt minimiert wird. Schnell wechselnde Szenen, wie sie im Straßenverkehr oder bei fahrerlosen Transportfahrzeugen auftreten, können durch die hohe Bildfrequenz des Stereokamerasystems gut erfasst werden. Nachteil der stereophotogrammetrischen Messmethoden ist der hohe Aufwand für die Auswertung der Bilder zur Gewinnung der 3D-Informationen.

Für beide Szenarien ist der Normalfall der Stereophotogrammetrie sehr gut geeignet. Dadurch kann ein großer Messbereich, sowohl in der Entfernung als auch in lateraler Richtung, realisiert werden. Weiterhin können kamerabasierte Systeme gut die vorhandenen Informationen in einem vom Menschen geprägten Umfeld auswerten.

Das Ziel der Umfelderkundung mit berührungslose Messverfahren von zivilen autonomen Systemen ist im Wesentlichen eine Kollisionsvermeidung durch Ausweichen oder rechtzeitiges Abbremsen. Für beide Fälle genügt die Detektion, die Bestimmung der Position und der Abmaße eines Objektes. Eine konkrete Klassifizierung ist meist nicht erforderlich.

Der Spurwechselassistent warnt den Fahrer bei einem Spurwechsel auf mehrspurigen Fahrbahnen vor Kollisionen mit überholenden Fahrzeugen. Dazu ist eine Beobachtung des Rückraumes eines Fahrzeugs im Bereich von 10 - 150 m mit einer Bildfrequenz von 25 Hz erforderlich.

Der erforderliche Rechenaufwand ist aufgrund der hohen Abstraten und des aus dem großen Messbereich resultierenden großen Disparitätsbereiches sehr hoch. Daher wurde hierfür ein hierarchisches Verfahren zur Auswertung von Stereobildern entwickelt, welches die Änderung der Objektgröße über die Messbereiche berücksichtigt. Dazu wird eine Stereobildpyramide aufgebaut, deren Ebenen jeweils zueinander reduzierte Bildauflösungen haben. Die Korrelation erfolgt in allen Ebenen gleichzeitig. Anschließend werden die Ergebnisse kombiniert.

Das im Rahmen dieser Arbeit entwickelte und vorgestellte hierarchische Messverfahren ermöglicht eine starke Reduktion des Rechenaufwandes bzw. des Ressourcenaufwandes. Der datenflussorientierte Algorithmus des hierarchischen Verfahrens begünstigt eine Hardwareimplementierung durch seine starke Parallelisierbarkeit. Dadurch kann eine Auswertung der Stereobilder auch für große Messbereiche online zur Bildaufnahme bei hohen Bildauflösungen (ab 1024x1024 Pixel) und Bildfrequenzen über 25 Bilder/s in einem eingebetteten System mit Hilfe einer FPGA-Implementierung realisiert werden.

Durch die Anwendung von fahrerlosen Fahrzeugen im Außenbereich entstehen deutlich komplexere Situationen, z.B. durch nicht geschulte Personen. Daher ist eine verbesserte Hindernisdetektion erforderlich. Aufgrund der relativ niedrigen Geschwindigkeiten ist eine Vermessung in einem Bereich von 1 - 20 m vor dem Fahrzeug mit 10 Hz Bildfrequenz ausreichend. Aufgrund der höheren Flexibilität der Objekte im Umfeld ist ein Öffnungswinkel von mindestens 90° erforderlich.

Ist der Einsatz des hierarchischen Verfahrens aufgrund eines größeren Öffnungswinkels und der Gestaltung des Messbereiches nicht notwendig oder nicht möglich, kann eine lineare blockbasierte Korrelation verwendet werden. Bei diesem Ansatz steigt der Rechen- bzw. Ressourcenaufwand mit der Größe des Messbereiches stark an. Durch eine datenflussorientierte Implementierung mit Hilfe einer Korrelatorkaskade kann der Algorithmus an das Problem angepasst und echtzeitfähig realisiert werden.

Die Korrelatorkaskade ermöglicht eine direkte Zuführung der Daten des Suchbildes, während die Daten des Referenzbildes schrittweise durch die Stufen der Korrelatorkaskade geleitet werden. Dadurch kann eine Vermessung in einem sehr großen Disparitätsbereich erfolgen.

Beide in diesem Kapitel vorgestellten Verfahren nutzen die Parallelität der Operatoren und der Beziehungen zwischen den Daten, um eine Implementierung für eine kontinuierliche Verarbeitung von Kameradaten mit einem hohen Datendurchsatz in Hardware zu ermöglichen.

Die Korrelatorkaskade ist für eine Hardwareimplementierung optimiert. Sie kann für eine Portierung einer Vielzahl von Aufgaben der Bildverarbeitung in Hardware verwendet werden. Aufgrund der guten Skalierbarkeit kann das Verfahren schnell an neue Gegebenheiten angepasst werden.

Beim hierarchischen Verfahren wird die Auflösung des Bildes in den einzelnen Ebenen reduziert, so dass das Berechnungsverfahren in allen Ebenen identisch bleiben kann und dennoch die Objekte in der für sie erforderlichen Auflösung bearbeitet werden können. Es ermöglicht im Gegensatz zur Korrelatorkaskade eine effektive Implementierung von hierarchischen Korrelati-

onsverfahren auf Hardware- und Softwareplattformen. Ein im Rahmen der Arbeit eingereichtes Patent für das hierarchische Verfahren wurde im September 2008 erteilt.

Als Korrelationskriterien sind für die Auswertung von Stereobildsequenzen mit variierenden Beleuchtungsbedingungen für beide Verfahren die SADMF und die QNKKFMF vorteilhaft. Ihr Rechenaufwand wurde sowohl auf FPGAs (Stratix EP1S60) als auch auf DSP (Analog Devices Blackfin BF533 und TigerShark TS201) ermittelt. Während auf DSP nur die SADMF-Funktion für eine Echtzeitverarbeitung realisiert werden konnte, ist aufgrund einer starken Parallelisierung in FPGAs auch die Implementierung einer QNKKFMF möglich.

Anhand von Messungen im Nahbereich (1 - 20 m) und Fernbereich (10 - 150 m) mit entsprechend angepassten Aufnahmesystemen konnten die beiden Messverfahren in ihrer Leistungsfähigkeit verglichen werden. Hierbei zeigte sich die gute Leistungsfähigkeit des hierarchischen Verfahrens bei großen Messbereichen. Der Messfehler des hierarchischen Verfahrens in Z-Richtung liegt bei den gewählten Parametern immer unterhalb von 5 %, während der des linearen Verfahrens bei Messungen oberhalb von 60 m deutlich ansteigt. Im Nahbereich, insbesondere unterhalb von 10 m, bilden die Messungen über das lineare blockbasierte Korrelationsverfahren die wahre Entfernung genauer ab. Somit sollte es in diesen Fällen vorgezogen werden.

Die zwei in dieser Arbeit vorgestellten Verfahren konnten durch eine FPGA-Implementierung als System zur Echtzeitstereobildauswertung realisiert werden. Durch die Verarbeitung der Daten parallel zur Bildaufnahme, steht 70  $\mu s$  nach Beendigung der Bildaufnahme die Tiefenkarte zur Verfügung. Somit sind beide Verfahren für die in dieser Arbeit diskutierten Szenarien einsetzbar.

Dies wurde durch die Umsetzung und den Test einer Hindernisdetektion auf der Basis eines statistischen Clusterungsverfahrens untersucht. In Testfahrten mit einem elektrischen automatisierten Fahrzeug konnten die Genauigkeit und die Geschwindigkeit in der Vermessung erfolgreich überprüft werden.

Unabhängig vom Aufnahmesystem ist mit der QNKKFMF auf der in der Arbeit beschriebenen Hardwareplattform unter Verwendung des hierarchischen Verfahrens eine Stereobildauswertung bis zu einer Bildfrequenz von 50 Hz realisierbar.

Aufgrund des einfacheren Aufbaus der SADMF kann sie auf vielen Implementierungsplattformen sehr effektiv realisiert werden. Mit ihr kann auf der für die Arbeit verwendeten Hardware durch Einsatz der Korrelatorkaskade eine Bildfrequenz von maximal 150 Hz erreicht werden. Mit aktueller Hardware (z.B. Stratix IV) stehen damit zwei Verfahren zur Verfügung, die für die Online-Auswertung von Hochgeschwindigkeitsbildern verwendet werden können.

Die Berechnung einer Tiefenkarte ist im Allgemeinen nur ein Zwischenschritt für eine Anwendung. Auf Basis der FPGA-Implementierung der blockbasierten Korrelation wurde eine Hindernisdetektion für fahrerlose Transportfahrzeuge entwickelt, bei der Objekte mit Hilfe eines statistischen Clusterungsverfahrens erkannt wurden. Dazu wurde der vermessene Raum in regelmäßige Quader eingeteilt und die Anzahl der 3D-Punkte innerhalb der Quader bestimmt. Durch diese Vorgehensweise ergibt sich ein 3D-Histogramm. Anhand von Häufungen können erhabene Objekte ermittelt und ihre Position bestimmt werden. Diese Hindernisdetektion ist

einfach und ebenfalls datenflussorientiert in Hardware implementierbar. In der Arbeit wurde eine sehr niedrige Auflösung für die Clusterung verwendet. Durch Verwendung einer höheren Auflösung zur Berechnung der Tiefenkarte wird auch eine verbesserte Auflösung der Clusterung und dadurch eine höhere Genauigkeit in der Objektdetektion möglich.

Im Rahmen dieser Arbeit wurde auf zum einen die QNKKFMF mit dem hierarchischen Verfahren realisiert und mit einer statistischen Clusterungsmethode aus [38] kombiniert und im Labor als Sensorsystem für einen Spurwechselassistenten erfolgreich getestet.

Zum anderen wurde eine Korrelatorkaskade in Verbindung mit der SADMF implementiert und mit dem oben beschriebenen Clusterungsverfahren zu einer Hindernisdetektion für fahrerlose Fahrzeuge erweitert. Sie wurde anschließend in Verbindung mit einem fahrerlosen Fahrzeug erfolgreich getestet.

Durch eine Verwendung von mehreren Messsystemen, die auf unterschiedlichen Technologien basieren, kann die Zuverlässigkeit und der abgedeckte Bereich der Umfelderkennung deutlich erhöht werden. Je nach Anwendungsfall kann der Einsatz von Radar, Laserscannern oder Ultraschall-Sensoren sinnvoll sein.

Um eine Serienfertigung eines solchen Messsystems realisieren zu können, ist eine Untersuchung der Auswirkungen verschiedener Fehler bei der Anordnung des Kamerasystems im Stereonormalfall notwendig. Ziel ist es, die Fehler, die durch das Kamerasystem entstehen, soweit zu reduzieren, dass der verbleibende Fehler mit Hilfe einfacher Algorithmen ausgeglichen werden kann.

Für eingebettete Systeme, die primär Bildverarbeitung mit einer begrenzten Anzahl von Bildverarbeitungsoperationen durchführen, sollte die datenflussorientierte Implementierung in einem FPGA geprüft und stückweise realisiert werden. Die Entwicklung eines allgemeinen Verfahrens zur Bewertung der Relevanz einer Hardwareimplementierung wurde von Kaszubiak [39] bearbeitet. Diese Arbeit könnte in Verbindung mit der vorliegenden Arbeit zu einem allgemeinen Verfahren zur Portierung von softwareorientierten Algorithmen in Hardware weiterentwickelt werden.

Die in dieser Arbeit verwendete verhältnismäßig dünne Tiefenkarte ist für eine Umfelderkennung im Sinne einer Hindernisdetektion ausreichend. Für einige Anwendungen ist jedoch eine deutlich dichtere Tiefenkarte zur Bestimmung der äußeren Form erforderlich. Durch die technische Weiterentwicklung auf dem Gebiet der FPGAs wird eine Kombination der Korrelatorkaskade mit dem hierarchischen Verfahren ermöglicht. Somit kann theoretisch für jedes Pixel eines Bildes ein 3D-Wert effektiv und schnell berechnet werden. Dadurch könnte eine effektive und performante Implementierung für klassische Bildpyramiden für die Stereophotogrammetrie erreicht werden.

Dadurch wird auch die Anwendung deutlich komplexerer Ansätze, wie z.B. der der Carnegie Mellon University, in Hardware möglich. Durch eine verbesserte 3D-Auflösung und zusätzlich deutlich höhere Pixeltaktfrequenzen wird eine kontinuierliche Echtzeitverarbeitung mit Bildfrequenzen über 100 Hz möglich. Davon könnten vor allem Anwendungen in Medizin und Robotik, aber auch in der Automatisierungstechnik, profitieren.

# ANHANG





# Anhang A

## Algorithmen der Stereophotogrammetrie und Bildverarbeitung

Stereokamerasysteme stellen die minimale Konfiguration eines Mehrkamerasystems dar. Die Aufnahme und Auswertung aus Mehrkamerakonfigurationen wird unter dem Begriff Photogrammetrie zusammengefasst. Durch die Verwendung von mehreren Kameras bei der Bildaufnahme werden mehr unabhängige Daten gewonnen, wodurch sich zusätzliche Informationen auswerten lassen. Durch diese zusätzlichen Informationen können konkrete 3D-Informationen für alle Punkte, die in mindestens 2 Kameras vorhanden sind, bestimmt werden. Je nach Anwendung können durch die Verwendung von mehreren Kameras z.B. Verdeckungsprobleme aufgelöst oder Räume vollständig vermessen werden. Erforderlich sind hierfür insbesondere die Kenntnis der Position und der Winkel der Aufnahmen zueinander. Diese Informationen werden im Kalibrierprozess gewonnen.

Entwickelt wurde die Photogrammetrie kurz nach der Erfindung der Fotografie. Sie wurde durch den französischen Offizier Laussedat(1851) und den deutschen Architekten Meydenbauer (1858) unabhängig voneinander entwickelt und zunächst vorwiegend zur Vermessung von Gebäuden und zur Landvermessung eingesetzt. Im Laufe der Zeit wurden verschiedene Apparaturen zur effektiveren Bildaufnahme und Auswertung entwickelt. Die Auswertung erfolgte im folgenden Jahrhundert vorwiegend manuell unter Zuhilfenahme spezieller Geräte, zum Beispiel des Stereokomparators ( [55] S.20). Nähere Informationen zu der geschichtlichen Entwicklung der Stereophotogrammetrie werden zum Beispiel bei Luhmann ( [55] S. 18ff und Kraus ( [46] S. 2ff) ausgeführt.

Mit der kontinuierlichen Entwicklung opto-elektronischer Kameras gewann die digitale Bildaufnahme und -auswertung auch in anderen Anwendungsbereichen stetig an Bedeutung. Zum Beispiel wurden beim Eureka Prometheus Projekt [94] Stereokameras auch für die Umfelderkennung verwendet. Die Auswertung erfolgte auf Spezialrechnern im Laderaum des Fahrzeugs. Anschließend werden im nächsten Abschnitt die Möglichkeiten zur digitalen Auswertung von Stereobildpaaren näher beleuchtet.

## A.1 Innere und äußere Orientierung einer Kamera

Durch die innere und äußere Orientierung der Kameras wird der Zusammenhang zwischen den Bildkoordinatensystemen der beiden Kameras und dem Objekt- bzw. Weltkoordinatensystem hergestellt [55].

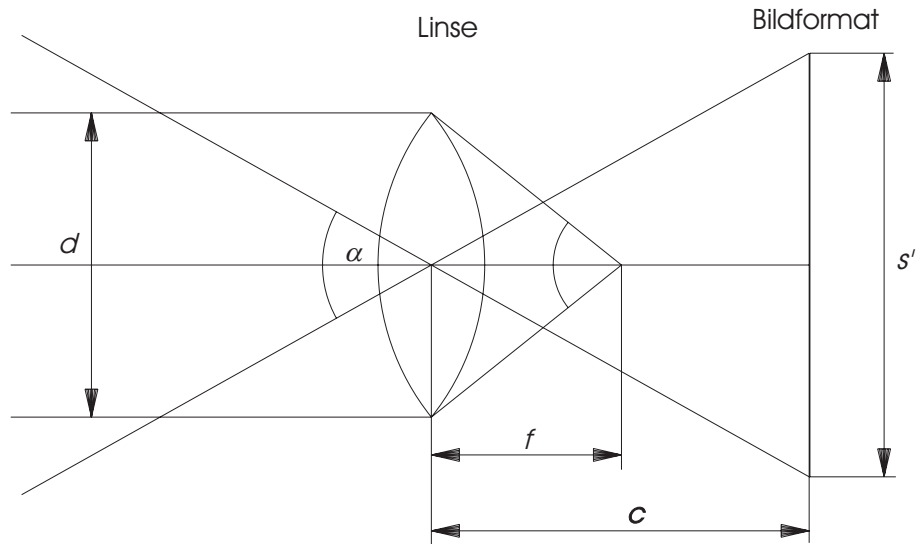


Abbildung A.1: Elemente der inneren Orientierung am Kameramodell

In Abb. A.1 ist eine vereinfachte Darstellung eines Kameramodells gegeben. Hier ist  $\alpha$  der Formatwinkel, der durch die Kamerakonstante  $c$  und die Breite der Abbildung  $s'$ , die bei digitalen Kameras durch die Sensorgröße festgelegt wird, ergibt. Die Kamerakonstante wird bei digitalen Kameras häufig in Pixeln angegeben, da die Länge  $c$  durch die Pixelbreite bzw. -höhe dividiert wird. Dadurch können Messungen im Bild über die Beziehungen innerhalb der Kamera auf die Originale abgebildet werden.  $f$  ist die Brennweite des Objektivs.

In Abb. A.2 sind die Elemente der inneren Orientierung einer Kamera und ihre Beziehung zum Weltkoordinatensystem dargestellt. Das Modell der Lochkamera begründet die hierbei zugrunde gelegte Zentralprojektion.  $O(X_0, Y_0, Z_0)$  stellt das Projektionszentrum dar. Die Winkel  $\phi, \kappa, \omega$  beschreiben die Verdrehung der Kamera gegenüber dem Weltkoordinatensystem. Damit sind die Elemente der äußeren Orientierung beschrieben.

Die Elemente der inneren Orientierung sind der Bildhauptpunkt  $H'(x'_0, y'_0)$  sowie die Kamerakonstante  $c_k$ . Je nach Genauigkeit des Modells werden noch die Parameter  $dx, dy$  zur Beschreibung der Bildverzerrung durch die Linse hinzugefügt. Der Maßstabsfaktor  $m$  beschreibt die unterschiedliche Größe zwischen beiden Abbildungen bei Kameras mit unterschiedlichen Parametern. Bei identischen Kameras ist  $m$  gleich eins.

Damit ist eine konkrete Beziehung zwischen einem Objektpunkt  $P(X, Y, Z)$  im Weltkoordinatensystem und seiner Abbildung  $P'(x', y')$  im Bildkoordinatensystem hergestellt. Sie kann durch eine Gerade im Raum beschrieben werden.

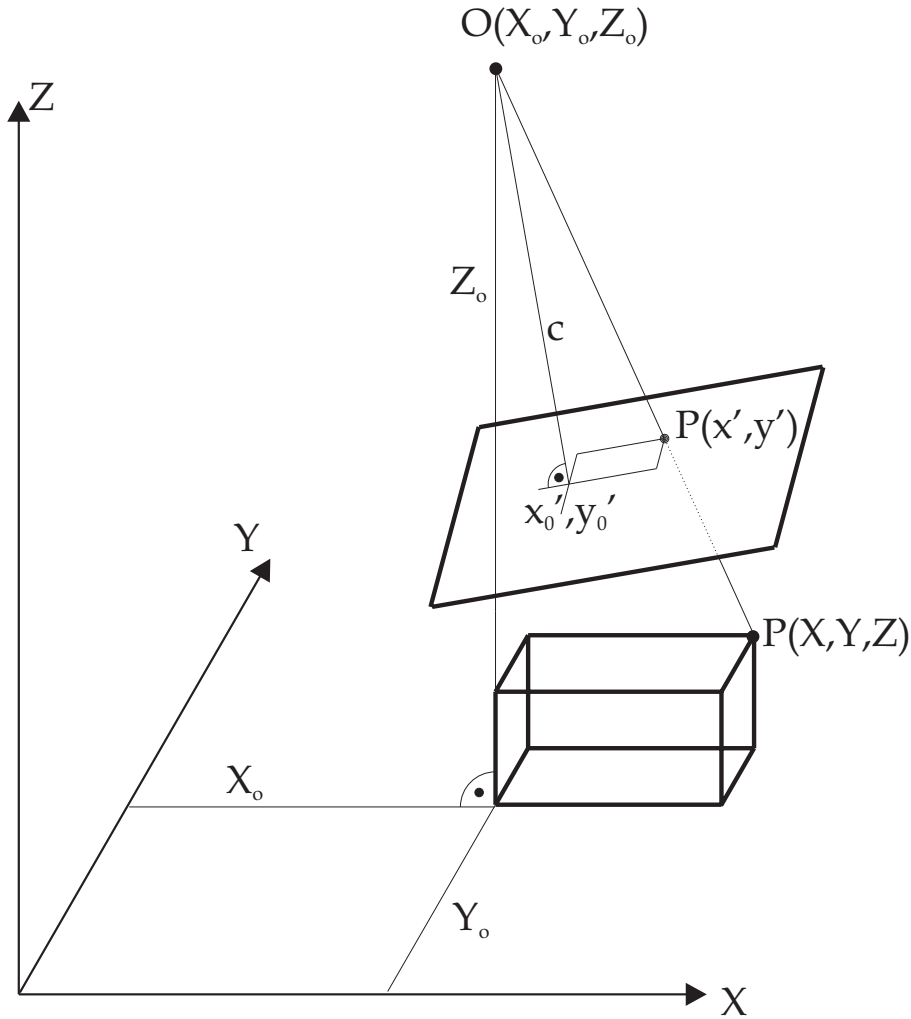


Abbildung A.2: Elemente der inneren und äußeren Orientierung

$$P = P_0 + m \cdot R \cdot p' \quad (\text{A.1})$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} + m \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ -c \end{bmatrix} \quad (\text{A.2})$$

Durch Umstellen ergeben sich die Kollinearitätsgleichungen [A.3](#) und [A.4](#).

$$x' = x'_0 - c_k \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + dx \quad (\text{A.3})$$

$$y' = y'_0 - c_k \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + dy \quad (\text{A.4})$$

$r_{ii}$  mit  $i \in N; 1 \leq i \leq 3$  sind die Elemente der Rotationsmatrix, welche die Verdrehung

des Bildes und des Weltkoordinatensystems zueinander beschreibt. Sie ergibt sich durch die Matrizenmultiplikation der Einzeldrehungen (siehe Gleichung A.5).

$$\begin{aligned}
 R &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_\omega \cdot R_\phi \cdot R_\kappa & (A.5) \\
 &= \begin{bmatrix} \cos \phi \cos \kappa & -\sin \phi \sin \kappa & \sin \phi \\ \cos \omega \sin \kappa + \sin \omega \sin \phi \cos \kappa & \cos \omega \cos \kappa - \sin \omega \sin \phi \sin \kappa & -\sin \omega \cos \phi \\ \sin \omega \sin \kappa - \cos \omega \sin \phi \cos \kappa & \sin \omega \cos \kappa + \cos \omega \sin \phi \sin \kappa & \cos \omega \cos \phi \end{bmatrix}
 \end{aligned}$$

Die Kollinearitätsgleichungen bilden die Grundlage für die verschiedenen Kalibrierverfahren. Auf die Kalibrierverfahren wird in Abschnitt A.6 näher eingegangen. Zur vollständigen Rekonstruktion der Position des Punktes  $P$  ist zusätzlich zur bekannten inneren und äußeren Orientierung und der Position der Abbildung in einer Kamera noch mindestens ein weiteres Merkmal erforderlich. Dies kann bei monokularen Verfahren z.B. die bekannte Größe des Objektes bei einer bestimmten Entfernung sein. In der Photogrammetrie wird hierfür die Auswertung mindestens eines weiteren Bildes verwendet, wodurch keine a priori Informationen über das zu vermessende Objekt zur vollständigen Bestimmung der Position des Objektpunktes notwendig sind.

## A.2 Der allgemeine Fall der Stereophotogrammetrie

Mit den Mitteln der Photogrammetrie können beliebig viele Aufnahmen aus unterschiedlichen Winkeln miteinander kombiniert werden. Die minimale Konfiguration besteht aus einem Kamerasystem mit zwei Kameras. Die Bestimmung der 3D-Koordinaten eines Objektpunktes erfolgt durch die Zuordnung homologer Punkte. Homologe Punkte sind Punkte, deren Abbildungen auf dem gleichen Ursprung basieren.

Der allgemeine Fall der Stereophotogrammetrie beschreibt eine freie Anordnung der Kameras im Raum. Bei bekannter innerer und äußerer Orientierung kann durch Anwendung des räumlichen Vorwärtsschnittes, beschrieben in Albertz&Kreiling [4] und Luhmann [55], die Position eines Raumpunktes mittels der Positionen seiner beiden Abbildungen ermittelt werden.

Die Position des Objektpunktes im Raum wird durch zwei Geraden näherungsweise bestimmt.

$$X' = X_{01} + R_1 \cdot x' \quad (A.6)$$

$$X'' = X_{02} + R_2 \cdot x'' \quad (A.7)$$

Im allgemeinen Fall der Stereophotogrammetrie sind die beiden Geraden windschief. Wird eine Ebene in XY-Richtung in der Entfernung  $Z$  des Punktes  $P$  definiert, schneiden die beiden Gera-

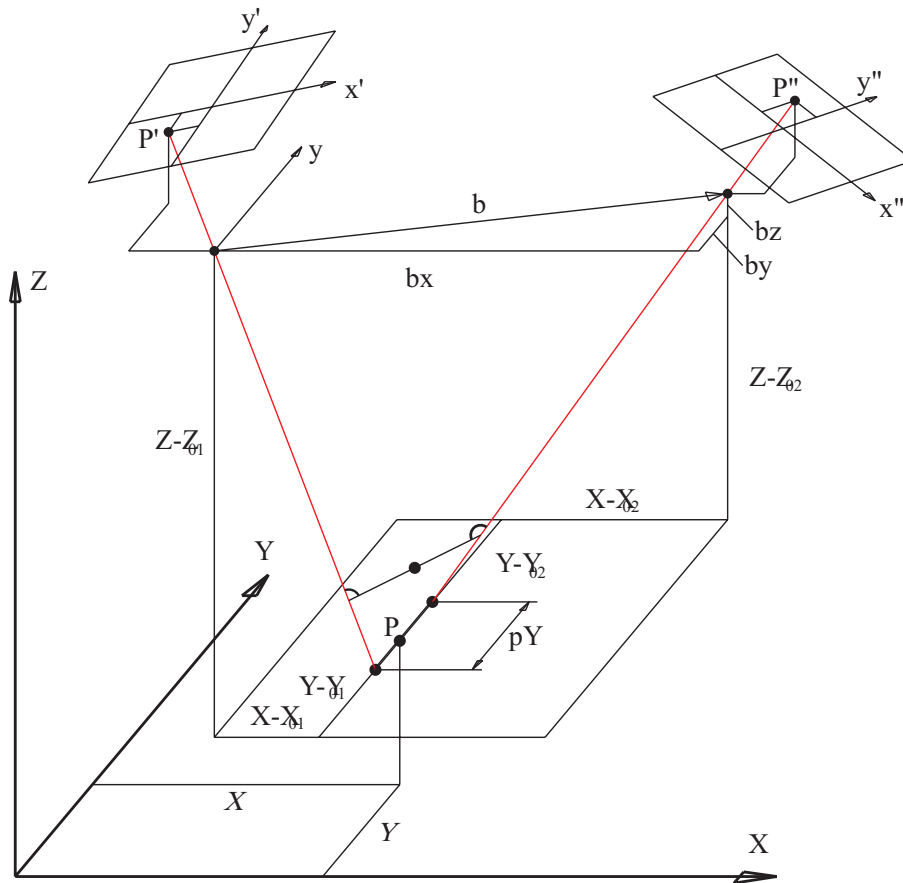


Abbildung A.3: Ermittlung der 3D-Position im allgemeinen Fall der Stereophotogrammetrie

den diese XY-Ebene in der Entfernung  $Z$ . Somit wird der Punkt  $P$  durch folgende Komponenten bestimmt:

$$X = X_1 = X_2 \quad (\text{A.8})$$

$$Z = Z_1 = Z_2 \quad (\text{A.9})$$

$$Y = \frac{Y_1 + Y_2}{2} \quad (\text{A.10})$$

Dadurch können die Koordinaten des Raumpunktes  $P$  wie folgt bestimmt werden:

$$\begin{aligned} X &= X_{01} + \lambda \cdot (X' - X_{01}) \\ Z &= Z_{01} + \lambda \cdot (Z' - Z_{01}) \\ Y_1 &= Y_{01} + \lambda \cdot (Y' - Y_{01}) \\ Y_2 &= Y_{02} + \mu \cdot (Y' - Y_{02}) \\ Y &= \frac{Y_1 + Y_2}{2} \\ pY &= Y_2 - Y_1 \end{aligned}$$

Mittels der Basiskomponenten  $bx$ ,  $by$ ,  $bz$  können die Maßstabsfaktoren  $\lambda$  und  $\mu$  bestimmt werden.

$$\begin{aligned} bx &= X_{02} - X_{01} \\ by &= Y_{02} - Y_{01} \\ bz &= Z_{02} - Z_{01} \\ \lambda &= \frac{bx \cdot (Z'' - Z_{02}) - bz \cdot (X'' - X_{02})}{(X' - X_{01}) \cdot (Z'' - Z_{02}) - (X'' - X_{02}) \cdot (Z' - Z_{01})} \\ \mu &= \frac{bx \cdot (Z'' - Z_{02}) - bz \cdot (X'' - X_{02})}{(X' - X_{01}) \cdot (Z'' - Z_{02}) - (X'' - X_{02}) \cdot (Z' - Z_{01})} \end{aligned}$$

Dieses Verfahren stellt keine exakte Methode zur Bestimmung der Koordinaten des Objektpunktes dar. Als Gütekriterium kann  $pY$  verwendet werden. Wenn die beiden Geraden sich schneiden, nimmt es den Wert Null an. In diesem Fall ist die Position exakt bestimmt. In Luhmann [55] Abschnitt 4.3.3.1 wird eine Methode des Vorwärtsschnittes durch ein überbestimmtes Ausgleichsproblem beschrieben. Dabei wird auf eine Mehrbildauswertung zurückgegriffen.

Der allgemeine Fall der Stereogrammetrie kommt ohne aufwändige Justage des Kamerasystems aus. Dafür muss jedoch eine deutlich aufwändigere Kalibrierung durchgeführt werden. Sowohl der einfache Ansatz für den räumlichen Vorwärtsschnitt, als auch der komplexere über das Ausgleichsverfahren sind aufgrund des Rechenaufwandes für eine Auswertung in Echtzeit nur bedingt geeignet. Abgesehen davon muss die Kalibrierung des Kamerasystems vor der Echtzeitauswertung vorgenommen werden. Für eine Offlineauswertung ist es jedoch eine adäquate und sichere Methode.

### A.3 Der Normalfall der Stereogrammetrie

Durch Ausrichtung der Kameras zueinander kann der allgemeine Fall zum Normalfall der Stereogrammetrie spezialisiert werden.

Dabei werden zwei möglichst identische Kameras so ausgerichtet, dass die Projektionsebene beider Kameras in der gleichen Ebene liegt und alle Drehwinkel zwischen den beiden Kamerakoordinationssystemen Null sind. Weiterhin stellt die Basisbreite  $b$ , beschrieben durch  $B = X_{02} - X_{01}$  den einzigen Versatz der beiden Kameras zueinander dar. Wird der Ursprung des Weltkoordinatensystems in das Projektionszentrum  $O'$  der ersten Kamera gelegt, können die 3D-Koordinaten des Objektpunktes durch Zusammenhänge in den Gleichungen A.11, A.12, A.13 beschrieben werden.

Durch diese Voraussetzungen reduziert sich die Suche auf den Versatz in x-Richtung. Wird der Versatz in x-Richtung als Disparität  $\Delta u = x' - x''$  definiert, ergeben sich folgende Zusammenhänge für die Koordinaten des Punktes P. Dieser Zusammenhang ergibt sich für die obigen Annahmen auch aus den Kollinearitätsgleichungen [55].

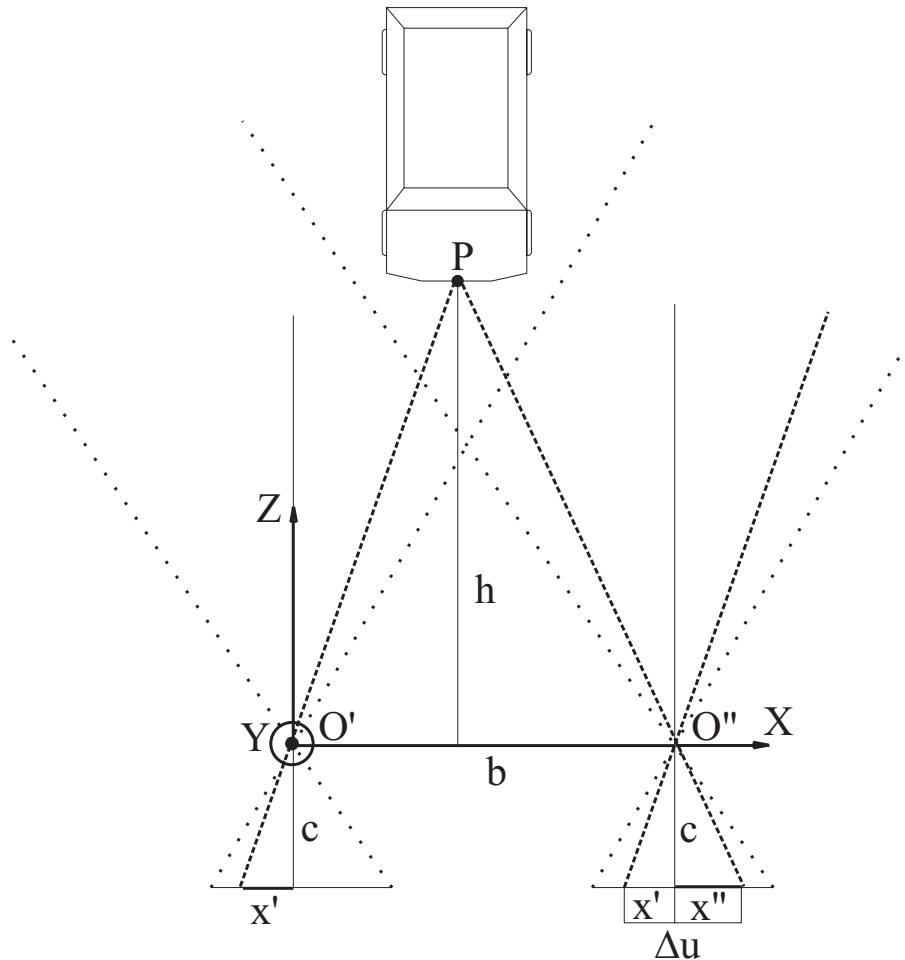


Abbildung A.4: Ermittlung der 3D-Position im Normalfall der Stereophotogrammetrie

$$X = \frac{b \cdot x'}{\Delta u} \quad (\text{A.11})$$

$$Y = \frac{b \cdot y'}{\Delta u} \quad (\text{A.12})$$

$$Z = \frac{b \cdot c}{\Delta u} \quad (\text{A.13})$$

Durch diese Anordnung wird die Auswertung wesentlich vereinfacht. Der Aufwand für die Justage der Kameras steigt dadurch jedoch wesentlich. Es müssen alle Drehwinkel beider Kameras und der Versatz in X- und in Z-Richtung frei variiert werden können. Während der laufenden Messung hätte eine Veränderung dieser Parameter jedoch einen Messfehler zur Folge. Für eingebettete Systeme kommt somit nur eine sehr exakte Fertigung und Fixierung der Kameras in einem festen Rahmen in Frage.

## A.4 Rektifizierung

Die Rektifizierung [11] beschreibt ein Verfahren, um Bilder, die im allgemeinen Fall der Stereophotogrammetrie aufgenommen wurden, mit dem Normalfall der Stereophotogrammetrie zu verarbeiten. Dazu wird mindestens eines der beiden Bilder so verzerrt, dass der Normalfall erfüllt ist. Im Gegensatz zum Vorwärtsschnitt, bei dem zuerst die homologen Punkte bestimmt werden und dann die Elemente der inneren und äußeren Orientierung angewendet werden, erfolgt zunächst eine Umrechnung der Bilder und anschließend eine vereinfachte Suche und Auswertung. Die Transformation wird entsprechend auf Basis der Kalibrierung durchgeführt. Für Systeme, die sich dem Normalfall der Stereophotogrammetrie sehr gut annähern, kann eine Rektifizierung hardwareunterstützt in Echtzeit realisiert werden [74]. Die Rektifizierung wird von T. Luhmann [55] auch als Berechnung von Normalbildern bezeichnet.

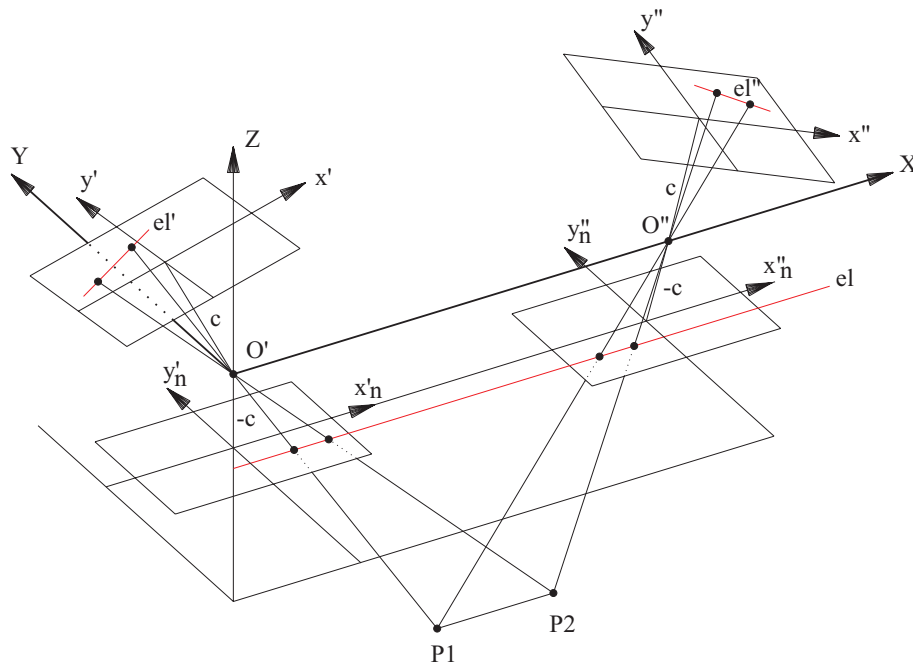


Abbildung A.5: Rektifizierung eines Stereobildpaares

Zur Rektifizierung werden zwei Bildebenen (die Normalbilder) angenommen, deren Projektionszentrum in dem der wahren Kameras liegt, die jedoch den Normalfall der Stereophotogrammetrie erfüllen. Die Grauwerte werden durch eine Koordinatentransformation in die Bildkoordinaten der Normalbilder umgerechnet. Das Verfahren kann für konvergente Stereobilder angewendet werden. Bei konvergenten Stereobildaufnahmen sind die Kameraachsen bei der Aufnahme zueinander geneigt.

Die Rektifizierung wirkt sich auch auf die Epipolargeometrie aus, welche nachfolgend beschrieben wird.



## A.5 Epipolargeometrie

Die Epipolargeometrie [16] beschreibt die Lage der homologen Punkte durch eine Ebene durch beide Projektionszentren und einen Bildpunkt bzw. den Objektpunkt. Dadurch ergibt sich für die zweite Kamera ein Bereich im Bild, in dem sich der entsprechend korrespondierende Punkt befinden wird. Im allgemeinen Fall der Stereophotogrammetrie ergeben sich zu den Achsen der Bildkoordinatensysteme windschief verlaufende Linien in den Bildebenen. Dieser Bereich wird im Allgemeinen als Epipolarlinie oder Kernlinie bezeichnet (siehe Abbildung A.6).

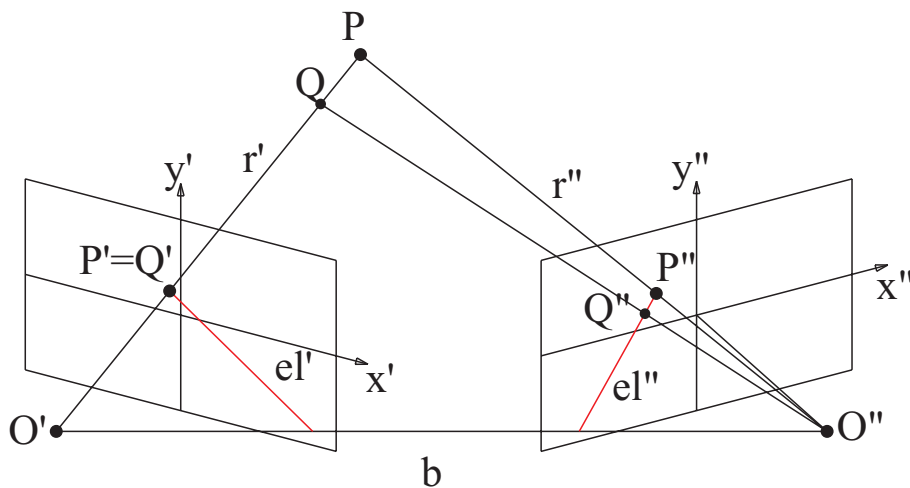


Abbildung A.6: Epipolargeometrie im allgemeinen Fall der Stereophotogrammetrie

Bei Verwendung des Normalfalles der Stereophotogrammetrie verlaufen aufgrund der speziellen Annahmen über die Koordinatensysteme (siehe Abschnitt A.3) Epipolarlinien, in x-Richtung, wie in Abbildung A.7 dargestellt. Dies vereinfacht die Suche nach den korrespondierenden Punkten. Im Normalfall kann die Suche entsprechend in Richtung der Zeilen begrenzt werden.

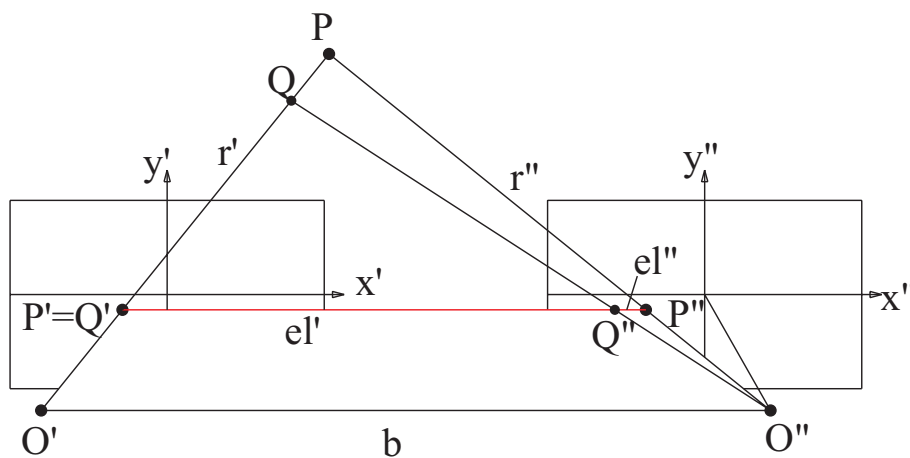


Abbildung A.7: Epipolargeometrie im Normalfall der Stereophotogrammetrie

Durch die Vereinfachungen bei der Ermittlung der 3D-Koordinaten und der Suche nach den

korrespondierenden Punkten eignet sich der Stereonormalfall sehr gut für die Auswertung mit eingebetteten Systemen.

## A.6 Kalibrierung und Fehlerbetrachtung von stereogrammetrischen Systemen

Die Kalibrierung eines Stereokamerasystem [25] ist für eine sichere und zuverlässige Auswertung der Stereobilder unerlässlich. Es gibt unterschiedliche Herangehensweisen zur Kalibrierung eines Stereokamerasystems. Zum einen können beide Kameras separat kalibriert werden. Dadurch ist die innere und äußere Orientierung beider Kameras bekannt und die Bilder können entsprechend ausgewertet werden. Bei der zweiten Variante werden die beiden Kameras relativ zueinander kalibriert. Dabei wird ausgehend von einer relativen Orientierung der beiden Kameras mit Hilfe des Modells der Anordnung eine absolute Orientierung ermittelt. Anschließend erfolgt die äußere Orientierung des Stereokamerasystems.

Während bei der getrennten Orientierung der beiden Kameras zunächst jede Kamera für sich kalibriert wird und sich dadurch über das Objektkoordinatensystem die Beziehung ergibt, erfolgt die Bestimmung der Orientierung bei dem gemeinsamen Verfahren für beide Bilder in einem Schritt. Das Verfahren wird auch Doppelbildeinschaltung genannt und verwendet für die Bestimmung der Orientierung die Bündelblockausgleichung.

Zunächst werden verschiedene Verfahren der Kalibrierung einer Kamera erläutert. Es werden hierzu drei Verfahren prinzipiell unterschieden: die Laborkalibrierung, die Testfeldkalibrierung und die Simultankalibrierung.

**Laborkalibrierung** Bei der Laborkalibrierung erfolgt die Ermittlung der Parameter der inneren Orientierung beim Hersteller der Kameras mit Kollimatoren oder Goniometern. Dabei werden Richtungen oder Winkel der Bildstrahlen durch das Objektiv der Kameras hindurch gemessen. Die Ergebnisse werden im Kalibrierungszertifikat niedergelegt. Bei der späteren Verwendung der Kameras ist daher keine Kalibrierung der inneren Orientierung notwendig. Unter Berücksichtigung der zeitlichen Änderung der Parameter werden vom Hersteller Fristen zur Neukalibrierung vorgegeben. Dieses Verfahren eignet sich nur für Systeme mit stabiler Mechanik [55].

**Testfeldkalibrierung** Zur Bestimmung der Parameter der Kameraorientierung mittels der Testfeldkalibrierung wird ein Testfeld mit ausreichender Anzahl an Referenzpunkten aufgenommen, wobei die Koordinaten der Testpunkte zueinander bekannt sein müssen. Aus diesen bekannten Koordinaten können mittels des Verfahrens des räumlichen Rückwärtsschnittes oder der Bündelblockausgleichung die gesuchten kameraspezifischen Größen ermittelt werden.

Die Testfeldkalibrierung muss unter Berücksichtigungen der zeitlichen Konstanz der Orientie-

rungsparameter zeitlich nah zur Objektaufnahme oder direkt während der Objektaufnahme durch gleichzeitige Aufnahme von Objekt und Passpunktfeld erfolgen.

Die lokalen Koordinaten der Passpunkte des Testfeldes können zur Datumsfestlegung des Objektkoordinatensystems genutzt werden.

**Simultankalibrierung** Gegenüber den obengenannten Verfahren muss bei der Simultankalibrierung der Aufnahmegegenstand kein Testfeld mit einer großen Anzahl bekannter Passpunkte, sondern kann auch das zu vermessende Objekt selbst sein. Dadurch ist es nicht notwendig gesonderte Kalibrieraufnahmen anzufertigen. Lediglich die Datumsfestlegung des Objektkoordinatensystems muss durch Passpunkte erfolgen. Da die innere Orientierung exakt für den Zeitpunkt der Objektaufnahme bestimmt wird, erlaubt dieses Verfahren höchste Genauigkeiten bei der Objektauswertung.

Wird die Simultankalibrierung nur mit photogrammetrischen Beobachtungen ausgeführt, spricht man von Selbstkalibrierung.

**Räumlicher Rückwärtsschnitt** Der räumliche Rückwärtsschnitt ist ein rechnerisches Verfahren zur Ermittlung der inneren und äußeren Orientierung einer Kamera. Auf Basis der Kollinearitätsgleichungen werden Gleichungen zur iterativen Optimierung erstellt. Es wird zwischen der Durchführung bei bekannter und bei unbekannter innerer Orientierung unterschieden.

Die Kollinearitätsgleichungen werden zu Verbesserungsgleichungen umgestellt und anschließend durch eine Taylor-Entwicklung linearisiert.

**Bekannte innere Orientierung** Beim räumlichen Rückwärtsschnitt werden bei bekannter innerer Orientierung nur die Parameter der äußeren Orientierung mit Hilfe von drei Passpunkten bestimmt. Durch jeden der Punkte können zwei Gleichungen zur iterativen Optimierung erstellt werden. Mit Hilfe dieser 6 Gleichungen können die 6 Unbekannten der äußeren Orientierung  $(X_0, Y_0, Z_0, \omega, \phi, \kappa)$  bestimmt werden. Dieses Verfahren ist im Wesentlichen für Kameras mit bereits bestimmter innerer Orientierung, wie z.B. spezieller Messkameras relevant.

**Unbekannte innere Orientierung** Zusätzlich zur äußeren Orientierung müssen beim Fall des Rückwärtsschnittes mit unbekannter innerer Orientierung die Elemente der inneren Orientierung bestimmt werden. Neben  $-c, x'_0, y'_0$  müssen ebenfalls die Verzeichnungsparameter  $dx'$  und  $dy'$  bestimmt werden. Prinzipiell können die 11 Parameter durch 6 Passpunkte mit einer Beobachtung bestimmt werden.

Als Verfahren zur Ermittlung der Elemente der Kameraorientierung mittels des räumlichen Rückwärtsschnittes können der "Rückwärtsschnitt mit minimaler Objektinformation" nach Kyle [50] und Wrobel [100] und das Verfahren der "Direkten Linearen Transformation" nach Abdel-Aziz und Karara [1] genannt werden. Es handelt sich in beiden Fällen um iterative Verfahren bei denen die Genauigkeit von der Anzahl und Verteilung der Punkte bestimmt wird. Die 6 Punkte

dürfen weder in einer Linie noch in einer Ebene liegen. Dadurch würde das Gleichungssystem instabil oder singular und die Bestimmung der Orientierungsparameter schlägt fehl oder ist sehr stark Fehler behaftet.

**Bündelblockausgleichung** Bei der Bündelblockausgleichung werden viele Passpunkte oder bekannte Strecken aus mehreren Beobachtungen rechnerisch miteinander kombiniert, um sehr exakte Werte für die Orientierung von Kameras zu erhalten. Durch den nahezu unbegrenzten Einsatz neuer Passpunkte lassen sich mit der Methode auch sehr komplexe Mehrkameraanordnungen kalibrieren.

Die Methode der Bündelblockausgleichung basiert ebenfalls auf den Kollinearitätsgleichungen und kann als Verallgemeinerung des räumlichen Rückwärtsschnittes betrachtet werden. Durch die Anzahl der Passpunkte bzw. die entsprechend hohe Anzahl unabhängiger Beobachtungen ergeben sich sehr viele Gleichungen zur Ausgleichung der Parameter. Ebenso wie die erreichbare Genauigkeit hängt auch der Rechenaufwand von der Anzahl der Passpunkte und der Beobachtungen ab. Die Kalibrierung mittels der Bündelblockausgleichung kann nur nach der Bildaufnahme mit leistungsfähigen Rechnern durchgeführt werden [46].

Die Verbesserungsgleichungen für die Bündeltriangulation unterscheiden sich in der Berücksichtigung vieler Passpunkte, multipler Kameras sowie der getrennten Betrachtung der äußeren Orientierung sämtlicher Bilder zum Aufnahmezeitpunkt (siehe [55] S.260) von denen des Rückwärtsschnittes. Damit können sowohl Luftbildaufnahmen kalibriert und ausgewertet werden als auch komplexe räumliche Anordnungen von Kameras in Mehrkamerasystemen. [83] [85]

## A.7 Fehlerbetrachtung von stereogrammetrischen Systemen

Die Genauigkeit eines Stereokamerasystems ist vor allem von der Geometrie der Aufnahmekonfiguration und der erreichbaren Bildmessgenauigkeit abhängig [25]. Oft erhält man eine sehr gute Abschätzung, indem man die Gesetze der Fehlerfortpflanzung auf die Formeln des Normalfalles der Stereogrammetrie (Gleichung A.11ff) anwendet. Der mittlere Fehler  $\sigma_z$  in der meist dominanten Z-Richtung ist dann

$$\sigma_z = \frac{Z^2}{c \cdot B} \cdot \sigma_u \quad (\text{A.14})$$

Die Messung wird somit umso genauer, je geringer der Objektabstand, je größer die Kamerakonstante, je breiter der Basisabstand der Kameras ist und je genauer die Bildkoordinaten bestimmt werden können [56].

Durch die Auflösung der Kameras ergibt sich eine Quantelung der messbaren Entfernung. Ab einer bestimmten Objektdistanz liegt die Disparität unterhalb einer Pixelbreite. Mit Hilfe einer

Subpixelinterpolation kann eine weitere Steigerung der Auflösung erreicht werden. Die Präzision der Messung wird jedoch auch durch optische Einflüsse der Luft bestimmt, wodurch dieser Effekt bei großen Entfernungen relativiert wird. Der Öffnungswinkel der Kameras legt fest, wie weit die Umsicht des Systems ist. Bei großen Öffnungswinkeln entstehen jedoch auch meist unerwünschte Verzerrungen, die durch entsprechende nichtlineare Korrekturterme in Gleichung A.3 und Gleichung A.4 korrigiert werden müssen. Dies ist bei Präzisionsmessungen [83] grundsätzlich erforderlich.

$$\Delta u_r = \Delta u + \frac{\frac{1}{2}(C_{(1)} - C_{(-1)})}{2 \cdot C_{(0)} - C_{(1)} - C_{(-1)}} \quad (\text{A.15})$$

Die Genauigkeit der Bestimmung der Disparität kann sinnvoll auf 1/10 Pixel ausgeweitet werden. Hierzu werden die Korrelationswerte vor und nach dem Extremum gespeichert und mit Hilfe der Gleichung A.15 eine Bestimmung der Position des Maximums der so entstehenden quadratischen Funktion durchgeführt. Dadurch kann die Position des Maximums auf ca. 1/10 Pixel sinnvoll bestimmt werden.



# Anhang B

## Vergleich der Eigenschaften verschiedener Korrelationsverfahren

Dieser Abschnitt stellt verschiedene Korrelationskriterien bezüglich der Qualität ihrer Ergebnisse bei der Auswertung von Stereobildern gegenüber. Nach einer kurzen Vorstellung der Kriterien werden einige Kriterien auf synthetische und reale Bilder angewendet und bezüglich ihrer Eigenschaften miteinander verglichen.

### B.1 Vorstellung verschiedener Korrelationskriterien

Um rechnerisch eine Aussage über die Ähnlichkeit bzw. die statistische Abhängigkeit zweier Signalfolgen treffen zu können, werden Korrelationskriterien verwendet. Der Einsatzbereich der Korrelationskriterien umfasst die gesamte digitale Signalverarbeitung, begonnen mit der digitalen Korrelation von Signalen Mobilfunk bis hin zur Bildverarbeitung. Aber auch Messreihen im Allgemeinen werden mit Hilfe von Korrelationskriterien auf Ähnlichkeit hin untersucht.

Korrelationskriterien können in Anlehnung an [9] in Ähnlichkeits- (KKF, NKKFMF) und Unähnlichkeitskriterien (SAD, SSD, NSSDMF) unterteilt werden. Im Folgenden werden die für diese Arbeit relevanten Kriterien vorgestellt. In Kapitel 4 werden einige Kriterien bezüglich ihres Rechenaufwandes und ihrer Eignung zur Echtzeitimplementierung für verschiedene Plattformen untersucht.

#### B.1.0.4 Unähnlichkeitskriterien

Eine große Familie von Fehler- bzw. Unähnlichkeitskriterien stellen die Summe der Differenzenquadrate (sum of squared differences - SSD) bzw. Summe der absoluten Differenzen (sum of absolute differences - SAD) dar. Bei ihnen werden, wie die Namen schon sagen, die Differenzen der Signalfolgen aufsummiert. Der enorme Vorteil dieser Verfahren ist ihre einfache Realisierung. Jedoch sind sie gegenüber Helligkeitsdifferenzen der Bilder empfindlich. Daher

sollte entweder eine gute und für beide Stereokameras synchron realisierte Helligkeitsregelung eingesetzt werden, oder statt der Pixel  $P_r(i, j)$  und  $F(\xi + i, \eta + j)$  die mittelwertfreien Pixel  $\overline{P_r(i, j)}$  und  $\overline{F(\xi + i, \eta + j)}$  verwendet werden.  $m, n$  beschreiben die Ausdehnung des Referenz- bzw. Suchfensters und  $\xi, \eta$  beschreiben die Verschiebung des Suchfensters gegenüber der Startposition.

$$SAD(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} |P_r(i, j) - F(\xi + i, \eta + j)| \quad (\text{B.1})$$

$$SSD(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (P_r(i, j) - F(\xi + i, \eta + j))^2 \quad (\text{B.2})$$

Um ein robustes Unähnlichkeitsmass zu realisieren, welches eindeutige Minima erzeugt und dessen Wertebereich überschaubarer ist, wurde an der INRIA (das französische Institut für Informatik und Automatisierungstechnik) das Normierungssystem der NKKFMF mit der SSDMF verknüpft, wodurch die normierte mittelwertfreie Summe der Differenzenquadrate (NSSDMF) entsteht. Ein Nachteil der Unähnlichkeitsmaße ist der nach oben hin unbeschränkte Wertebereich. Nach unten hin stellt die Null das absolute Minimum dar und drückt vollständige Übereinstimmung aus.

$$NSSDMF(\xi, \eta) = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} - \overline{F(\xi + i, \eta + j)} \right)^2}{\sqrt{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2}} \quad (\text{B.3})$$

### B.1.0.5 Ähnlichkeitskriterien

Eine weitere große Familie stellen die Kreuz- und Autokorrelationsfunktion dar. Sie gehören zu den Ähnlichkeitsmaßen. Bei der Autokorrelationsfunktion wird innerhalb eines Bildes ein bestimmter Block mit seiner Umgebung verglichen und so die statistische Abhängigkeit untersucht, sie wird z.B. in den klassischen Auswerterverfahren der Photogrammetrie zur Auswahl eindeutiger Punkte verwendet.

Bei der Kreuzkorrelationsfunktion hingegen werden Blöcke unterschiedlicher Bilder miteinander verglichen. Für ein breites Einsatzspektrum wurde eine große Anzahl an Varianten geschaffen.

Die Kreuzkorrelationsfunktion entsteht durch die Faltung zweier Signalfolgen. In der Signaltheorie wird dann auch von der Korrelationsenergie  $\text{xxxxxx}$  zweier Signalfolgen gesprochen.

$$KKF(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (P_r(i, j) \cdot F(\xi + i, \eta + j)) \quad (\text{B.4})$$



Ebenso wie bei der SSD und der SAD-Funktion sind die Werte weder nach unten noch oben hin begrenzt. Sie funktioniert auch bei verrauschten Signalfolgen noch sehr gut [9].

Durch Verwendung der mittelwertfreien Pixel kann eine Unabhängigkeit von einem additiven Unterschied zwischen beiden Folgen erreicht werden.

$$KKFMF(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right) \quad (\text{B.5})$$

$\overline{P_r(i, j)}$  - mittelwertfreie Pixel des Referenzblockes

$\overline{F(\xi + i, \eta + j)}$  - mittelwertfreie Pixel des Suchblockes

$m, n$  - Blockbreite

$\xi, \eta$  - Verschiebung in x, y- Richtung

Diese Funktion ist damit lokal mittelwertfrei und deshalb symmetrisch zu Null. Dies ist bereits ein entscheidender Vorteil für die Auswertung, da positive Werte eine Äquivalenz und negative Werte eine Antivalenz ausdrücken. Der Betrag der Werte ist jedoch unbegrenzt. Somit kann kein globaler Schwellwert definiert werden. Es fällt auf, dass das Produkt der beiden mittelwertfreien Signalfolgen die Kovarianz der Signalfolgen darstellt. Durch Normierung mit der Wurzel des Produktes der beiden Varianzen der Signalfolgen entsteht die normierte mittelwertfreie Kreuzkorrelationsfunktion. Bei identischen Signalfolgen ergibt sich unabhängig vom Wertebereich der Wert +1 und für entgegengesetzte Signalfolgen der Wert -1. Nimmt das Ergebnis den Wert Null an, korrelieren die zwei Signalfolgen nicht miteinander korreliert.

$$NKKFMF(\xi, \eta) = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right)}{\sqrt{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2}} \quad (\text{B.6})$$

Die NKKFMF ist lokal mittelwertfrei und wertebegrenzt. Dadurch sind eindeutige Aussagen und Vergleiche über eine Signalfolge hinaus möglich. Dies kann bei der Stereobildverarbeitung einen deutlichen Vorteil darstellen.

Durch Vereinfachung der normierten mittelwertfreien KKF entsteht das Moravec-Kriterium Gl. 2.12 Es wird auch als pseudonormierte KKF bezeichnet. Das Moravec-Kriterium hat eine leichte Sensitivität gegenüber multiplikativen Störungen. Nach Aschwanden [9] ist es sehr stark von den Aufnahmebedingungen abhängig, insbesondere ob multiplikative Helligkeitsschwankungen auftreten.

Aschwanden hat in der Dissertation verschiedene Korrelationskriterien miteinander verglichen. Er hat dafür vorwiegend synthetische Bilder verwendet, in denen er die Empfindlichkeit bezüglich unterschiedlicher Rauscharten und bezüglich additiver und multiplikativer Helligkeitsstörungen untersucht. Im Ergebnis seiner Untersuchungen kam er zu dem Schluss, dass die

NKKFMF und die NSSDMF am robustesten gegenüber allen untersuchten Störungen sind. In seiner Arbeit hat er insbesondere für die SAD und SSD als empfindlich gegenüber additiven und multiplikativen Helligkeitsstörungen festgestellt. Diese kann jedoch durch die Verwendung von lokal mittelwertfreien Blöcken weitestgehend ausgeschlossen werden.

## B.2 Rechenaufwand von Korrelationskriterien

### B.2.0.6 Nichtnormierte Funktionen

Die einfachsten Korrelationsfunktionen sind die SAD und die SSD-Funktion. Sie bestehen jeweils aus einer Differenz, die quadriert bzw. von der der Betrag gebildet wird. Diese wird dann über die gesamte Fenstergröße aufsummiert. In beiden Fällen wird somit der Fehler, der zwischen dem Referenzblock und den Kandidaten besteht, pixelweise berechnet.

$$SAD(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} |P_r(i, j) - F(\xi + i, \eta + j)| \quad (\text{B.7})$$

$$SSD(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (P_r(i, j) - F(\xi + i, \eta + j))^2 \quad (\text{B.8})$$

Welche Funktion schneller implementierbar ist, hängt im Wesentlichen von der Plattform ab. Zunächst einmal wollen wir die Implementierbarkeit in Software untersuchen. Die SAD-Funktion muss drei arithmetische Operationen je Pixel ausführen. Eine Subtraktion, Betrag und die Aufsummierung. Die SSD-Funktion im Vergleich dazu besteht mit einer Subtraktion einer Quadrierung (Multiplikation) und der Aufsummierung ebenfalls aus 3 Operationen. Je nach Prozessor kann die Geschwindigkeit jedoch bei beiden sehr stark variieren.

Verwendet man Mikroprozessoren, welche normalerweise keine hart verdrahteten Multiplizierer besitzen, so ist die SAD-Funktion schneller, da die Multiplikation der SSD durch mehrere Operationen ersetzt werden muss.

Auf einem DSP hingegen wird die SSD vorteilhafter sein. Durch ihre Struktur kann die Quadrierung und die Aufsummierung mit Hilfe der MAC-Operation in einem Takt erfolgen (siehe Abb. B.2). Somit kann die Berechnung idealerweise in 2 Takten je Pixel umgesetzt werden. Die SAD benötigt auch hier 3 Takte. Wenn man dann eine Fenstergröße von 16 Pixel zu Grunde legt, dann braucht die Berechnung eines Kandidaten 32 Takte bei der SSD und 48 Takte bei der SAD-Funktion. Dementsprechend würde für die Berechnung eines Suchbereiches von 16 Pixeln, bei dem in dieser Arbeit verwendeten Versuchsaufbau, ein DSP mit einem Takt von 889 MHz für die Berechnung der SSD und 1,33 GHz für die SAD benötigt werden. Hierbei würden 64 Blöcke in einem Bereich von 16 Pixeln verschoben werden. Dies sind somit 1024 verschiedene Positionen.

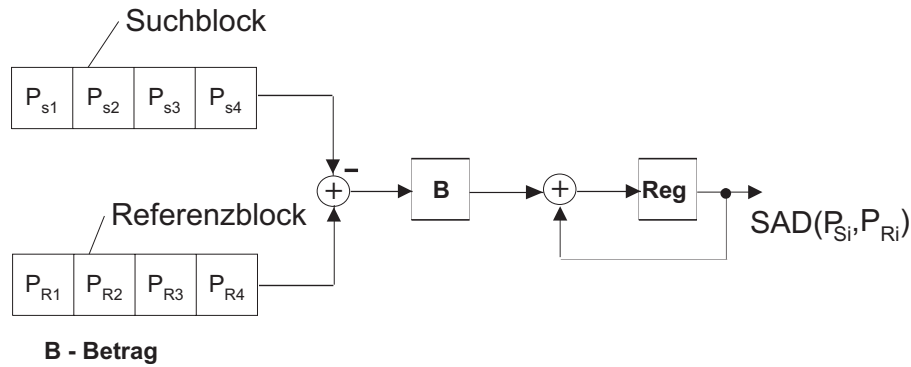


Abbildung B.1: Struktur einer SAD-Funktion mit sequentieller Abarbeitung

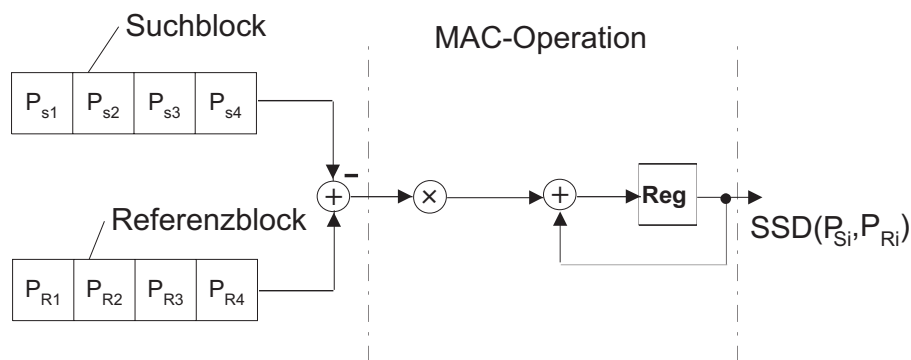


Abbildung B.2: Struktur einer SSD-Funktion mit sequentieller Abarbeitung

Bei Verwendung des hierarchischen Verfahrens könnte bei Berechnung jeder zweiten Zeile somit eine Berechnung in Echtzeit erreicht werden. Allerdings wurde hierbei angenommen, dass sich die Daten bereits in der benötigten Form im Hauptspeicher befinden. Da es bei DSPs durchaus möglich ist, dass ein DMA-Kontroller direkt in den Speicher auf dem Chip schreibt, könnte dies prinzipiell realisiert werden. Allerdings sind damit weder die Ebenen aus dem Bild berechnet noch die jeweiligen Maxima ermittelt.

Die Verwendung eines Dualcore-DSPs, wie z.B. den Blackfin 561 von Analog Devices, würde diese Aufgabe bewältigen können. Allerdings sind diese nur mit einer maximalen Taktfrequenz von 600 MHz verfügbar.

Eine Diskussion bezüglich der Implementierbarkeit in PC-Prozessoren ist nur schwer möglich, da die konkreten Strukturen nicht direkt bekannt sind und die Geschwindigkeiten bei den unterschiedlichen Architekturen zu stark variieren. Dies würde den Rahmen der Arbeit sprengen.

Eine Implementierung in Hardware dieser beiden Kriterien ist mit verhältnismäßig geringem Aufwand möglich. Die einfachste Variante ist eine an Software angelehnte Version wie sie in Abb. B.1 für die SAD-Funktion und in Abb. B.2 für die SSD Funktion dargestellt ist. Nachteil dieser Version ist, dass sie mit einer sehr hohen Frequenz laufen muss. Da jedoch alle Elemente parallel vorhanden sind, muss sie in beiden Fällen nur mit einfacher Geschwindigkeit abgearbeitet werden. Bei dem obigen Beispiel müsste der Takt 445 MHz betragen. Weiterhin könnten

dann mehrere dieser Elemente parallel geschaltet werden, um die verbleibenden Ebenen des hierarchischen Ansatzes zu berechnen. Die bei Altera verfügbaren DSP-Blöcke wären für die Implementierung der SSD-Funktion ideal, da sie eine MAC-Operation durchführen. Die Subtraktion könnte vorher mit Hilfe regulärer Logikzellen implementiert werden. Diese können in den schnellsten Ausführungen der FPGAs jedoch maximal eine Taktfrequenz von 300 MHz erreichen. Wenn sich jeweils zwei solche Elemente eine Zeile teilen, würde ein Takt von 225 MHz ausreichend sein. Damit bräuchte man 10 solcher Korrelatoren. Die Auswertelogik kann in den durch die Ausnutzung der DSP-Elemente noch sehr leeren FPGA implementiert werden. Ein Problem bei dieser Variante stellt die Belieferung der Korrelatoren mit Daten dar.

Unter Verwendung der DSP-Blöcke würden für jeden Korrelator zusätzlich 16 Register mit 10 Bit Breite (bei einer Implementierung für 10 Bit Kamerasignale) für den Referenzblock und 11 Logikelemente für den Addierer entstehen. Da für die SAD-Funktion keine DSP-Blöcke genutzt werden können, würde sie auch hier Logikelemente benötigen. In diesem konkreten Beispiel werden 10 LE für Negierer und 10 weitere für einen MUX der die jeweils positive Zahl auswählt. Weiterhin 16 LE für den Akkumulator, da er ohne Verluste arbeiten soll. Insgesamt braucht die SAD-Funktion ohne Datenzuführung damit 47 LE und weitere 160 LE für den gespeicherten Referenzblock. Um den Aufwand vergleichen zu können, ist jedoch eine Implementierung der SSD ohne DSP-Blöcke nötig, wobei dann 155 LE benötigt werden würden.

Für diese Implementierung sowie die Softwareimplementierungen gilt, dass mit größerer Blockgröße oder größerem Suchfenster (nur Software) die Durchlaufzeit oder die Taktrate erhöht werden muss. Die Hardware kann größere Suchfenster durch Mehrfachinstanzierungen abfedern.

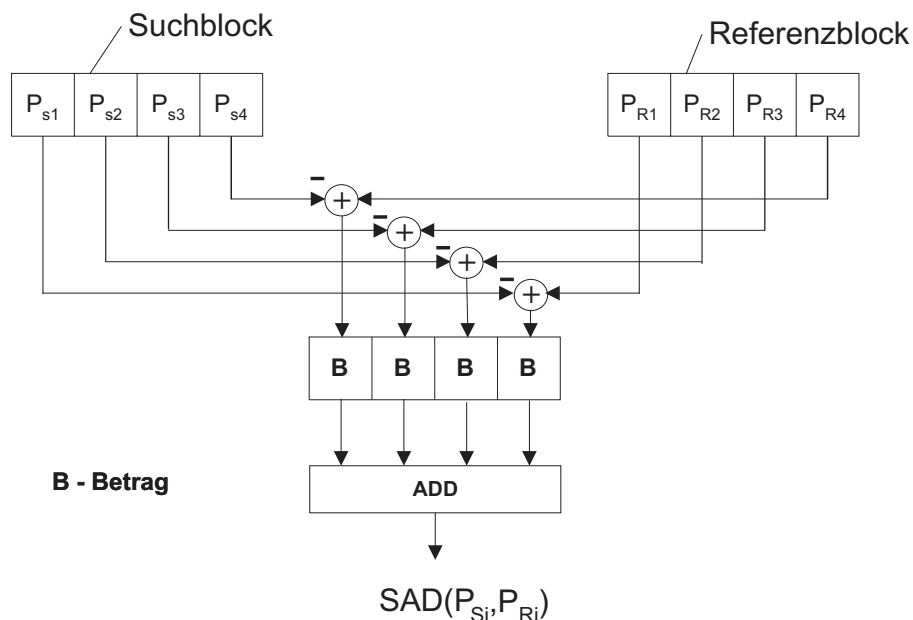


Abbildung B.3: Struktur einer SAD-Funktion mit paralleler Abarbeitung

Ein anderer Ansatz stellt eine parallele Struktur dar. Dabei werden alle Operationen für jedes Wertepaar parallel implementiert. Dies ist in Abb. B.3 für die SAD und in B.4 für die

SSD dargestellt. Der vollständige Durchlauf dauert hierbei länger, aber durch den Einsatz von massivem Pipelining kann die Taktrate niedrig bleiben und der Datendurchsatz stark erhöht werden. So kann in jeden Takt eine Blockkombination berechnet werden. Der dafür benötigte Hardwareaufwand ist dem entsprechend größer.

	SAD	SSD ohne DSP	SSD mit DSP
Stufe 1	176 LE	176 LE	176LE
Stufe 2	320 LE	1936 LE	16 DSP-Bl.
Stufe 3	240 LE	345 LE	345LE
Summe:	736 LE	2457 LE	521LE

Tabelle B.1: Vergleich der Ressourcen von SAD und SSD

Der Vorteil dieser Implementierungsform ist die niedrige Geschwindigkeit. Für einen Bilddatenstrom mit 28 MHz Pixeltakt kann die Hardware mit 28 MHz laufen um eine Suchweite von 16 zu realisieren. Durch Implementierung eines zweiten Korrelators könnten zeitgleich die anderen Ebenen berechnet werden. Und somit eine Suchweite von 256 Pixeln realisiert werden.

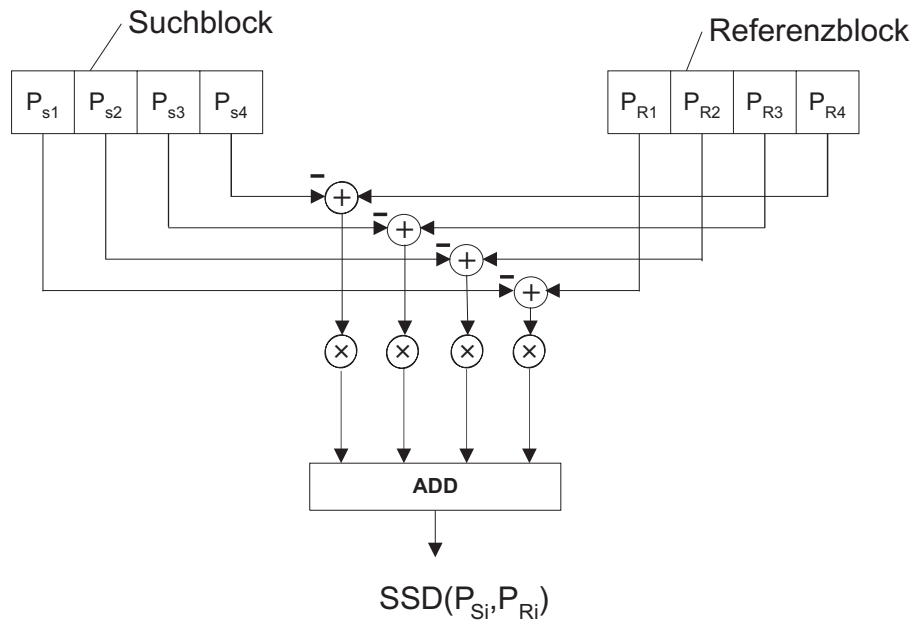


Abbildung B.4: Struktur einer SSD-Funktion mit paralleler Abarbeitung

Die nächste Gruppe von Funktionen sind die SAD-Funktion und die SSD-Funktion in ihrer mittelwertfreien Form. Hierfür muss vor der Berechnung der Funktion der Mittelwert jedes Blockes berechnet werden und dann dieser von allen Elementen des Blockes abgezogen werden. In Hardware ist das sowohl für die sequentiellen wie für die parallelen Implementierungsvarianten mit einfachen Mitteln möglich.

$$SADM F(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} |P_r(i, j) - F(\xi + i, \eta + j)| \quad (\text{B.9})$$

$$SSDMF(\xi, \eta) = \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} (P_r(i, j) - F(\xi + i, \eta + j))^2 \quad (\text{B.10})$$

In Abb. B.5 ist eine Schaltung für einen gleitenden Mittelwert dargestellt. Hierbei werden für jeden neuen Block  $n+2$  Additionsoperationen benötigt, wobei  $n$  die Blockbreite ist. Da der Referenzblock jeweils für die Zeit des Vergleiches konstant bleibt können  $(n-1)$ \*Suchweite Additionen eingespart werden. Der DSP bräuchte somit für jeden Suchschritt 18 Additionen und zum Start ebenfalls einmal 18 Additionen zusätzlich für jede der beiden Funktionen. Damit ist der Unterschied zwischen SSDMF(20) und SADMf(21) nicht mehr wahrzunehmen. Der Aufwand um eine Echtzeitrealisierung zu erreichen, ist damit ca. um den Faktor 10 gestiegen.

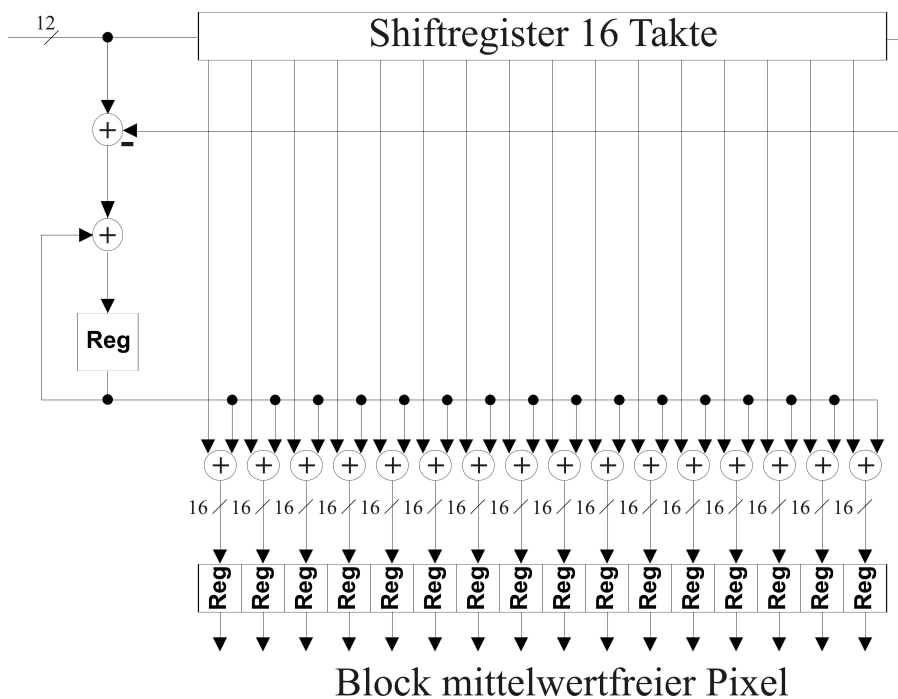


Abbildung B.5: Realisierung des gleitenden Mittelwertes in Hardware

Bei der Hardwareimplementierung hingegen steigt der Aufwand nicht so imens. Die  $2 \times 18$  Additionen benötigen bei einer Bitbreite von 10 Bit 768 Logikeinheiten. Bei der SADMf und der SSD mit DSP-Blöcken steigt der Bedarf an Hardware somit um ca. 100%. Bei der SSDMF ohne DSP-Blöcke hingegen steigt der Bedarf nur um ca. 30%. Dabei bleibt das Design weiterhin verhältnismäßig unabhängig von der Geschwindigkeit.

### B.2.0.7 Normierte Funktionen

Das bekannteste Beispiel für die normierten Korrelationsfunktionen ist die normierte mittelwertfreie Kreuzkorrelationsfunktion (KKFMf; siehe B.11). Bei ihr werden die Elemente zweier mittelwertfreier Pixelfolgen miteinander multipliziert und dann durch Division durch das Produkt der Varianzen der beiden Wertefolgen normiert. Jedoch sind die Anforderungen an die

Rechenleistung sehr hoch (siehe Abb. B.6. Es müssen zum einen die mittelwertfreien Pixel berechnet werden. Dies hat bei den unnormierten Korrelationsfunktionen bereits zu einem starken Anstieg der erforderlichen Rechenleistung geführt. Das Produkt und die Aufsummierung im Zähler könnte leicht durch eine MAC-Operation durchgeführt werden. Ebenso kann diese bei der Berechnung der Varianzen kombiniert werden.

$$KKFMF(\xi, \eta) = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right)}{\sqrt{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2}} \quad (\text{B.11})$$

Allerdings bleiben sowohl die Wurzel als auch die Division sehr unbequeme arithmetische Operationen. Die Division wird iterativ wie bei schriftlicher Division mit Rest oder durch ein Annäherungsverfahren ermittelt. In beiden Fällen ist die genaue Länge der Operation nicht bestimmbar. Bei dem Verfahren der schriftlichen Division mit Rest ist jedoch die maximale Anzahl der Schritte endlich und direkt von der Bitbreite des Zählers abhängig. Aus diesem Grund wird beim Aufbau von Hardwaredividierern eher diese Variante verwendet.

**Ersetzen der Division** In Software und in Hardware kann die Division durch zwei Multiplikationen ersetzt werden sofern nicht direkt das Ergebnis weiterverwendet werden. Ist das Ergebnis eines Vergleich zweier Brüche gesucht und sind die konkreten Zahlenwerte nicht relevant, können die Divisionen durch entsprechende Multiplikation mit dem Nenner ersetzt werden. Dies soll in folgendem Beispiel mit zwei gegebenen Brüchen  $\frac{A}{B}$  und  $\frac{C}{D}$  verdeutlicht werden. Somit wird

$$\frac{A}{B} \leq \frac{C}{D} \quad (\text{B.12})$$

unter Beachtung des Vorzeichens durch

$$A \cdot D \leq C \cdot B \quad (\text{B.13})$$

ersetzt. Ob hierbei eine Reduzierung des Aufwandes liegt kommt auf den konkreten Prozessoren an. Viel Mikroprozessoren die in PC eingesetzt werden besitzen eine für Divisionen optimierte Pipeline.

**Wurzel Implementierung** Ein weiteres großes Problem ist die Wurzel im Nenner der normierten Kreuzkorrelationsfunktion. Die Wurzel kann mit Hilfe der digitaltechnik nicht exakt berechnet werden. Die Problematik ergibt sich durch die Echtzeitfähigkeit der Ermittlung der Ergebnisse und des großen Wertebereichs des Radikanten. Zur Lösung des Problems ergeben sich folgende Möglichkeiten:

## ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN

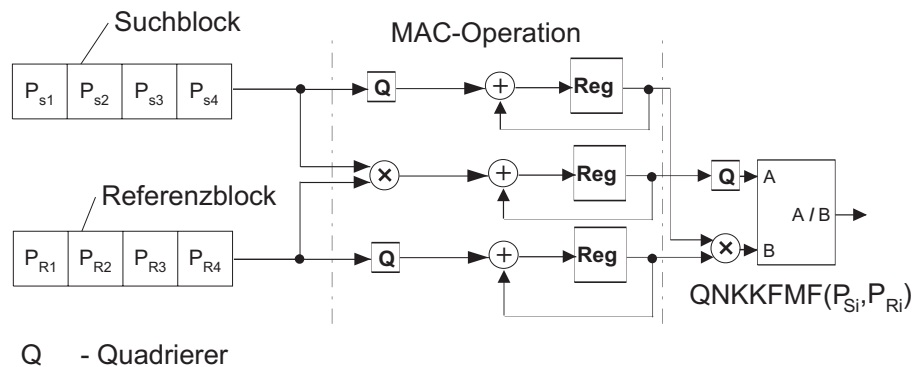


Abbildung B.6: Struktur der sequentiellen Implementierung der QNKKFMF

- Iteration
- Näherungsfunktion
- Look Up Table
- Quadrierung der gesamten Funktion

Aufgrund der Tatsache, dass der Wertebereich des Radikanten über einen sehr großen Bereich variiert, wurde die Quadrierung der Kreuzkorrelationsfunktion als beste Möglichkeit ausgewählt.

Die Iteration entfällt aufgrund des für die Echtzeitforderungen zu hohen Zeitbedarfs. Das Problem der Näherungsfunktion ist die begrenzte Gültigkeit ihrer Ergebnisse. Eine Näherungsfunktion, die den oben beschriebenen Wertebereich ausreichend gut abdeckt, z.B. als Polynom würde einen zu großen Rechenaufwand erzeugen. Die dritte Möglichkeit entfällt aufgrund der großen Bitbreite/Genauigkeit des Radikanten mit bis zu 72 Stellen. Eine solche große LUT würde einen extrem großen und schnellen Speicher voraussetzen. Bei einer Verkleinerung der LUT unter Verwendung von Interpolationen würden immer noch mehrere Megabyte Speicherplatz benötigt. Bei einer Reduzierung der Größe der LUT auf ein akzeptables Maß würde die Ungenauigkeit zu groß werden. Die Ergebnisse der sich aus der Quadrierung der KKFMF ergebenden Funktion liegen zwischen 0 und +1. Da die Ergebnisse der KKF nicht direkt interessieren, sondern nur das Maximum, ist es unproblematisch, die Funktion zu quadrieren. Durch die Quadrierung kann ein Minimum im negativen Zahlenbereich ein globales Maximum werden. Um dies zu verhindern, werden nur positive Werte berechnet. Die negativen Werte werden nicht benötigt, da nur das absolute Maximum über einer bestimmten Schwelle gesucht wird. Zur Ermittlung des Maximums werden zunächst alle Werte ausgeschlossen, die kleiner als 0,5 sind.

Die Quadrierung hat zu dem den Vorteil, dass von den zwei Operationen, die im Nenner mehr durchgeführt werden müssen, eine in den Zähler wandert. Die Wurzel entfällt im Nenner und dafür muss der Zähler quadriert werden. Dies wirkt sich abgesehen vom sehr viel geringeren Aufwand einer Multiplikation gegenüber der Ermittlung der Wurzel in Software nicht weiter aus. Bei einer Hardwareimplementierung hat dies jedoch einen zusätzlichen positiven Effekt.



Für eine Softwareimplementierung auf einem DSP ergeben sich somit  $3 \cdot n$  MAC-Operationen zusätzlich zwei größere Multiplikationen. Die Division sollte später während des Vergleiches durch zwei Multiplikationen (siehe [B.2.0.7](#)) ersetzt werden. Dabei werden die beiden Quotienten die mit einander verglichen werden ausmultipliziert und anschließend verglichen. Der Vorteil in der Ausführungszeit ist spürbar. Leider ist dieses Verfahren durch den verfügbaren Speicherplatz beschränkt. Es müssen für jeden Wert im Suchbereich 2 Werte statt einem abgespeichert werden.

Dies ist innerhalb eines DSP dementsprechend nur mit kleineren Suchbereichen möglich. Bei dieser Variante wären  $(1+3) \cdot n + 6$  Rechenoperationen die jeweils 1 Takt benötigen bei einer idealen Implementierung in einem DSP erforderlich. Bei einer Fensterbreite von 16 Pixeln wären somit 288 Rechenoperationen für eine Blockkombination erforderlich. Für das Beispiel der unnormierten Funktionen mit 16 Pixel Suchbereich wären somit 4624 Takte (4608 für die  $288 \times 16$  und 18 für den mittelwertfreien Referenzblock) je Referenzblock und somit 294912 Rechenoperationen je Zeile notwendig. Für eine Realisierung in einem normalen Mikroprozessor wären erheblich mehr Takte nötig, da die Zwischenergebnisse oft noch einen Takt zum wegschreiben benötigen. Um mit den gängigen DSP ein echtzeitfähiges Design zu realisieren, müsste bei einer Pixelrate von 28 MHz die Verarbeitung Takt von mindestens 8.001 GHz betragen. Der Rechenaufwand gegenüber der SS-Funktion ist um den Faktor höher.

Implementiert man die sequentielle Version der KKFMMF in Hardware so benötigt man neben der Schaltung für die mittelwertfreien Blöcke zusätzlich 3 DSP-Blöcke die jeweils den Zähler und die zwei Terme im Nenner berechnen. Weiterhin werden zwei Multiplikationen mit hoher Genauigkeit benötigt und der Dividierer. Stehen keine DSP-Blöcke zur Verfügung müssen die DSP-Blöcke jeweils wieder durch eine Schaltung mit einem Multiplizierer und einem Addierer implementieren. Hierbei entsteht ein Aufwand von  $2 \cdot 768$  LE für die mittelwertfreie Implementierung, 3 DSP-Blöcke oder 747 LE. Anschließend sind zwei Multiplizierer mit einer Bitbreite von 28 Bit erforderlich. Und zum Schluss ein Dividierer, bei dem 32 durch 16 Bit dividiert werden. Um einen solch kleinen Dividierer einsetzen zu können, muss ein Blockfließkommadividierer eingesetzt werden. Hierbei wird die führende Null des Nenners gesucht. Da das Ergebnis immer kleiner gleich 1 ist, liegt die höchstwertige "1" immer im Zähler immer an der gleichen oder einer niedrigeren Stelle wie der Nenner. Anschließend kann das Ergebnis mit einem sehr viel kleineren Dividierer durchgeführt werden. Es müssen für eine Realisierung mit DSP-Blöcken 5630 LE und für die ohne DSP-Blöcke 6377 LE zur Verfügung stehen. Dies stellt bei der aktuellen Größe von FPGAs kein Problem dar. Die zu realisierende Geschwindigkeit bei den größeren Multiplizierern und beim Dividierer ist äquivalent mit dem Pixeltakt. Problematisch ist hier ähnlich wie bereits bei der SSD Funktion eine Taktfrequenz von mindestens 445 MHz. Welche so nicht realistisch ist. Wie jedoch bereits vorher beschrieben, könnte eine Realisierung durch Mehrfachinstanziierung umgesetzt werden. Bei Einsatz von 4 Instanzen würde sich der Hardwareverbrauch auf  $22520 \text{ LE} + 12 \text{ DSP-Blöcke}$  bzw. auf 25508 LE erhöhen.

Bei der parallelen Implementierung werden, wie in [Abb. B.7](#) dargestellt, alle Elemente der Berechnung parallel und mit Pipelining ausgelegt. Somit sinkt die nötige Geschwindigkeit auf die einfache Pixelfrequenz. Problematisch ist hierbei jedoch, dass in der Stufe 1 nach den Multiplizierern alle Bits gleichzeitig schalten müssen. Dies sind 1152 Bits. Diese Stelle beschränkt

ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN

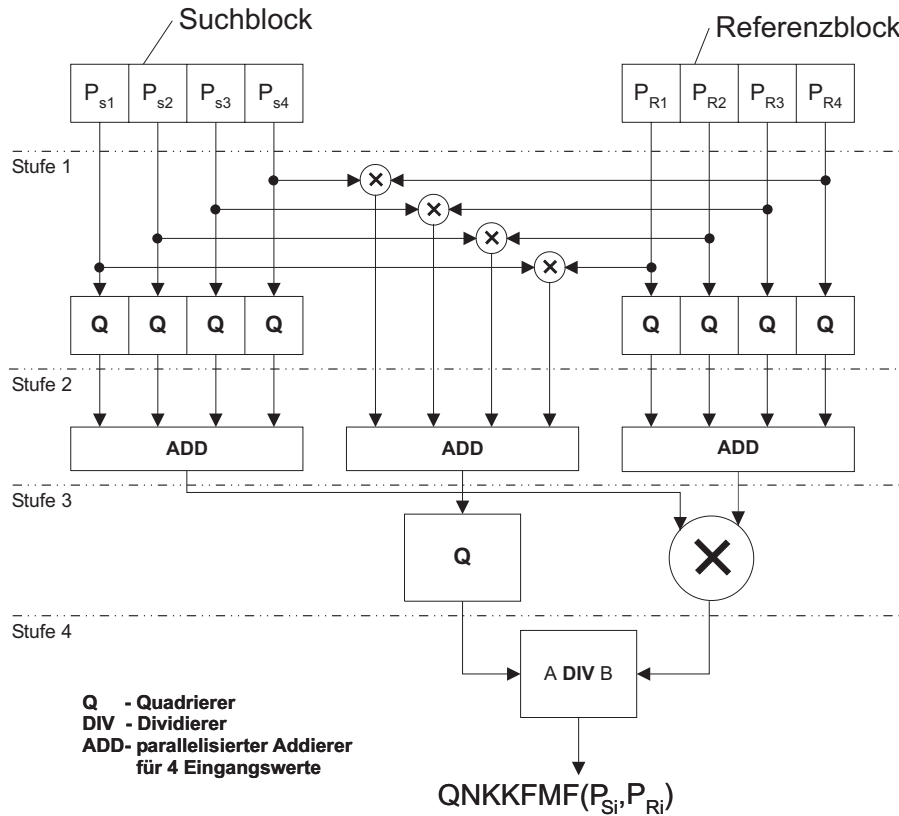


Abbildung B.7: Struktur der parallelen Implementierung der QNKKFMF

damit auch die maximale Geschwindigkeit. Die Gesamtgröße liegt bei 20500 LE und ist somit unter den der Mehrfachinstanzierungen der sequentiellen Schaltung.

Die NSSDMF Gl.B.14 ist im Vergleich zur KKFMMF von der Implementierung her geringfügig ungünstiger. Im Prinzip müssen die gleichen arithmetischen Operationen durchgeführt werden, jedoch erfolgt zusätzlich im Zähler eine Subtraktion zur Bildung der Differenz. Dies bedeutet, dass für jeden Pixel in der Fenstergröße ein Takt mehr erforderlich ist. Das bedeutet, dass eine Blockkombination 86 Takte braucht. Das entspricht in einem DSP einem um 22% höheren Rechenaufwand. In Hardware ist der Aufwandsunterschied aufgrund der sehr kleinen Addierer wesentlich geringer. So ist er bei der sequentiellen Implementierung nur um 0,26% höher und bei der parallelen um 1,3 %.

$$NSSDMF(\xi, \eta) = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right)}{\sqrt{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2}} \quad (B.14)$$

Das Moravec-Kriterium hingegen ist im Rechenaufwand zur KKFMMF geringer. Die Vorteile liegen in der geringeren Bitbreite der Operationen. Da diese Operationen jedoch nicht so häufig ausgeführt werden, fällt dies nicht so stark ins Gewicht. Der wesentliche Rechenaufwand entsteht

durch die drei MAC-Operationen am Anfang. ...

$$MOR(\xi, \eta) = \frac{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \left( \overline{P_r(i, j)} \cdot \overline{F(\xi + i, \eta + j)} \right)}{\sqrt{\sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{P_r(i, j)}^2 \cdot \sum_{j=0}^{n-1} \sum_{i=0}^{m-1} \overline{F(\xi + i, \eta + j)}^2}} \quad (\text{B.15})$$

### B.2.0.8 Vorverarbeitung von Bildern

Ein weiteres gängiges Prinzip bei der Auswertung von Stereobildern ist die Vorverarbeitung von Bildern, um bestimmte Merkmale hervorzuheben und Störungen zu unterdrücken. Zur Unterdrückung von Störungen werden auf das Problem angewendete Glättungs und Median-Filter verwendet.

Beim Hervorheben von Kanten und Mustern in Bildern zum Vergleich der Bilder mit Hilfe von Korrelationskriterien muss beachtet werden, dass noch ihre Eigenschaften z.B. der Grauwertgradient erhalten bleibt. Um Gleichzeitig eine Glättungsoperation und die Hervorhebung von Grauwertgradienten durchzuführen wurde der *Marr-Hildreth-Operator* entwickelt. Dabei wird auf einen 2D-Glättungsfilter mit Gauss-Verteilung ein Laplace-Filter angewendet, daher wird diese Operation auch oft Laplacian of Gaussian (LoG) genannt. Es entsteht die 2. Ableitung der Gauss-Verteilung.

Durch die Funktion

$$\Delta g(x, y) = -\frac{1}{\pi\sigma^4} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \left( 1 - \frac{x^2 + y^2}{2\sigma^2} \right) \quad (\text{B.16})$$

wird der Koeffizient in Abhängigkeit von der Position in der Filtermatrix und damit der Filterkern beschrieben. Die Standardabweichung  $\sigma$  gibt die Breite der Gaussverteilung und damit auch die Breite des Filterkerns an. Für ein  $\sigma$  von 1,0 und eine Filtergröße von 5x5 Pixel ergibt sich zum Beispiel folgender Filterkern

$$g(x, y) = \begin{bmatrix} 1 & 2 & 2 & 2 & 1 \\ 2 & 0 & -5 & 0 & 2 \\ 2 & -5 & -16 & -5 & 2 \\ 2 & 0 & -5 & 0 & 2 \\ 1 & 2 & 2 & 2 & 1 \end{bmatrix} \quad (\text{B.17})$$



(a) *linkes Bild*



(b) *rechtes Bild*

Abbildung B.8: Synthetisches Bildpaar (IM26) mit einer Sinushalbwelle mit einer Breite von 20 Pixeln. Im linken Bild befindet sich das Signal in allen Zeilen an der gleichen Stelle und im rechten Bild wird das Signal mit jeder Zeile um einen achtel Pixel verschoben, damit ergibt sich über 1024 Zeilen ein Disparitätsbereich von 0-128 Pixel

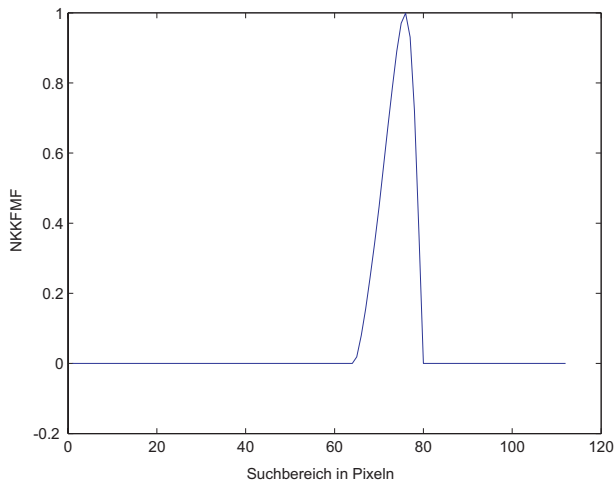
### B.3 Eigenschaften verschiedener Korrelationskriterien und Untersuchung der Subpixelgenauigkeit

Die Blockgröße während der Untersuchung betrug 16x1 Pixel. Um das Verhalten der Funktion an einem einfachen Beispiel zu betrachten, wurde synthetische Testbilder erstellt. Die ersten beiden Stereobildpaare zeigen die obere Hälfte eines Sinusverlaufs ohne Periodizität. Dabei steigt die Disparität mit jeder Zeile um  $1/8$  Pixel an. Das erste Stereobildpaar (Abb. B.8) zeigt einen schmalen Sinusverlauf, dessen Breite im Bereich der Blockbreite liegt. Das Sinussignal wurde mit der 8-fachen Genauigkeit erstellt und anschließend entsprechend der Verschiebung um ein achtel Pixel abgetastet. Dadurch ergeben sich nahezu identische Verläufe, jedoch nur alle 8 Zeilen absolut identische Werte.

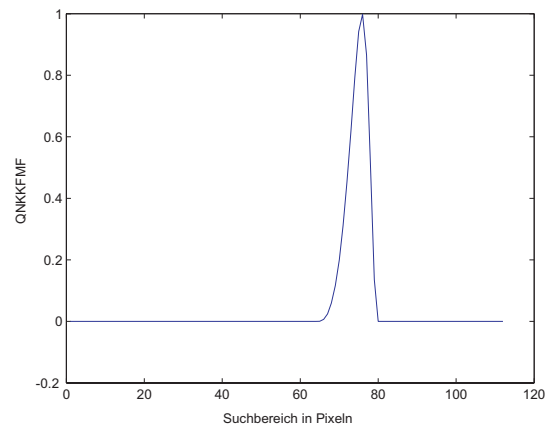
Auf dieses Stereobildpaar wurden die Kriterien NKKFMF, QNKKFMF, SAD, SADMF, SSD, SSDMF sowie die SSD und SAD auf die mit dem Marr-Hildreth-Operator gefilterten Stereobilder angewendet.

Die Grafiken B.9 zeigen alle den Korrelationsverlauf des Blockes 10 aus der Zeile 600. Die Kurvenverläufe der NKKFMF und der QNKKFMF zeigen ein einzelnes schmales Maximum welches auch gut für eine Subpixelinterpretation verwendbar ist. Durch die Quadrierung läuft das Maximum der QNKKFMF geringfügig steiler zu als das der NKKFMF. Die Verläufe der SSD und der SAD-Funktion zeigen ebenfalls deutliche Minima. Bei der SAD und der SSD hängt das Ergebnis zwar von beiden Bildern ab, jedoch kann ein starker Grauwertgradient in Suchbild allein die Ausbildung eines Nebenextremums bewirken. Insbesondere die Funktionsverläufe der mit dem mit dem Marr-Hildreth-Operator gefilterten Bilder reagieren sehr stark auf jede

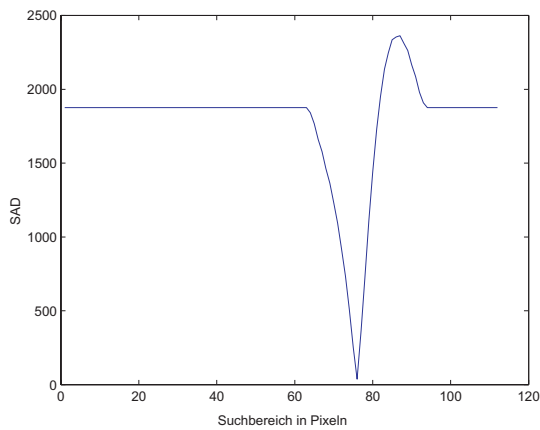
## ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN



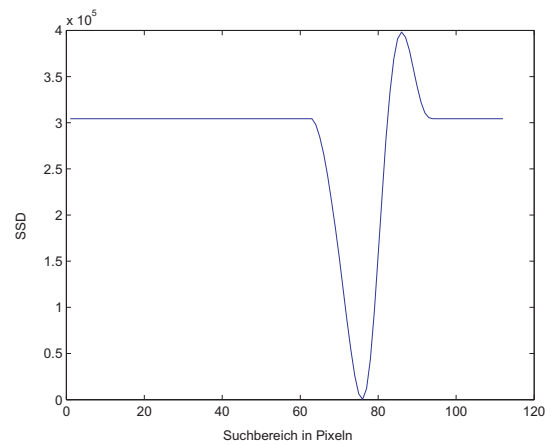
**(a) *NKKFMF***



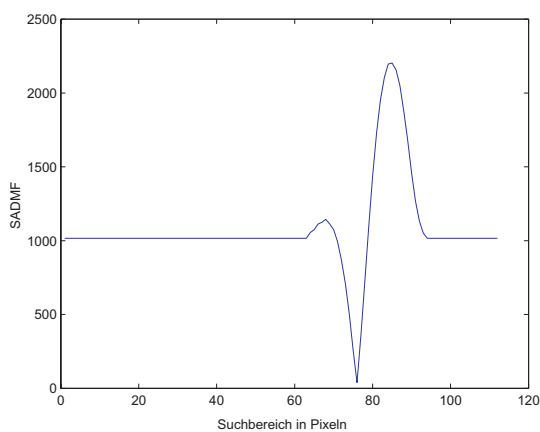
**(b) *QNKKFMF***



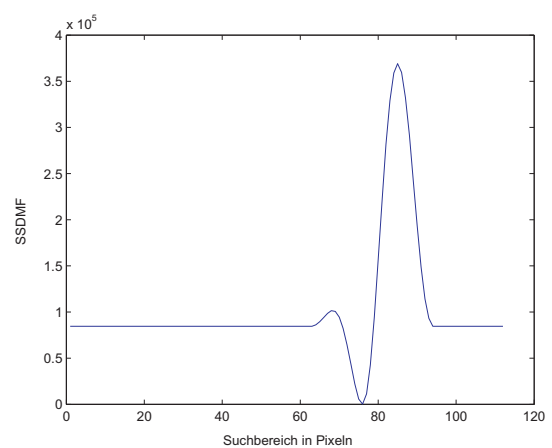
**(c) *SAD***



**(d) *SSD***



**(e) *SADMF***



**(f) *SSDMF***

Abbildung B.9: Korrelationsfunktion der NKKFMF, QNKKFMF, SAD, SSD sowie der SADMF und SSDMF angewendet auf Zeile 600 des Bildes IM26; erwartete Disparität 75

## ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN

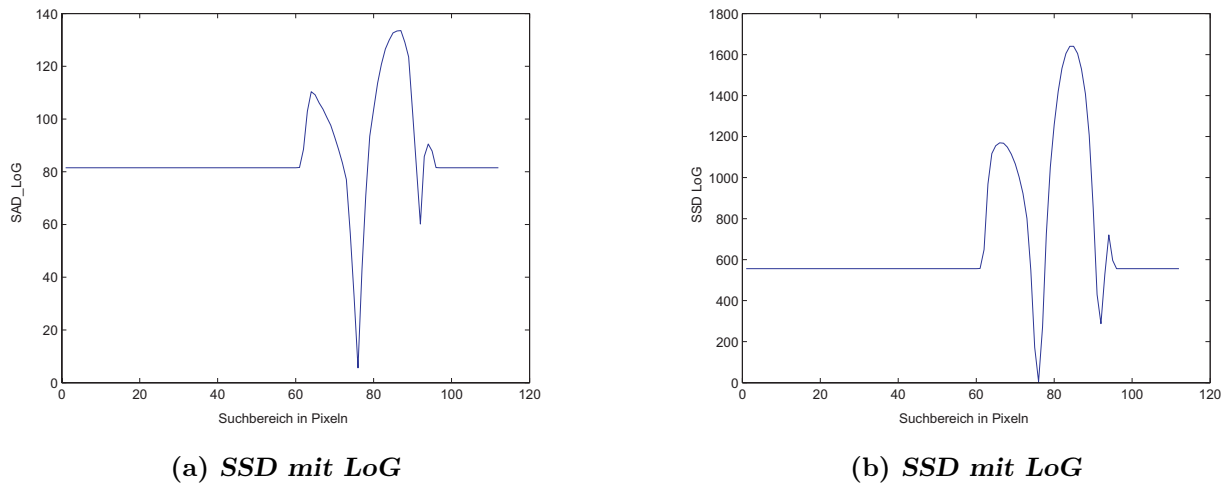


Abbildung B.10: Anwendung der SAD und der SSD-Funktion auf das mit dem Marr-Hildreth-Operator gefilterte Bildpaar IM26 (Zeile 600)

Diskontinuität im Grauwertverlauf.

Im Vergleich der SAD mit SSD Funktion erzeugt die SAD-Funktion ein deutlich schmaleres klares Minimum.

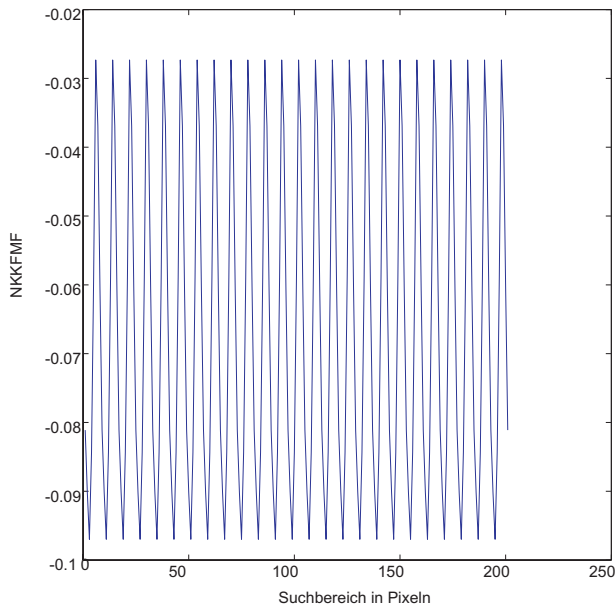
Die Verläufe in den Abbildungen B.11 zeigen den Fehler der subpixelgenauen Disparität über die ersten 200 Zeilen an. Hierbei wird ein Disparitätsbereich von 0-25 Pixel durchschritten. Der Fehler der NKKFMF ist sehr ähnlich zu dem der QNKKFMF und liegt im Mittel bei -0,06 Pixel mit einer Amplitude von 0,03 Pixeln. Der Verlauf für die SAD-Funktion liegt im Mittel bei -0,03 und hat eine Amplitude von 0,08 Pixel.

Das zweite Bildpaar (Abb. B.12) zeigt einen Sinusverlauf mit der mehrfachen Breite der Referenz und Suchblöcke. Durch den wesentlich geringeren Anstieg bilden sich wesentlich schwerer eindeutige Extrema aus.

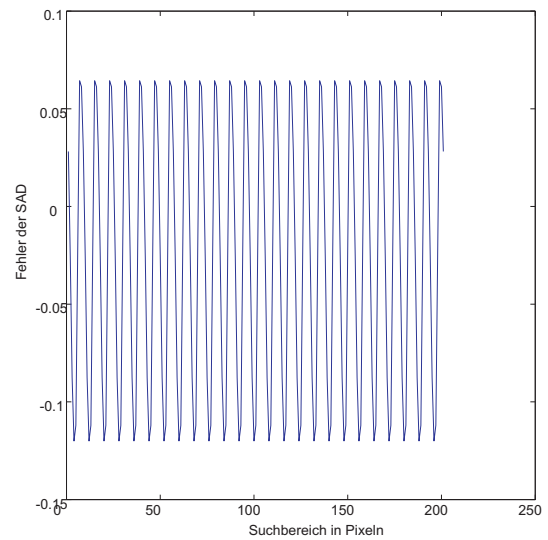
Bei einem flachen Verlauf haben alle mittelwertfreien Kriterien einen deutlich größeren Fehler. Einzig die SAD-Funktion hat ein deutliches und spitz zulaufendes Minimum, unter Berücksichtigung nahezu identischer Verläufe. Die Verläufe der SSD und der SADMF sind sehr breit und führen dann zu einem eindeutigen absoluten Minimum. Die SSDMF bildet kein eindeutiges Minimum aus. Die SAD- und SSD-Werte der gefilterten Bilder haben kein eindeutiges Minimum, dies ist dem flachen Sinusverlauf zuzuschreiben. Durch diesen bilden sich keine eindeutigen Signale für Kanten aus. Würde der Filter dem Signal angepasst werden, könnte voraussichtlich ein besseres Ergebnis erzielt werden.

Die Verläufe der NKKFMF und der QNKKFMF zeigen ein sehr breit zulaufendes Maximum. Es hat jedoch ein absolutes Maximum, welches gut im Bereich wahren Disparität liegt. Im Verlauf der Subpixelgenauigkeit für die KKFMF (Abb. B.15) zeigt sich ein bei 0,3 Pixel zentrierter periodischer Verlauf mit einer Amplitude von 1,7 Pixeln. Der Verlauf des Subpixelfehlers der SAD hingegen ist aufgrund des eindeutigen Minimas deutlich besser als der Verlauf der NKKFMF.

## ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN



(a) Fehler der KKFMM



(b) Fehler der SAD

Abbildung B.11: Vergleich der Fehler der subpixelgenauen Disparität der KKFMM und der SAD bei der Auswertung des Bildpaares IM26 über den gesamten Disparitätsbereich von 128 Pixeln



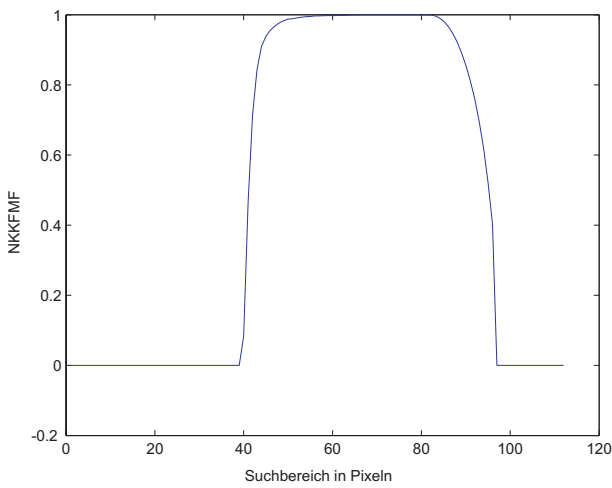
(a) Linkes Bild



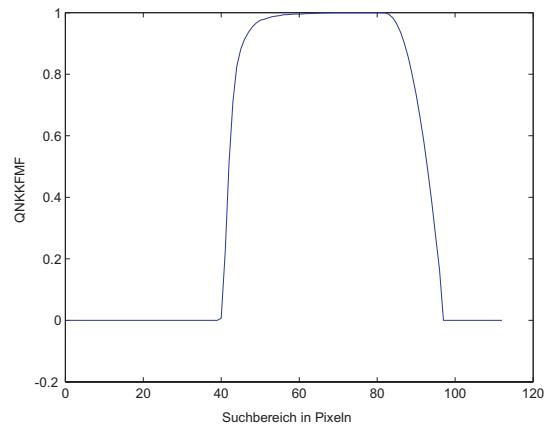
(b) Rechtes Bild

Abbildung B.12: Synthetisches Bildpaar (IM23) mit einer Sinushalbwellen mit einer Breite von 100 Pixeln. Im linken Bild befindet sich das Signal in allen Zeilen an der gleichen Stelle und im rechten Bild wird das Signal mit jeder Zeile um einen achtel Pixel verschoben, damit ergibt sich über 1024 Zeilen ein Disparitätsbereich von 0-128 Pixel

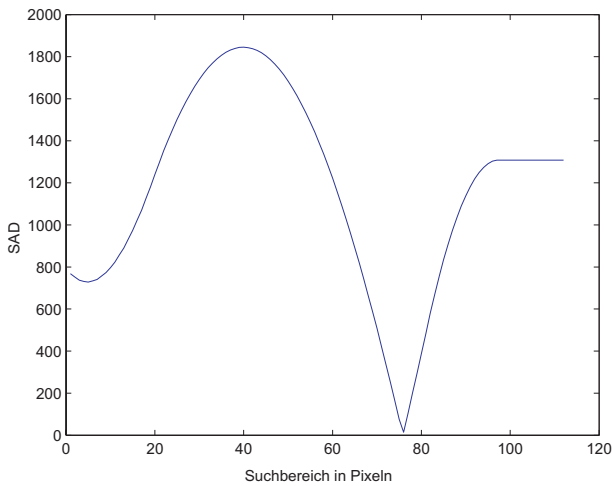
ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN



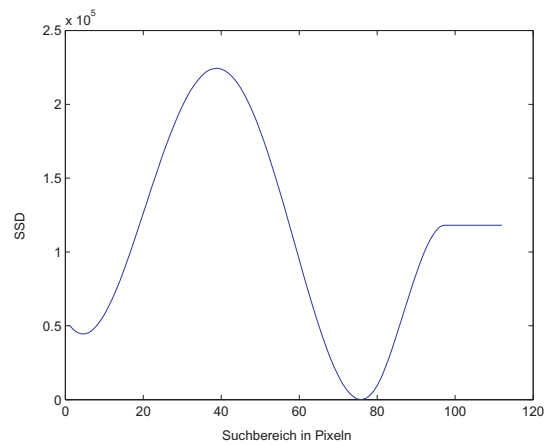
(a) *NKKFMF*



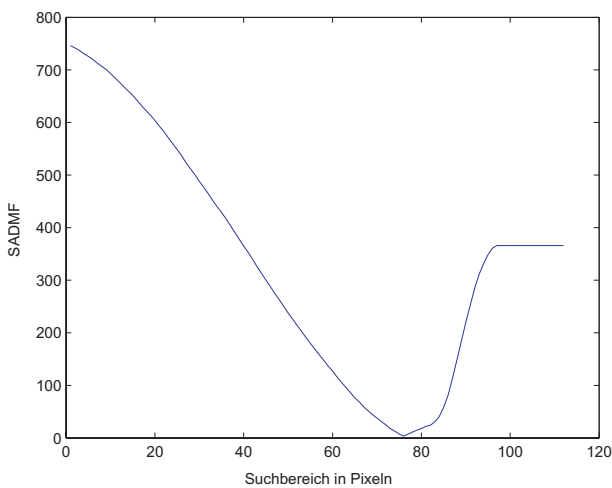
(b) *QNKKFMF*



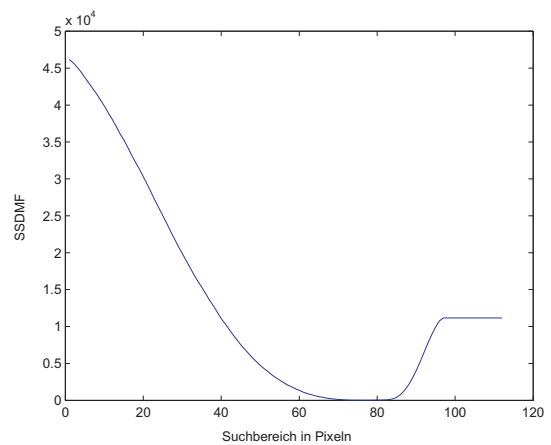
(c) *SAD*



(d) *SSD*



(e) *SADMF*

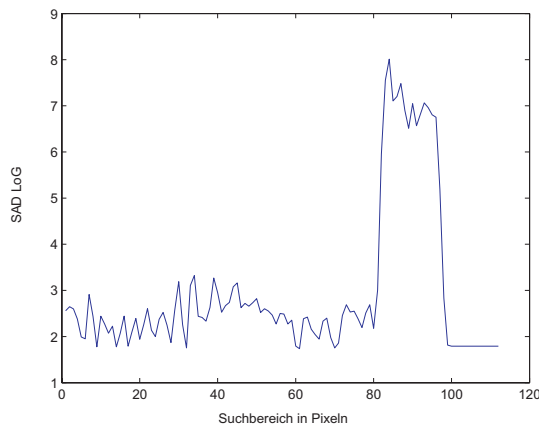


(f) *SSDMF*

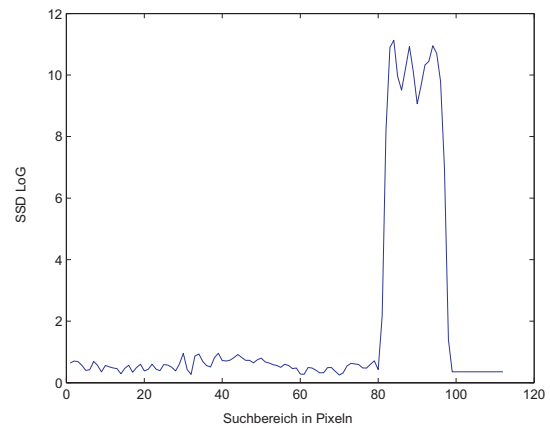
Abbildung B.13: Korrelationsfunktion der NKKFMF, QNKKFMF, SAD, SSD sowie der SADMF und SSDMF angewendet auf Zeile 600 des Bildes IM23; erwartete Disparität 75 Pixel



## ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN



(a) *SAD auf LoG*

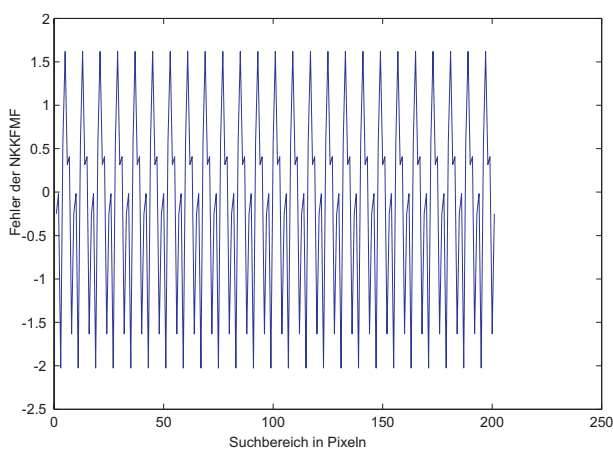


(b) *SSD auf LoGD*

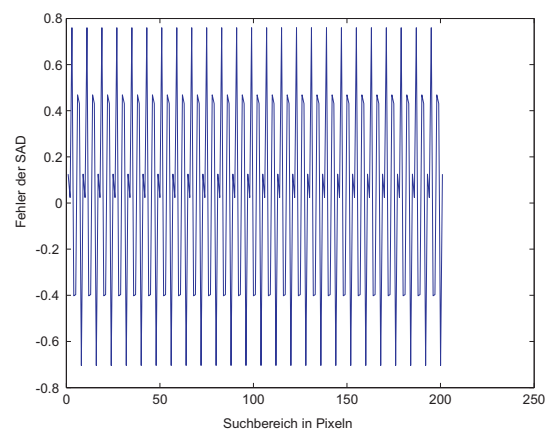
Abbildung B.14: Anwendung der SAD und der SSD-Funktion auf das mit dem Marr-Hildreth-Operator gefilterte Bildpaar IM23(Zeile 600)

Die Grafiken [B.13](#) und [B.14](#) zeigen alle den Korrelationsverlauf des Blockes 12 aus der Zeile 600.

Bei den synthetischen Bildern, die kein Rauschen und eindeutige nahezu identische Signale enthalten stellen die SAD-Funktion und die QNKKFMF die beste Wahl dar. Beide erzeugen eindeutige Extrema und der Fehler der Subpixeldisparität liegt in einem akzeptablen Bereich. Die SSD bringt hierbei keine Vorteile. Auch die mittelwertfreien Kriterien kommen hier nicht zur Geltung. Im Anschluss werden einige Vergleiche mit realen Bildern durchgeführt.



(a) *Fehler der NKKFMF*



(b) *Fehler der SAD*

Abbildung B.15: Vergleich der Fehler der subpixelgenauen Disparität der KKFMF und der SAD bei der Auswertung des Bildpaares IM23 über den gesamten Disparitätsbereich von 128 Pixeln



(a) linkes Bild



(b) rechtes Bild



(c) untersuchte Objekte

Abbildung B.16: Reales Szene mit Betrachtung mehrerer Ziele

## B.4 Untersuchung der Korrelationskriterien mit realen Bildern

Bei der Untersuchung der realen Bilder wird nicht mehr auf die SSD-Funktion eingegangen, da sie bis auf wenige Ausnahmen ein deutlich breiteres Minimum zeigt, insgesamt jedoch ähnliche Ergebnisse erzielt. Die NKKFMF erzielt geringfügig schlechtere Ergebnisse als die QNKKFMF, welche sich ebenfalls nur selten auf die Tiefenkarte auswirken.

In Abbildung B.16 ist das linke und rechte Bild eine Außenaufnahme auf einem Parkplatz dargestellt. In diesem Bild werden verschiedene Objekte mit unterschiedlichen Beleuchtungsbedingungen betrachtet. Das erste Objekt ist eine Tafel mit einem schwarzen Streifen in ca. 100m Entfernung. Das Objekt befindet sich in einer guten Beleuchtungssituation und es bilden sich mit allen Kriterien klare Extrema heraus.

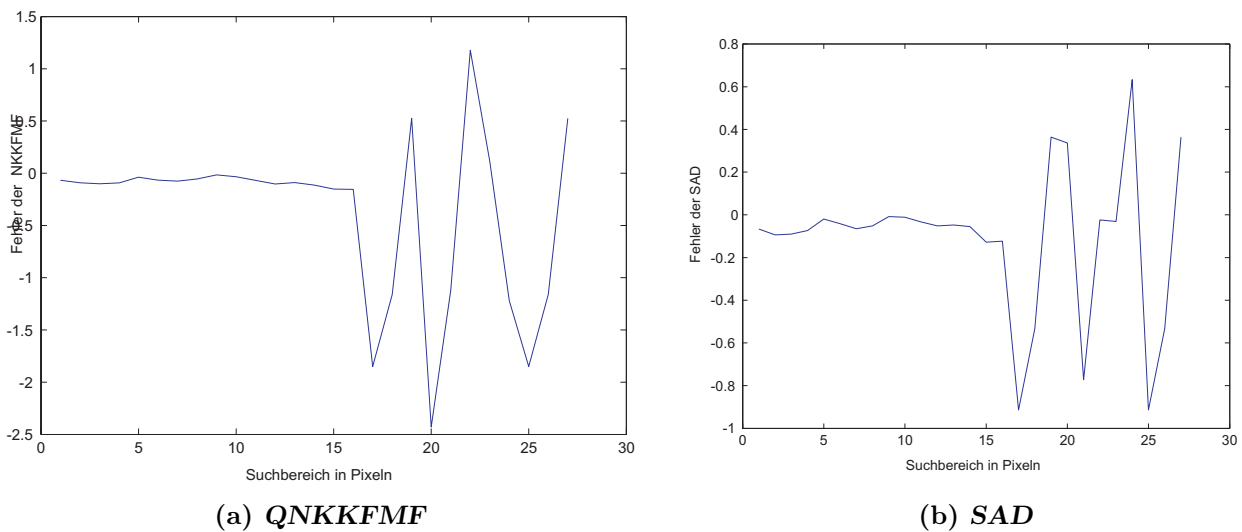


Abbildung B.17: Fehler der Subpixeldisparität für das erste Objekt (Tafel mit schwarzen Balken)(links QNKKFMF; rechts SAD)

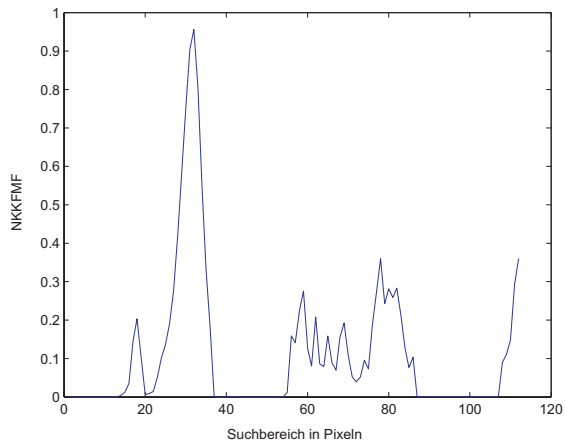
Die Abbildung B.17 zeigt links den Verlauf der Subpixelgenauigkeit gegenüber der Disparität von 16 Pixeln, die händisch aus dem Bild abgelesen wurde. Dabei wird der Fehler in einer Spalte der Tiefenkarte von oben nach unten betrachtet. Während der Fehler der subpixelgenauen Disparität zu Beginn bei beiden Kriterien ungefähr gleich ist, steigt er im unteren Teil der Tafel deutlich an und ist bei der QNKKFMF höher als bei der SAD-Funktion. Für die Darstellung der Fehlerverläufe wurden die direkten Disparitätswerte verwendet, ohne dass Disparitätswerte die aufgrund schlechter Extrema entstehen, aussortiert werden. Dies ist erforderlich damit eine Vergleichbarkeit zwischen den Kriterien erfolgen kann.

Das zweite Objekt ist ein schwarzer PKW im Schatten. Die Beleuchtungssituation ist deutlich schlechter, durch die Wagenfarbe stellt sich jedoch ein deutlicher Kontrast zur Umgebung ein. Bei der Betrachtung der Kurvenverläufe fällt auf, dass die SAD-Funktion zwar ein Minimum an der korrekten Position ausbildet, dieses aber nicht so eindeutig ist wie bei den synthetischen Bildern oder der guten Beleuchtungssituation. Da diese Bilder bezüglich der Helligkeit sehr gut geregelt sind, entsteht keine Fehlzuordnung. Die SADMF zeigt nach wie vor ein eindeutiges Minimum und lässt sich gut auswerten.

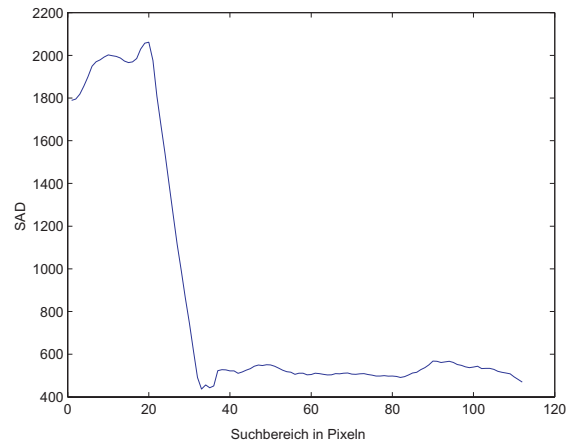
Bei Betrachtung des Fehlers der subpixelgenauen Disparität gibt es in beiden Kurven einen starken Ausreißer, der aufgrund eines sehr schlechten Kontrastes zustande kommt. Abgesehen davon ist die Positionsbestimmung mit der QNKKFMF sehr zuverlässig an diesem Objekt. Die SAD-Funktion kann insbesondere im oberen Teil des Objektes keine genaue Positionsbestimmung vornehmen.

Das dritte Objekt steht ebenfalls im Schatten, hat jedoch aufgrund seiner Farbe einen deutlich geringeren Kontrast zur Umgebung. Alle Kriterien ermitteln die korrekte Position mit einem deutlichen Extremum. Die QNKKFMF hat bei diesem Objekt größere Probleme bei der Bestimmung des korrespondierenden Blockes. Es entstehen mehrere deutliche Nebenmaxima, die

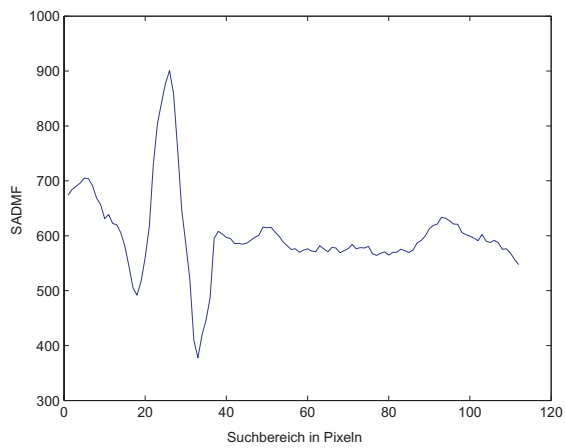
ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN



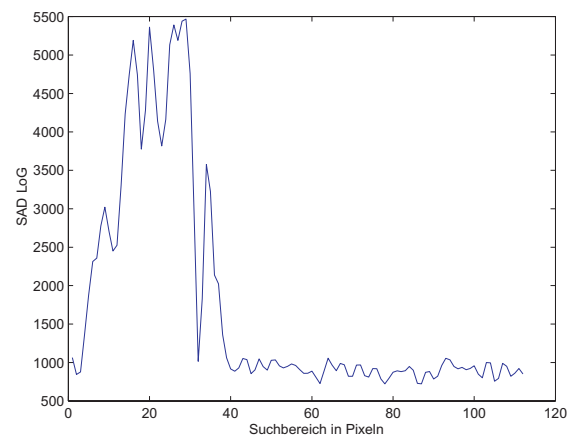
(a) *QNKKFMF*



(b) *SAD*



(c) *SADMf*



(d) *SAD LoG*

Abbildung B.18: Kurvenverläufe der Korrelationskriterien für das zweite Objekt

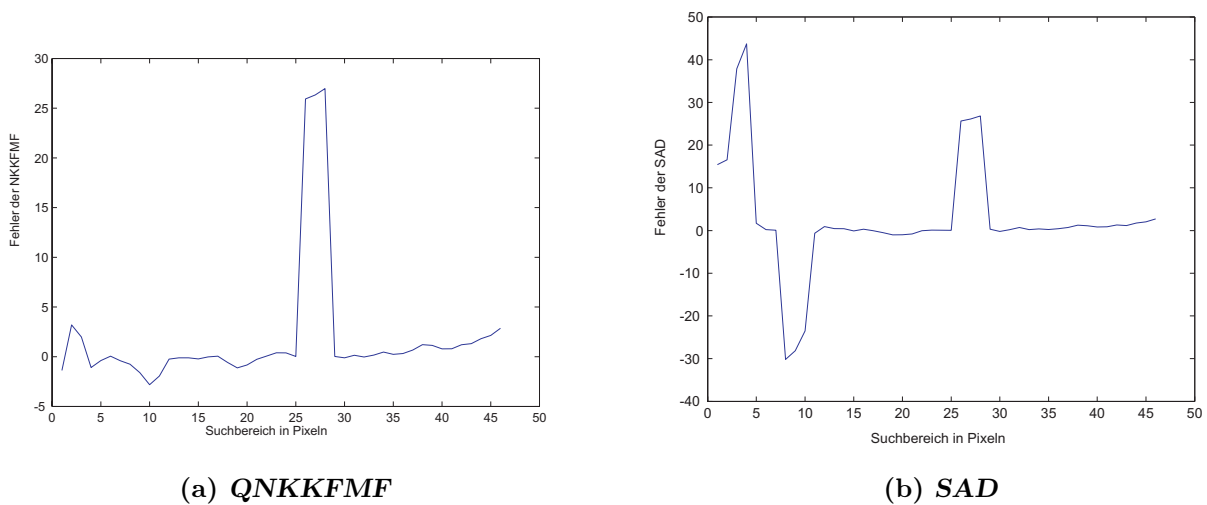


Abbildung B.19: Fehler der Subpixeldisparität für den schwarzen PKW (zweites Objekt)(links QNKKFMF; rechts SAD)

von ähnlichen Grauwertverläufen in der näheren Umgebung resultieren. Auf die SAD-Funktion haben diese Grauwertverläufe einen geringeren Einfluss. Bei der SADMF sind sie jedoch deutlich sichtbar, auch wenn die Extrema hier nicht so stark ausgeprägt sind. Die SADMF hat das eindeutigste Maximum.

Die Auswirkungen des schlechten Kontrastes zeigen sich in den Fehlern der subpixelgenauen Disparität. beide Funktionen können nur im unteren Teil des Objektes klar die Position bestimmen.

## B.5 Diskussion

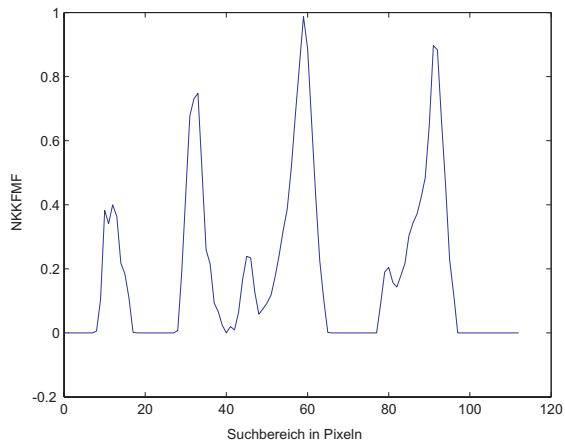
Der Vergleich verschiedener Korrelationskriterien, insbesondere der QNKKFMF und der verschiedenen Ausführungen der SAD-Funktion zeigt die Verwendbarkeit der QNKKFMF, der KKFMF, der SAD, SSD, sowie ihrer mittelwertfreien Versionen.

Die Ergebnisse der SAD und der SSD auf den Marr-Hildreth(LoG)-Operator gefilterten Bilder zeigten große Empfindlichkeiten in Regionen mit Strukturen, für die der Filterkern nicht optimal war. Durch die Filterung kann die Größe der betrachteten Objekte bestimmt werden.

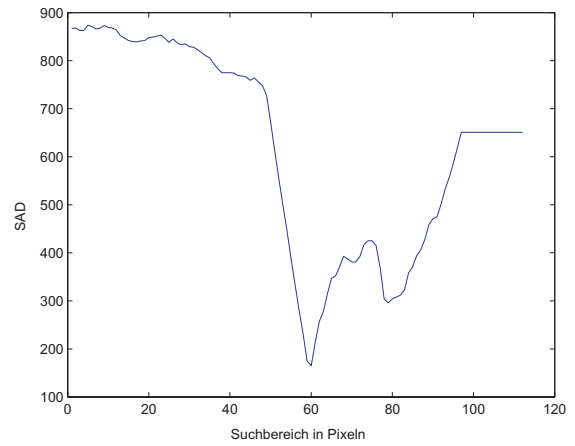
Bei der Untersuchung der synthetischen Bilder wurde auch das Verhalten bezüglich eines additiven Fehlers untersucht. Dieser wirkte sich bei der SAD-Funktion nur bei langen flach ansteigenden Gradienten aus, abgesehen davon ergaben sich keine Beeinträchtigungen. Auf die mittelwertfreien Funktionen, wie die QNKKFMF und die SADMF wirkt dieser Offset nicht.

Bei der Untersuchung realer Bilder zeigte sich eine leichte Schwäche der SAD-Funktion bezüglich der Abhängigkeit des Korrelationswertes von den aktuellen Bildinformationen eines Bildes. Läuft die SAD über einen starken Gradienten kann eventuell der Gradient allein für ein Mini-

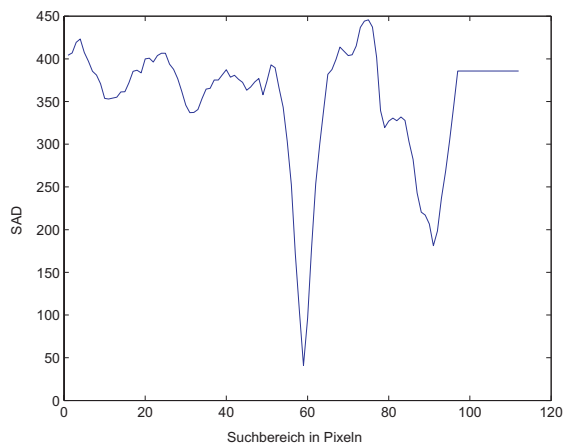
ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN



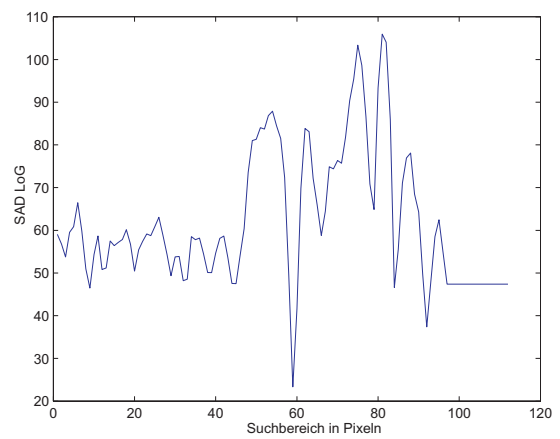
(a) *QNKKFMF*



(b) *SAD*



(c) *SADMF*



(d) *SAD LoG*

Abbildung B.20: Kurvenverläufe der Korrelationskriterien für das dritte Objekt

## ANHANG B. VERGLEICH DER EIGENSCHAFTEN VERSCHIEDENER KORRELATIONSVERFAHREN

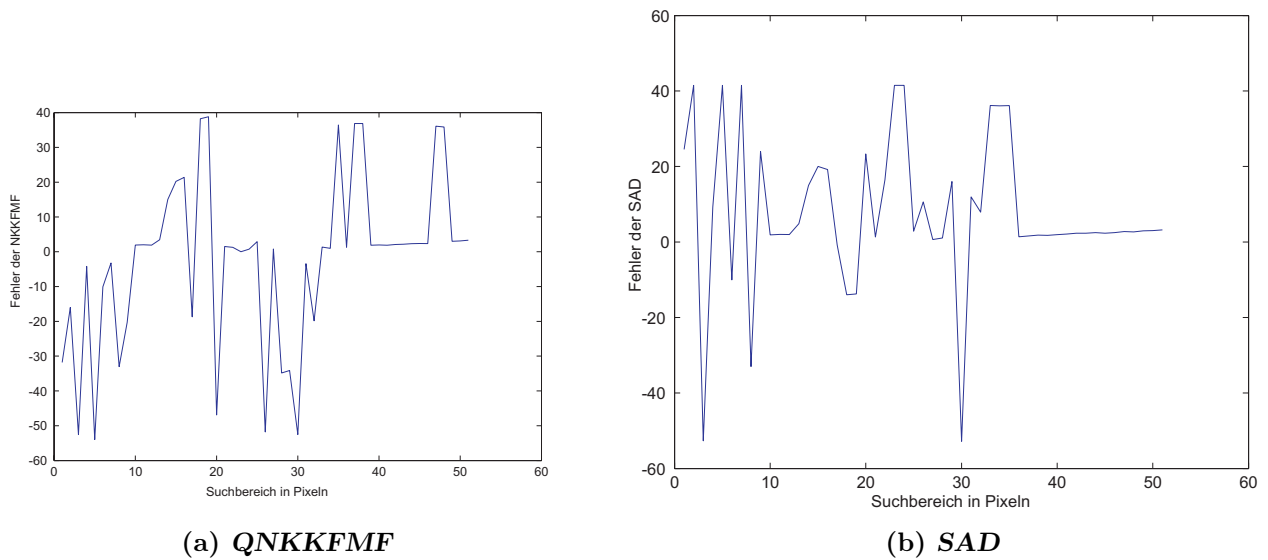


Abbildung B.21: Fehler der Subpixeldisparität für den grauen PKW im Vordergrund (drittes Objekt)(links QNKKFMF; rechts SAD)

zum ausreichen, ohne dass ein korrespondierender Punkt gefunden wurde. Die SADMF kann dieses Manko in den meisten Fällen beheben. Alternativ dazu können Aufnahmen mit gut ausgeregelter Helligkeit verwendet werden.

Die QNKKFMF hat Schwächen an niedrigen Gradienten. Es treten deutliche Nebenmaxima auf und es können fehlerhafte Korrespondenzen resultieren.

Die QNKKFMF und die SADMF bieten für den Einsatz in Anwendungsfällen mit wechselnder Beleuchtung die meisten Vorteile gegenüber anderen Verfahren und werden daher für diese Arbeit favorisiert.

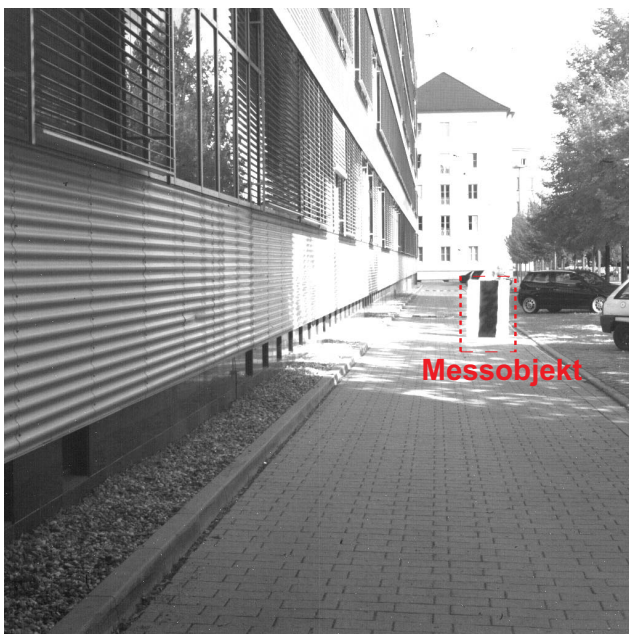




# Anhang C

## Messergebnisse

In diesem Anhang werden einige weitere Diagramme aufgeführt, die in Kapitel 6 aus Platzgründen nicht dargestellt werden konnten



(a) 30m



(b) 100m

Abbildung C.1: Linkes Bild der Messtafel in 30 und 100 m Entfernung

## Vermessung der Entfernung der Messtafel

Als Messtafel wurde eine weiße Gipskartonplatte (100 cm x 150 cm) mit einem 50 cm breiten schwarzen Streifen verwendet. Die Tafel war etwas schmaler als das Fahrzeug (10 m - Breite 198 Pixel; 30 m - Breite 68 Pixel; 100 m - Breite 21 Pixel; 144 m - Breite 15 Pixel). Die

beiden Kanten der Messtafel können wie in Abbildung 6.6 über den gesamten Messbereich gut wahrgenommen werden.

Am Aufnahmeort, ein Parkplatz der Universität, variierten an diesem Tag die Lichtverhältnisse sehr stark. Dadurch ist die Versuchsreihe auch ein Test für die Wirkung verschiedener Helligkeiten auf die Entfernungsvermessung.

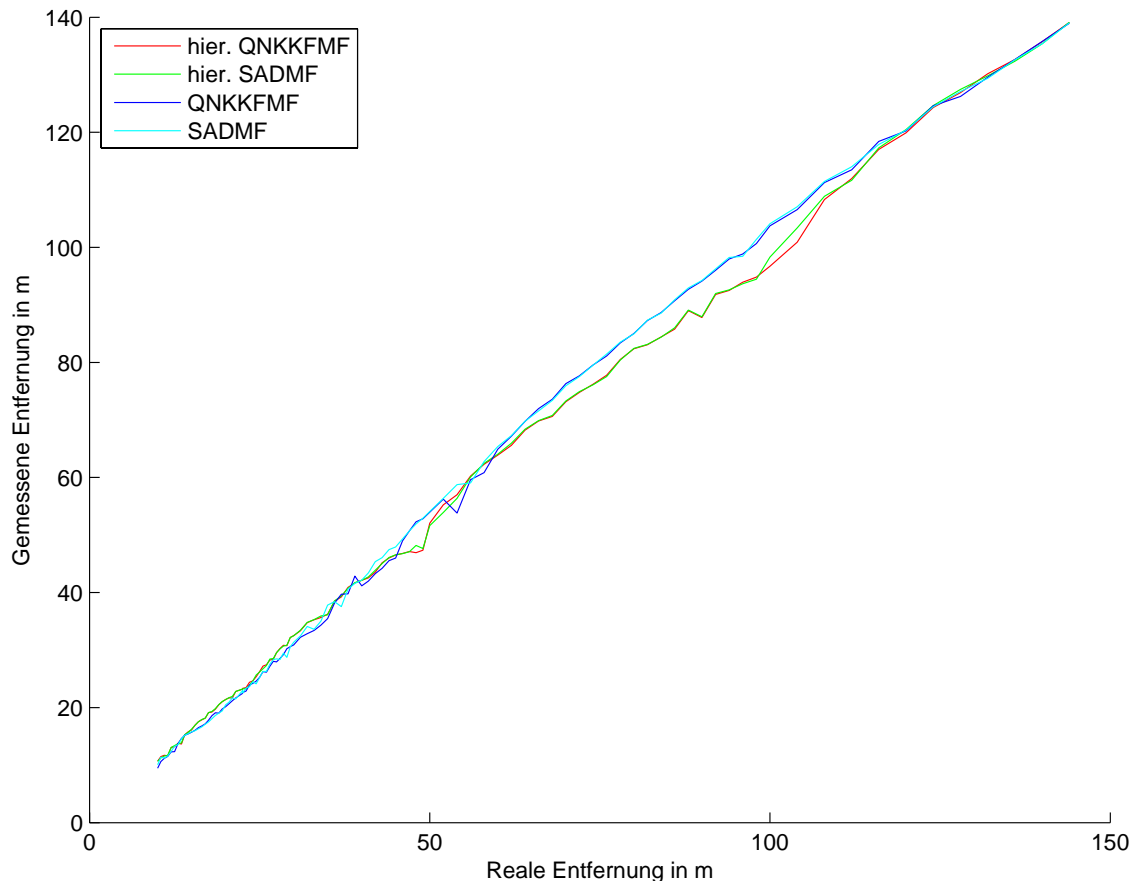
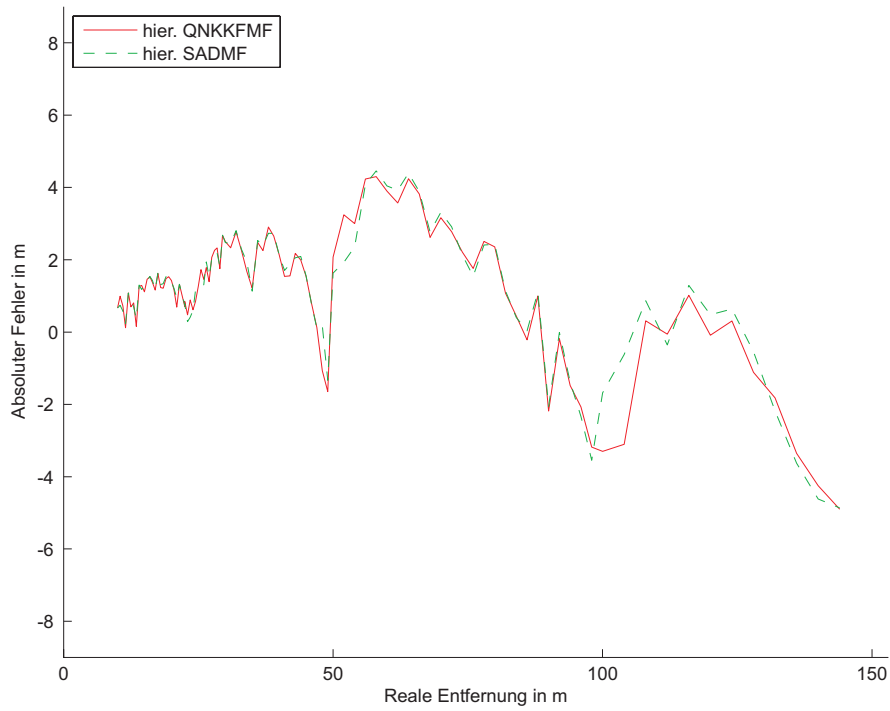


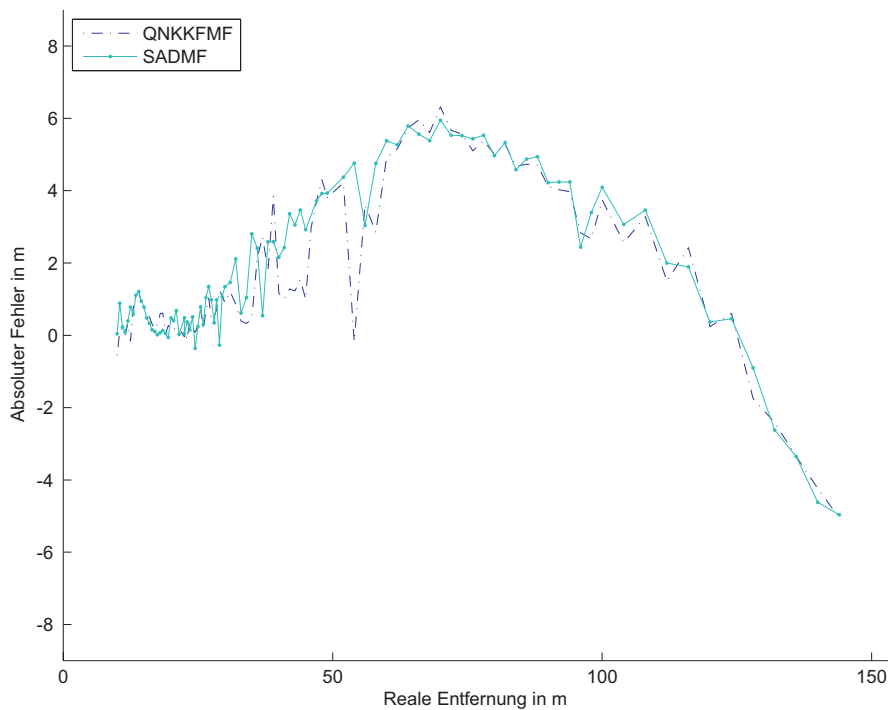
Abbildung C.2: Verlauf der Entfernung bei der Vermessung der Messtafel

Aufgrund der manuellen Platzierung der Tafel konnte der Sollwert der Entfernung mit Hilfe eines Maßbandes ausreichend genau ermittelt werden. Der Messfehler in der Handhabung sinkt dadurch im Vergleich zum Laserentfernungsmessgerät insbesondere in Entfernungen oberhalb von 100 m, da ein stabiles Anvisieren in dieser Entfernung selbst mit Stativ nur schwer möglich war.

Anschließend wurden nach der Berechnung der Tiefenkarten die Spalten der Tiefenkarte, in denen die Messtafel abgebildet ist, gegen die Referenzentfernung bewertet (siehe Abbildung 6.7(a)). Das synthetische Objekt überdeckte im Nahbereich mehr als 1000 3D-Punkte in allen Spalten, während am Ende des Messbereiches (bei 150 m) gerade noch 40 3D-Punkte ermittelt wurden.

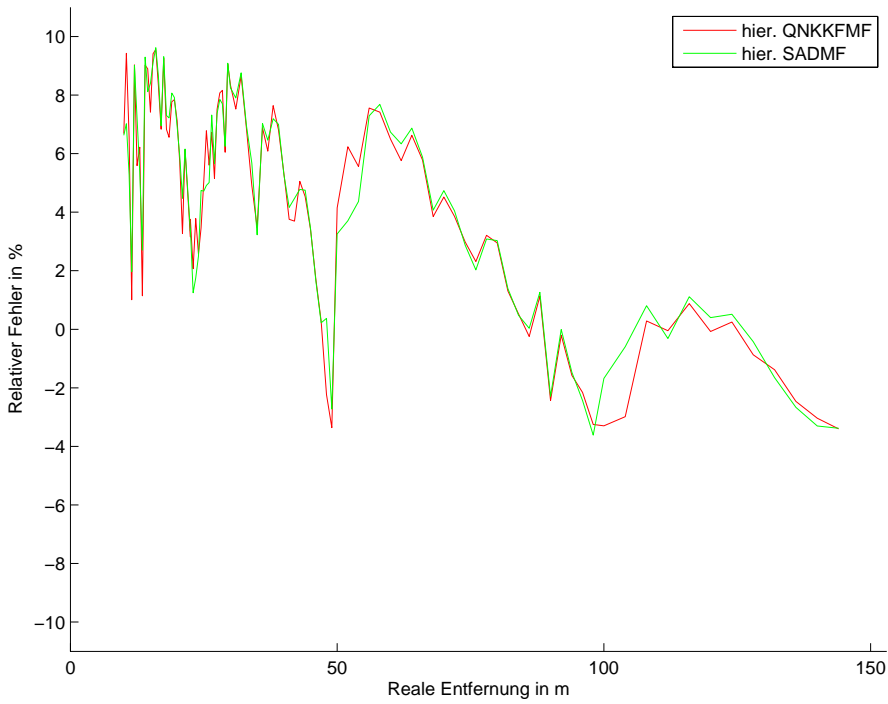


(a) *hierarchisches Verfahren*

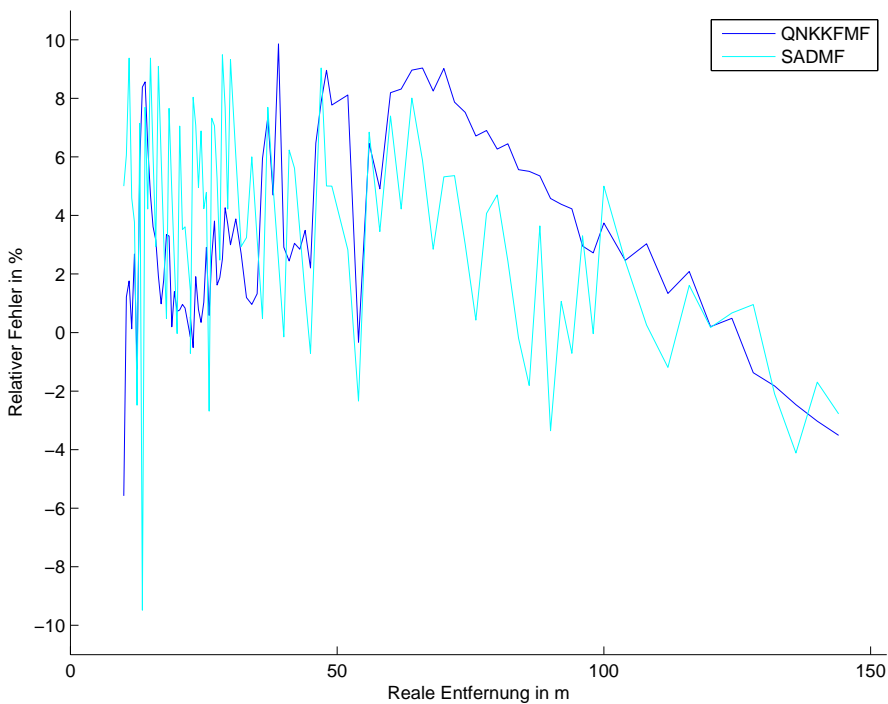


(b) *lineares Verfahren*

Abbildung C.3: absoluter Fehler beim Vermessen der Position eines Messobjektes mit hohem Kontrast im Bereich von 10-150 m



(a) *hierarchisches Verfahren*



(b) *lineares Verfahren*

Abbildung C.4: relativer Fehler beim Vermessen der Position eines Messobjektes mit hohem Kontrast im Bereich von 10-150 m

## Auswirkung der Kalibrierung

Zum Ausgleich der Dejustierung der Kameras wurde das vereinfachte Kalibrierverfahren aus Anhang D verwendet. Hierbei werden an 3 Punkten die Soll- und Ist-Werte ermittelt. Aus diesen Werten werden 3 Koeffizienten für eine quadratische Korrektur der Messergebnisse ermittelt. Diese Koeffizienten beinhalten bereits die Kamerakonstante und die Basisbreite, sodass eine separate Bestimmung dieser Größen entfällt [48]. Anschließend wird mit Hilfe dieser Koeffizienten eine Korrektur der Messergebnisse in Abhängigkeit von der Entfernung berechnet.

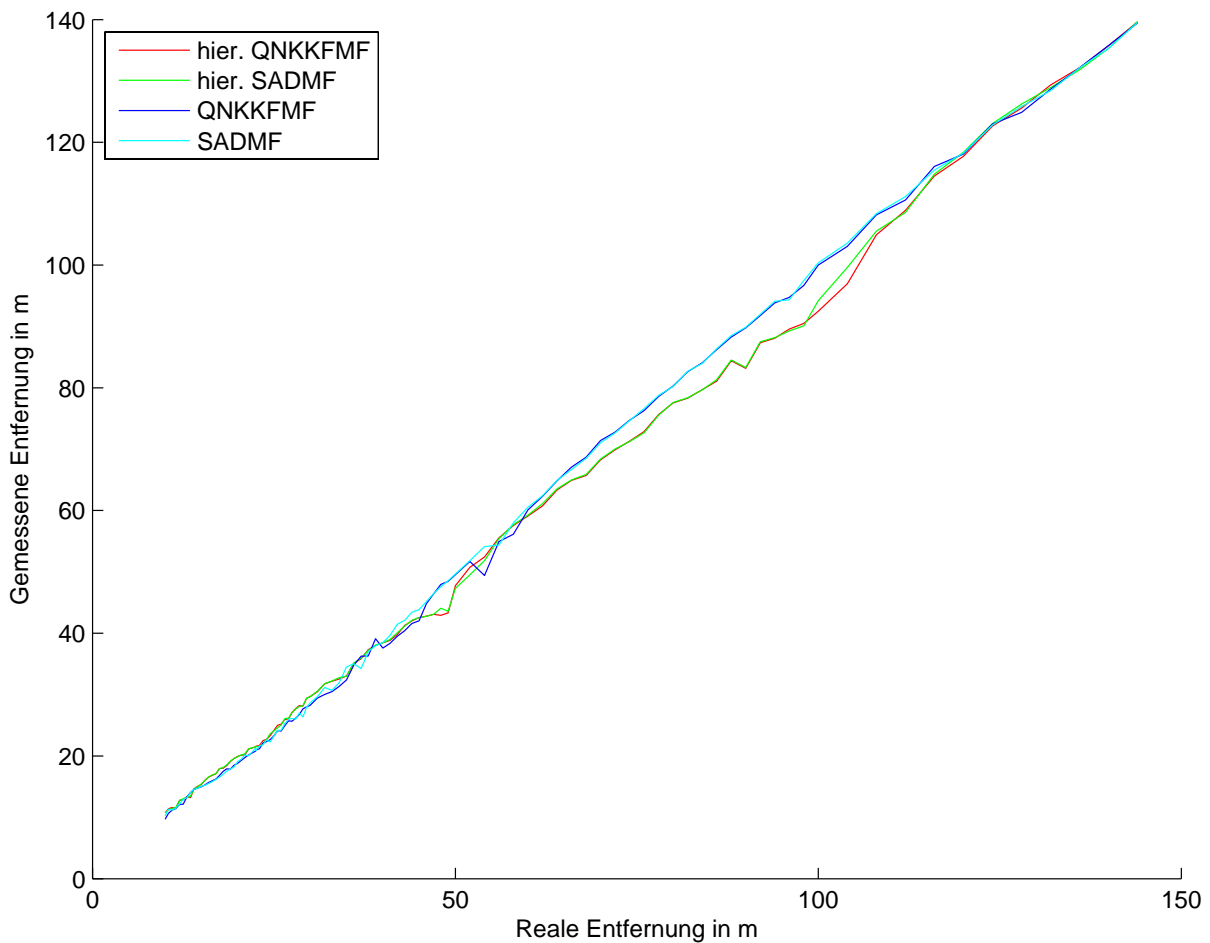
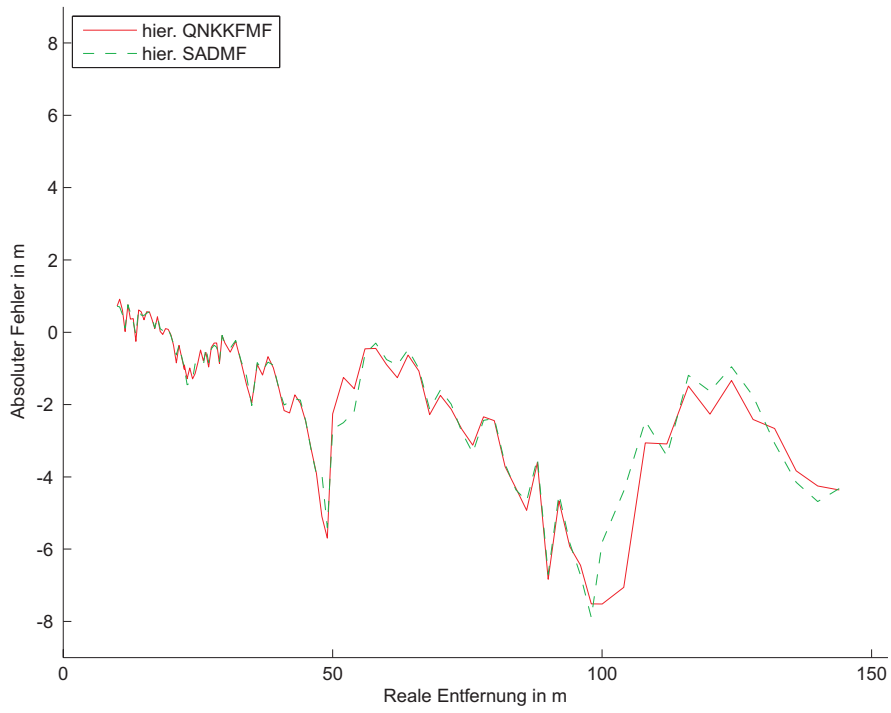
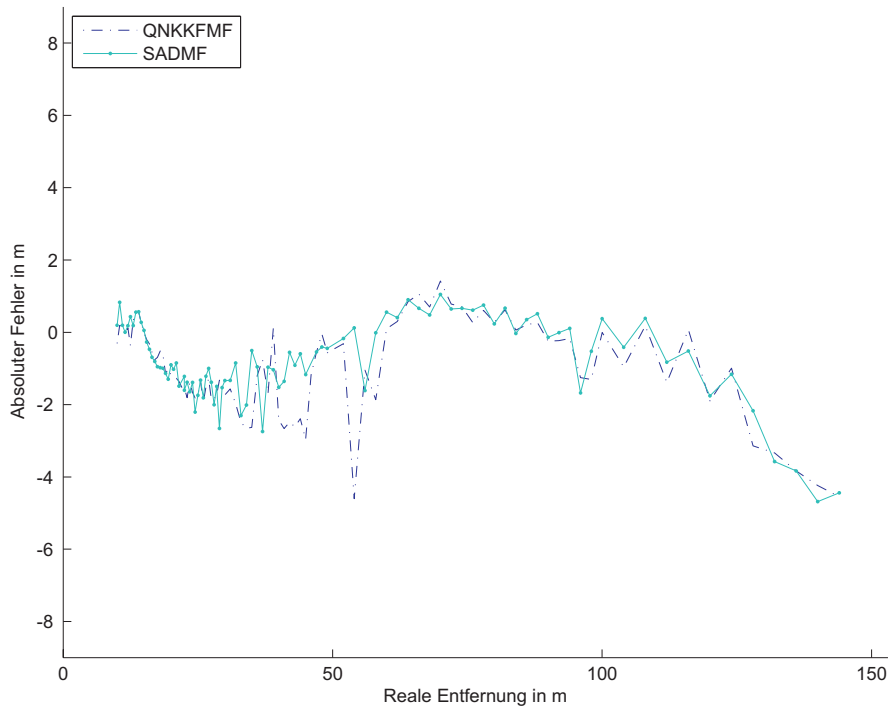


Abbildung C.5: (kalibriert) Verlauf der Entfernung

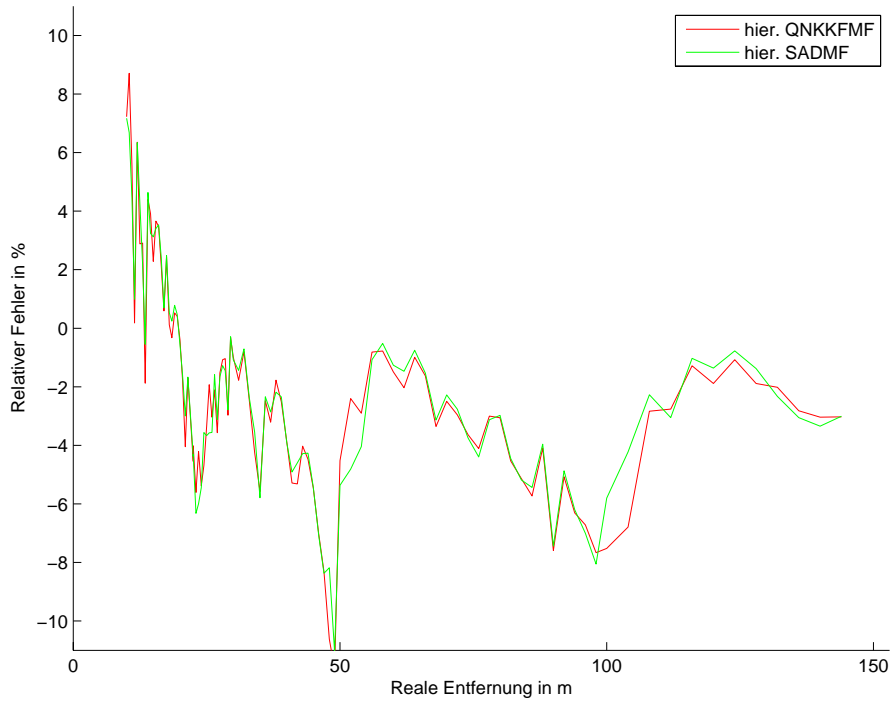


(a) *hierarchisches Verfahren*

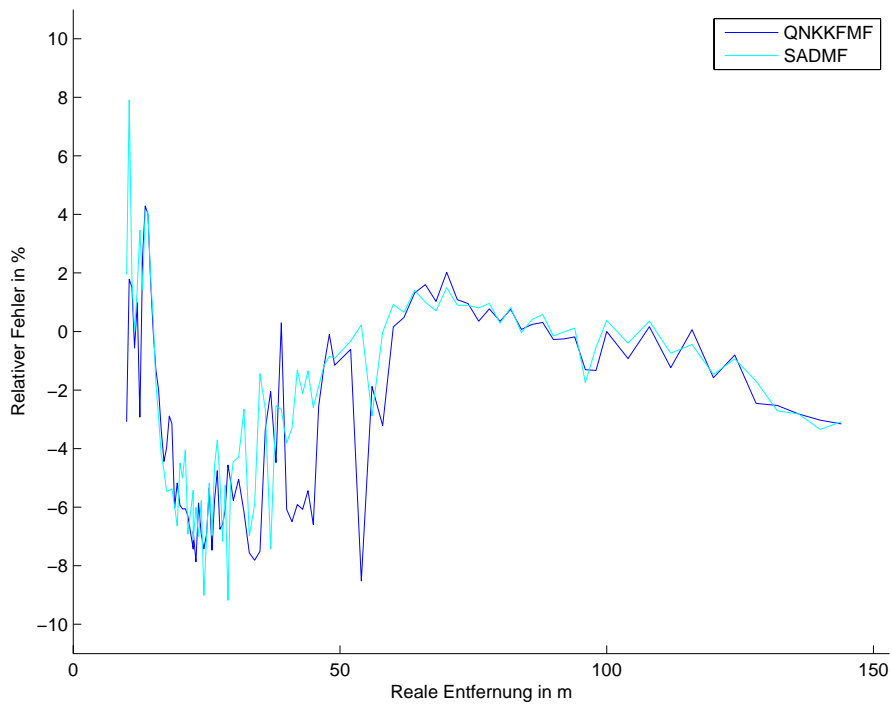


(b) *lineares Verfahren*

Abbildung C.6: absoluter Fehler (kalibriert)



(a) *hierarchisches Verfahren*



(b) *lineares Verfahren*

Abbildung C.7: relativer Fehler (kalibriert)

## Vermessung im Nahbereich

Im Nahbereich ist der zu überwachende Öffnungswinkel des Messsystems mit  $\geq 90^\circ$  deutlich größer als im Fernbereich mit  $\leq 30^\circ$ . Durch die größeren Öffnungswinkel des Messsystems sind entsprechend kleinere Brennweiten erforderlich, bei denen deutlich stärkere Verzerrungen auftreten.

Im Folgenden werden zur Untersuchung der Auswirkung der Brennweite auf die einzelnen Verfahren folgende Kamerasysteme zur Aufnahme verwendet:

1. **Kamerasystem 1** bestehend aus 2 Photonfocus-Kameras MV-D1024 angesteuert über Kameralink mit einer Basisbreite von  $b = 6,5$  cm und dem Objektiv C21211KP der Firma Cosmicar mit einer Brennweite von  $f = 12,5$  mm montiert auf zwei Goniometern und einem Hebe- und einem Drehtisch (siehe Abbildung 6.10(a))
2. **Kamerasystem 2** Bumblebee 2 [74] aufgebaut als starres System mit festmontierten Linsen mit einer Basisbreite von  $b = 12$  cm und einer Brennweite von  $f = 6$  mm; enthält Schaltkreis zur Rektifizierung in Echtzeit (siehe Abbildung 6.10(b))

In Abbildung C.8 ist von beiden verwendeten Kamerasystem jeweils ein Bild in 2,1 m und 10 m Entfernung dargestellt.

In Abbildung 6.12 ist jeweils die Gemessene Entfernung über der Realen Entfernung dargestellt. Es zeigt sich, dass die Verläufe bis zu einer Entfernung von 5 m gut mit der realen Entfernung übereinstimmen. Die Messung auf dem Kamerasystem 1 folgt dem Sollwert weiterhin sehr gut, während bei der Messung mit dem Kamerasystem 2 der Messfehler deutlich ansteigt.

Betrachtet man den absoluten Fehler (siehe Abbildung C.10(b)) der linearen Verfahren, so wird deutlich, dass die Messergebnisse dem Sollwert nahezu direkt folgen. Dadurch ergibt sich ein relativer Fehler von unter 1 Prozent (siehe Abbildung C.4(b)) über den gesamten Messbereich. Ab einer Entfernung von 5 Metern steigt der Fehler an und liegt zwischen 0,5 und 1,0 Prozent relativem Fehler. Die Auswertung mit den hierarchischen Verfahren zeigt hier deutlich die niedrigere Disparitätsauflösung im Nahbereich. Der Fehler in den ersten 5 m ist deutlich größer als der der linearen Verfahren (Abbildung C.10(a)). Er liegt jedoch im Mittel weitestgehend unter einem Prozent.

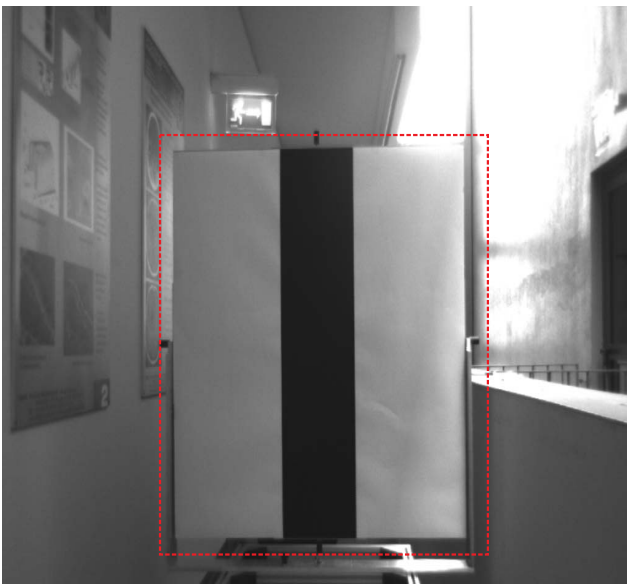




(a) *Photonfocus: Abstand 2,1m*



(b) *Photonfocus: Abstand 10m*



(c) *Bumblebee 2: Abstand 2,1m*



(d) *Bumblebee 2: Abstand 10m*

Abbildung C.8: Vergleich der Auswirkungen von Basisbreite und Brennweite auf die Messung anhand von zwei Kamerasystemen im Nahbereich: 2xPhotonfocus  $B = 6,5$  cm,  $f = 12,5$  mm; Bumblebee 2 (rektifiziert)  $B = 12$  cm,  $f = 6,3$  mm

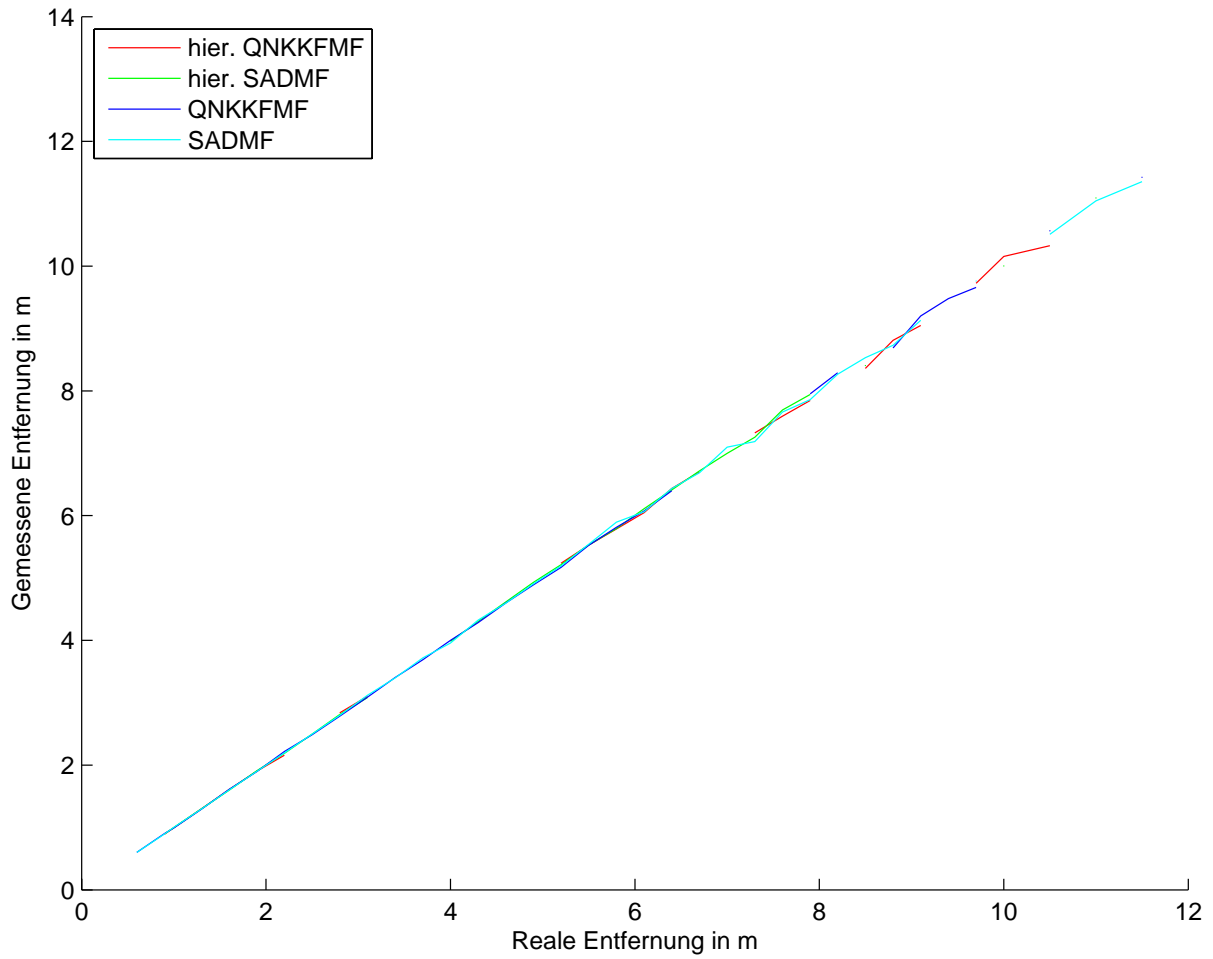
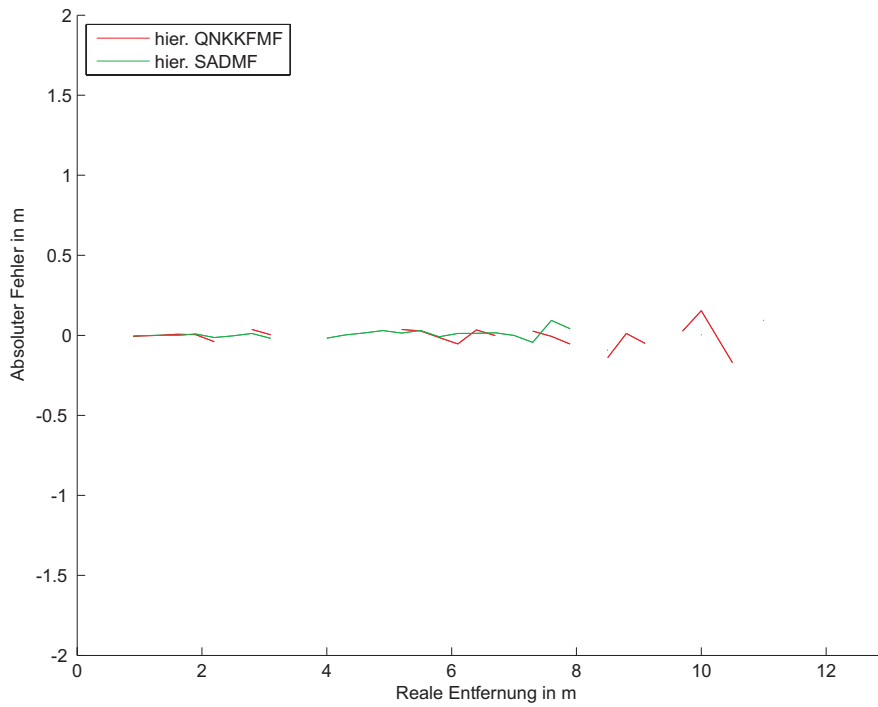
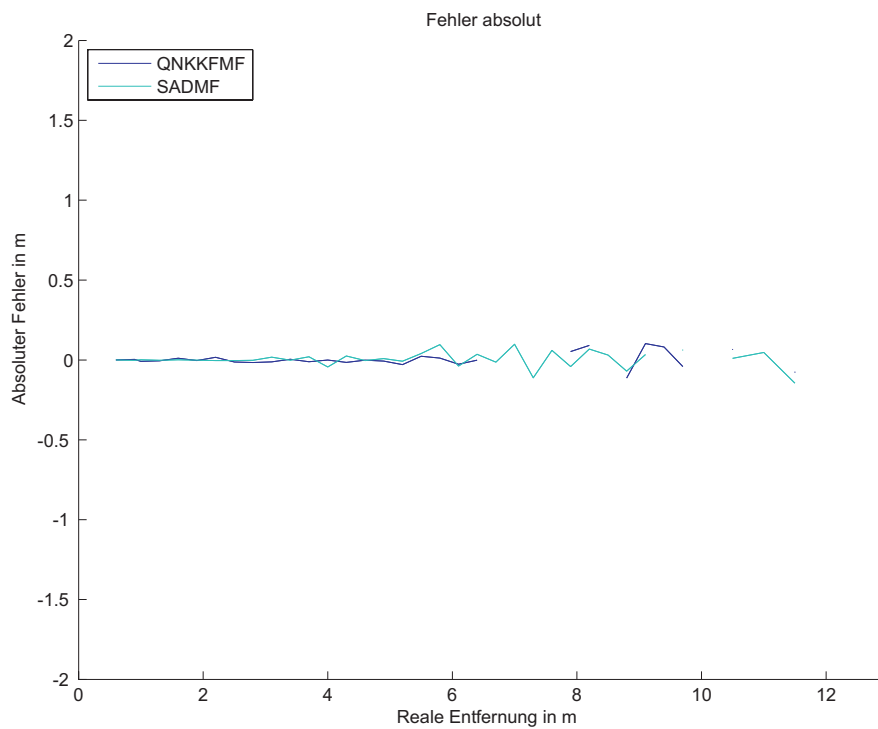


Abbildung C.9: Mittelwert der Entfernung im Nahbereich bei der Vermessung mit dem Kamerasystem 1 (Messbereich: 0,6-10m)



(a) *hierarchisches Verfahren*



(b) *lineares Verfahren*

Abbildung C.10: Absoluter Fehler beim Vermessen der Entfernung eines Messobjektes im Nahbereich mit dem Kamerasystem 1; Messbereich 0,6 - 10m

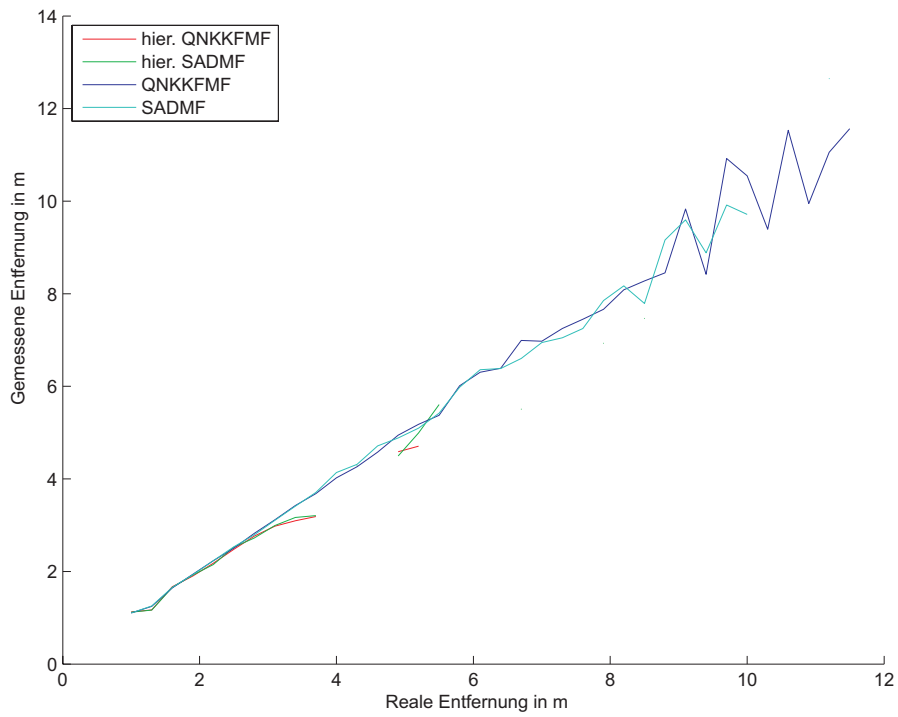


Abbildung C.11: Mittelwert der Entfernung im Nahbereich bei der Vermessung mit dem Kamerasystem 2 (Messbereich: 1-10 m)

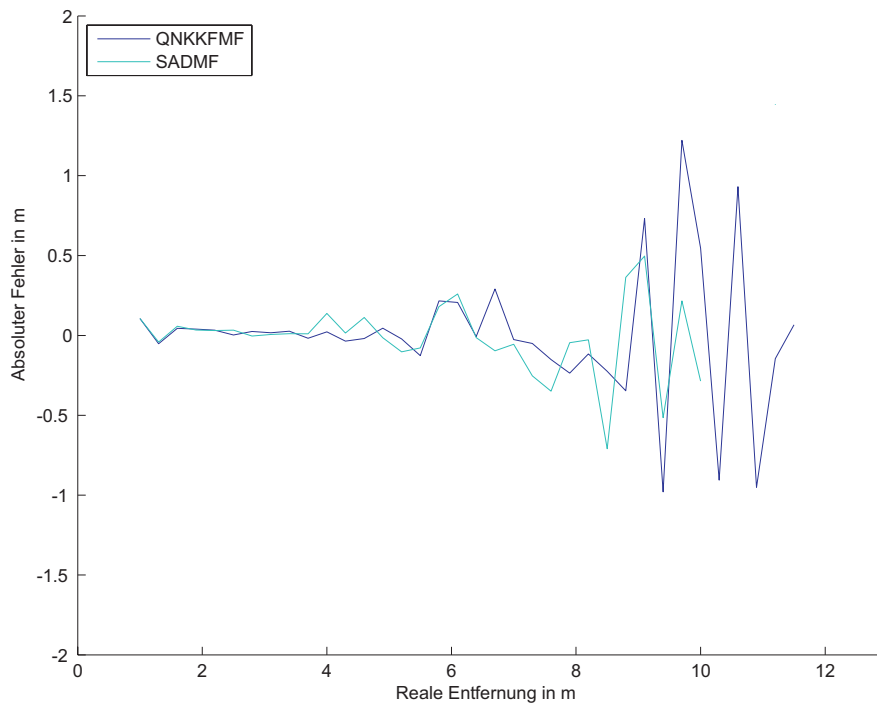


Abbildung C.12: Absoluter Fehler beim Vermessen der Entfernung eines Messobjektes im Nahbereich mit dem Kamerasystem 2; Messbereich 1 - 10m

# Anhang D

## Korrektur systematischer Fehler bei Normalfallaufnahmen im Objektraum

(nach Robert W. Kuhn [48])

### Korrektur des Abstandes

Die Koordinaten  $X$ ,  $Y$ , und  $Z$ <sup>1</sup> eines Punktes, welcher von zwei Kameras in Normalfallanordnung aufgenommen wurde, berechnen sich laut Kraus ([46]) wie folgt:

$$Z = \frac{c \cdot B}{du}, \quad X = \frac{Z \cdot x}{c}, \quad Y = \frac{-Z \cdot y}{c} \quad (\text{D.1})$$

mit

- $c$  = Kamerakonstante
- $B$  = Basisbreite
- $du$  = Disparität
- $x$  = horizontale Pixelkoordinate
- $y$  = vertikale Pixelkoordinate

Abweichungen zu den Voraussetzungen des Normalfalles wirken sich auf die Bildkoordinate  $x$  laut Albertz ([4], Seite 163) folgendermaßen aus:

---

<sup>1</sup> $X$  sei hier die horizontale Komponente,  $Y$  die vertikale und  $Z$  der Abstand

	$dx$
Fehler der Kamerakonstanten $dc$	$+\frac{X}{c} \cdot dc$
Hauptpunktverschiebung $dx_0$	$+dx_0$
Kantungsfehler $d\kappa$	$-Y \cdot d\kappa$
Neigungsfehler $d\omega$	$-\frac{XY}{c} \cdot d\omega$
Schwenkungsfehler $d\phi$	$-c \cdot (1 + \frac{X^2}{c^2})d\phi$

Es ist ersichtlich, dass der jeweilige Fehlereinfluss auf die Bildkoordinate unabhängig vom Abstand  $Z$  ist. Aus Gleichung D.1 erhält man für die Standardabweichung von  $Z$ :

$$\sigma_z = \frac{c}{du} \cdot \sigma_B + \frac{B}{du} \cdot \sigma_c - \frac{c \cdot B}{du^2} \cdot \sigma_{du}$$

Nach Erweiterung der einzelnen Terme mit  $(B)$ ,  $(c)$  bzw.  $(c \cdot B)$  erhält man

$$\sigma_z = \frac{Z}{B} \cdot \sigma_B + \frac{Z}{c} \cdot \sigma_c - \frac{Z^2}{c \cdot B} \cdot \sigma_{du} \quad (\text{D.2})$$

Somit ergibt sich für den mittleren Fehler  $m_Z$  folgende Gleichung:

$$m_Z = \pm \frac{Z}{B \cdot c} \cdot \sqrt{c^2 m_B^2 + B^2 m_c^2 + Z^2 m_{du}^2}$$

Die hier interessierende Eigenschaft eines systematischen Fehlers ist seine Konstanz. Das bedeutet für den Normalfall die Konstanz von  $\sigma_B$ ,  $\sigma_c$  und im Falle von  $Z$  auch von  $\sigma_{du}$ . Es lässt sich also der Einfluss des gesamten systematischen Fehlers auf die Berechnung von  $Z$  bestimmen. Man erhält infolgedessen nach Zusammenfassung von Gleichung D.2 für die Verbesserung  $\Delta Z$  eines Messwertes  $Z$  folgenden Zusammenhang:

$$\Delta Z = \frac{Z \cdot c \cdot \sigma_B + Z \cdot B \cdot \sigma_c}{c \cdot B} - \frac{Z^2}{c \cdot B} \cdot \sigma_{du} \quad (\text{D.3})$$

Natürlich lassen sich mittels Formel D.3 nur die systematischen Fehleranteile reduzieren. Durch Umformung und Einführung einer Additionskonstante erhält man für die Verbesserung  $\Delta Z$ :

$$\Delta Z = a * Z^2 + b * Z + c \quad (\text{D.4})$$

Daraus ergibt sich mittels Gleichung D.1 für ein verbessertes  $Z$  ( $\bar{Z}$ ):

$$\begin{aligned}
 \bar{Z} &= Z + \Delta Z \\
 &= \frac{c \cdot B}{du} + a \cdot Z^2 + b \cdot Z + c \\
 &= \frac{c \cdot B}{du} + a \cdot \left( \frac{c \cdot B}{du} \right)^2 + b \cdot \frac{c \cdot B}{du} + c \\
 &= \frac{c \cdot B}{du} + a \cdot \frac{c^2 \cdot B^2}{du^2} + b \cdot \frac{c \cdot B}{du} + c \\
 &= \frac{a \cdot c^2 \cdot B^2}{du^2} + \frac{c \cdot B + b \cdot c \cdot B}{du} + c \\
 &= \frac{k}{du^2} + \frac{l}{du} + c
 \end{aligned} \tag{D.5}$$

Somit ergibt sich für  $\bar{Z}$  ein einfacher formeller Zusammenhang (Gleichung D.5), für den eine separate Bestimmung von Basisbreite und Kamerakonstante nicht notwendig ist.

## Korrektur des seitlichen Versatzes (X)

Aus Gleichung D.1 folgt für die Berechnung der Standardabweichung von  $X$  unter der Annahme einer von systematischen Fehlern befreiten  $Z$ -Koordinate folgende Gleichung:

$$\sigma_X = \frac{Z}{c} * \sigma_x - \frac{Z * x}{c^2} * \sigma_c$$

Es lassen sich somit die linear wirkenden Fehleranteile mittels eines einfachen Zusammenhanges ermitteln. Es ergibt sich nach Vereinfachung ( $Z/c * \sigma_x = a * Z$ , da  $1/c * \sigma_x = const.$  sowie  $Z*x/c^2 * \sigma_c = X/c * \sigma_c = b * X$ , da  $1/c * \sigma_c = const.$ ) und Anbringung einer Additionskonstante für die Verbesserung  $\Delta X$

$$\Delta X = a * Z + b * X + c \tag{D.6}$$

## Versuche

### Z-Koordinaten

Zur Überprüfung der Herleitung wurden Versuche durchgeführt.

Dabei wurde ein Muster entlang einer Schiene, auf der auch die Kameras befestigt waren, bewegt und die Disparität gemessen sowie die  $Z$ -Koordinate mittels der Formel D.1 berechnet. Der

Sollabstand wurde mit einem Massband bestimmt. Die Ergebnisse dieses Versuches sieht man in Abbildung D.1. Hier ist der Absolutfehler jeweils vor und nach Anbringung der Korrektur (Korrekturwert berechnet mittels Gleichung D.4) über den Sollabstand aufgetragen. Deutlich sieht man, wie vor allem bei größeren Abständen der Absolutfehler der korrigierten Messung ( $ist - soll$ ) gegenüber dem Wert der ursprünglichen Messung kleiner ist.

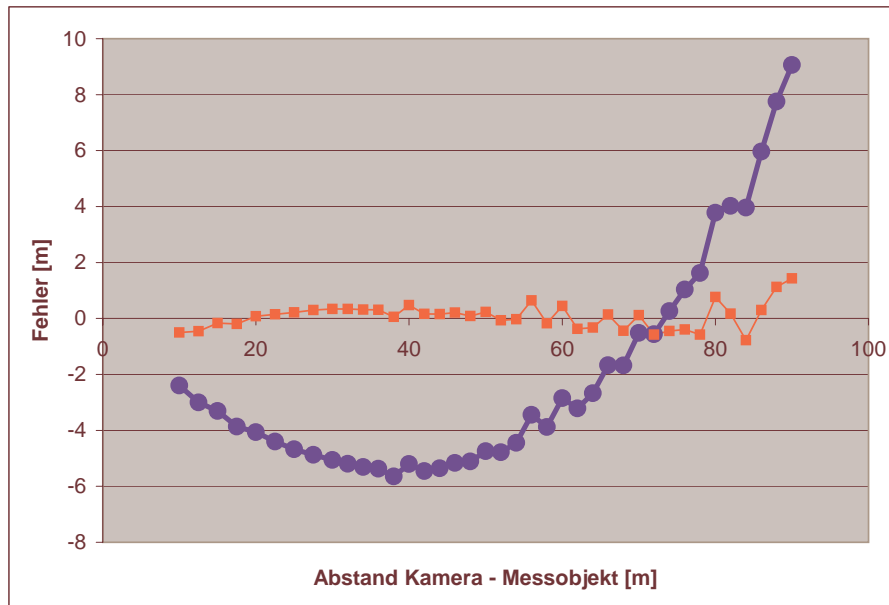


Abbildung D.1: Absolutfehler von  $Z$  nach der Korrektur. Korrekturformel:  $\Delta Z = 0,006Z^2 - 0,0111Z - 0,0762$ . Der mittlere Fehler liegt vor der Korrektur bei  $\pm 5,0m$ , nach der Korrektur bei  $\pm 0,5m$ .

In einer weiteren Messung wurde untersucht, ob sich experimentell nachweisen lässt, dass die Korrektur der  $Z$ -Koordinaten unabhängig von der Lage des Messobjektes in  $X$ -Richtung ist. Dazu wurde das bereits erwähnte Muster jeweils in verschiedenen Abständen zu den Kameras lateral, also auf der  $X$ -Achse, bewegt und die jeweilige Disparität gemessen. Das Ergebnis der Messung ist in Abbildung D.2 dargestellt. Man erkennt deutlich, dass der Fehler von  $Z$  unabhängig von der  $X$ -Koordinate des Messobjektes ist, die Lage und der Verlauf der einzelnen Graphen ist fast identisch. Die Differenz war stets kleiner als ein halber Pixel und lag somit im Bereich der Messgenauigkeit. Anzumerken ist, dass in allen Messungen hochwertige Objektive zum Einsatz kamen.

## X-Koordinaten

Analog zu den Versuchen für  $Z$  wurden auch für die Korrektur der  $X$ -Koordinate Versuche durchgeführt. Dazu wurde wieder ein Testmuster in verschiedenen Abständen zur Kamera aufgenommen. Zusätzlich wurde aber auch noch die  $X$ -Position variiert. Die Sollkoordinaten in  $X$ - und  $Z$ -Richtung wurden ebenso mittels eines Massbandes bestimmt.



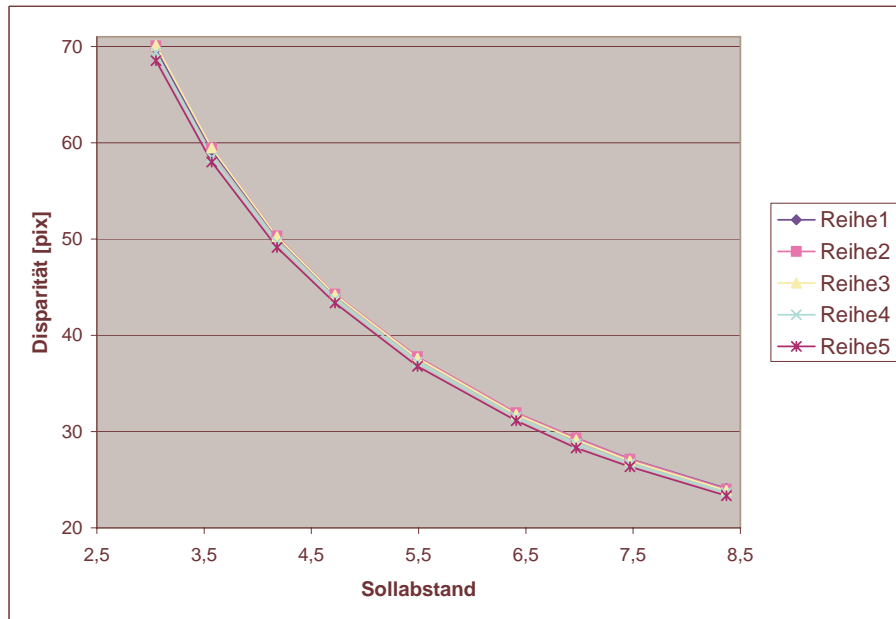


Abbildung D.2: Disparität gemessen in verschiedenen Abständen und lateralem Versatz des Messobjektes. Es wurde in  $X$ -Richtung von  $-2,5\text{m}$  bis  $2,5\text{m}$  bewegt.

In Abbildung D.3 sind die Soll- und die Istwerte der  $X$ -Koordinate über dem Abstand Kamera-Messobjekt aufgetragen. Die Sollwerte bewegen sich dabei zwischen  $-1.0\text{m}$  und  $1.0\text{m}$ . Die Istwerte liegen zwischen  $0.8\text{m}$  und  $8.5\text{m}$ . Nach der Korrektur bewegen sich die Werte der  $X$ -Koordinaten zwischen  $-1.2$  und  $1.1\text{m}$ , wie auch in Abbildung D.4 ersichtlich. Zu beachten ist, dass eine Hauptpunktkorrektur bei der Berechnung von  $X$  nicht angebracht wurde, diese also auch mittels Gleichung D.6 beachtet werden kann.

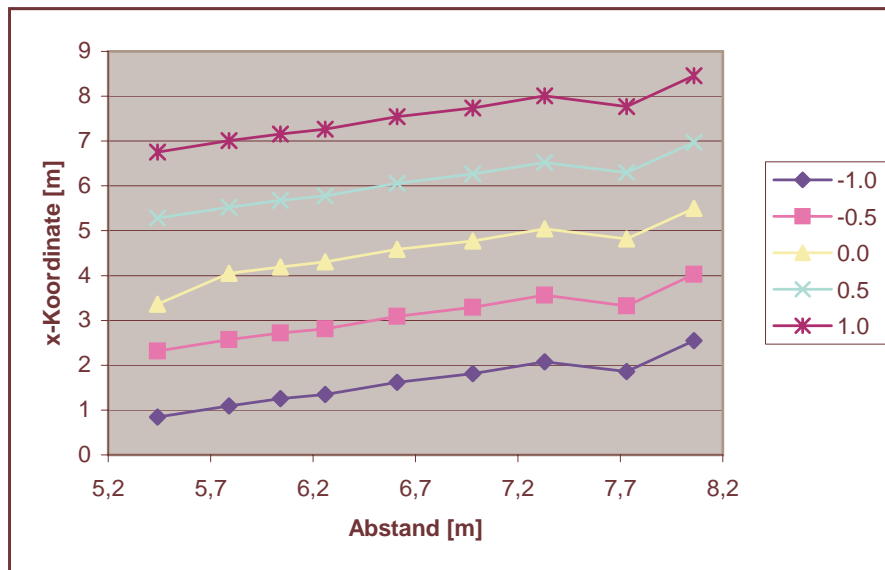


Abbildung D.3: Absolutfehler von X vor der Korrektur. Die einzelnen Sollwerte sieht man in der Legende. Die maximale Abweichung beträgt ca. 7m

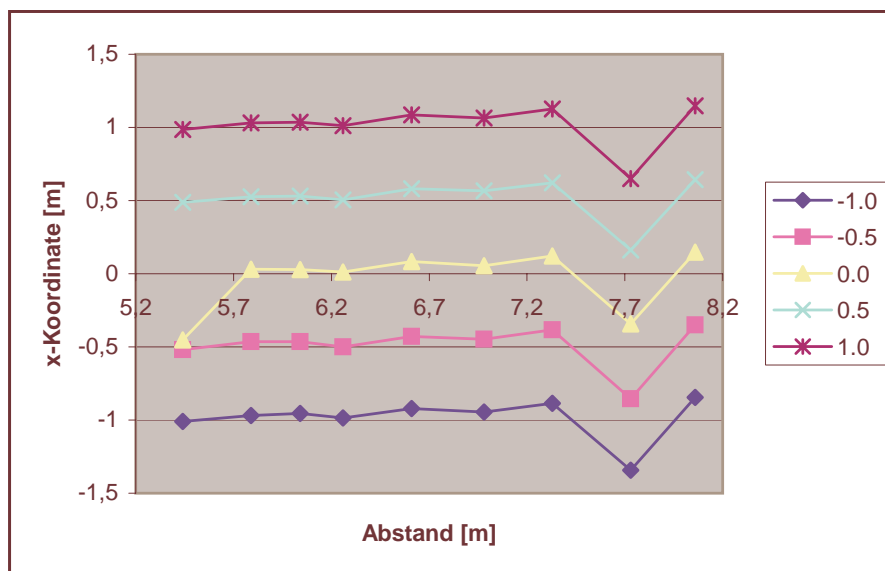


Abbildung D.4: Absolutfehler von X nach der Korrektur.  
 Korrekturformel:  $\Delta X = 0.588 * Z + 1.958 * X + 0.613$

# Literaturverzeichnis

- [1] ABDEL-AZIZ, Y.I. ; KARARA, H. M.: Direct linear transformation from comparator coordinates into object space coordinates in close range photogrammetry. In: *In Proc. ASP/UI Symposium on Close-Range Photogrammetry*. Urbana, IL, January 1971, S. 1–18
- [2] AG, Bosch: *www.bosch.de*. Bosch AG
- [3] AHO, Alfred V. ; LAM, Monica S. ; SETHI, Ravi ; ULLMAN, Jeffrey D.: *Compiler - Prinzipien, Techniken und Werkzeuge*. Zweite Auflage. Pearson Studium Addison-Wesley, 2008
- [4] ALBERTZ, Joerg ; KREILING, Walter: *Photogrammetric Guide*. 4. Herbert Wichmann Verlag GmbH, 1989
- [5] ALTERA: *Simultaneous Multi-Mastering with the Avalon Bus*. 1. Altera, 2002
- [6] ALTERA: *Stratix Device Handbook*. 1. Altera, 2004
- [7] Analog Devices, Inc.: *Analog Devices TigerSHARC ADSP-TS201S Processor Hardware Reference, Revision 1.1*. Dezember 2004
- [8] Analog Devices, Inc.: *Analog Devices Blackfin ADSP-BF533 Processor Hardware Reference, Revision 3.2*. Juli 2006
- [9] ASCHWANDEN, Peter F.: *Experimenteller Vergleich von Korrelationskriterien in der Bildanalyse*, ETH Zürich, Dissertation, 1993
- [10] AUER, Adolf ; RUDOLF, Dieter J.: *FPGA : feldprogrammierbare Gate arrays*. Heidelberg : Hüthig, 1995, 1995 ( ISBN 3-7785-2359-7)
- [11] AYACHE, N. ; HANSEN, C.: Rectification of images for binocular and tricocular stereovision. In: *Proc. 9th IAPR Int. Conf. on Pattern Recognition*. Rome, Italy, 1988, S. 11–16
- [12] BÄHR, Hans-Peter (Hrsg.) ; VÖGTLE, Thomas (Hrsg.): *Digitale Bildverarbeitung*. 4. Heidelberg : Herbert Wichmann Verlag, 2005. – ISBN 3-87907-409-7
- [13] BARBE, D.F.: Imaging devices using the charge-coupled concept. In: *Proceedings of the IEEE* Bd. 63 IEEE, IEEE, Jan. 1975, S. 38–67

## LITERATURVERZEICHNIS

- [14] B.K.P.HORN: Obtaining Shape from Shading Information. In: *McGrawHill* (1975), S. 115–155
- [15] BORG, M. ; MENTZER, R. ; SINGH, K.: Digital imaging using CMOS sensors. In: *International IC China* (2001), S. 37–47
- [16] BOUFAMA, B. ; MOHR, R.: Epipole and fundamental matrix estimation using virtual parallax. In: *Proc 5th Int. Conf. on Computer Vision*. Cambridge, MA, 1995, S. 1030–1036
- [17] BRINKSCHULTE, Uwe ; UNGERER, Theo: *Mikrocontroller und Mikroprozessoren*. Springer Verlag, 2002 (Berlin, Heidelberg)
- [18] BÄHRING, Helmut: *Mikrorechner - Technik, Mikroprozessoren und digitale Signalprozessoren*. Bd. Band 1. Springer Verlag, 2002
- [19] DANIEL, Florian (. ; KRELL, Gerald ; SCHNELTING, Olaf (. ; SCHUENEMANN, Stefan (.: Elektronische Bildkorrektur für Rückprojektionsdisplays zur Qualitätsverbesserung und Kostenoptimierung. In: *Bildschirme und Anzeigesysteme, ihre Bauelemente und Baugruppen(18. Electronic displays 2003 Wiesbaden)*. Hagenburg : Network-OSE GmbH, September 2003, S. 140–146
- [20] DEBEVEC, Paul E.: Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In: *Proceedings of SIGGRAPH 98, Computer Graphics Proceedings, Annual Conference Series*. (July 1998, Orlando, Florida) : Addison Wesley. Edited by Michael Cohen. ISBN 0-89791-999-8, July 1998, S. 189–198
- [21] DEEPSEA, Tyzx ; SPEED, High ; SYSTEM, Stereo V. *Computer Vision and Pattern Recognition*
- [22] EETIMES. *Programmable Logic (FPGA, CPLD, PLD) Top Ten Suppliers 1998 Ranking*. EETimes 4-OCT-1999. October 1999
- [23] EL-ETRIBY, S. ; AL-HAMADI, A.K. ; MICHAELIS, B.: Dense Stereo Correspondence with Slanted Surface using Phase-based Algorithm. In: *IEEE International Symposium on Industrial Electronics Industrial Electronics, 2007. ISIE 2007.*, 2007, S. 1807–1813
- [24] ENGEL, P: Die rhythmischen Schwankungen der Reaktionszeit beim Menschen. In: *Psychological research* Vol. 32 (1969), Nr. No. 4, S. p. 324
- [25] FAUGERAS, O. D. ; TOSCANI, G.: The Calibration Problem for Stereo. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*. Miami, FL, 1986, S. 15–20
- [26] FLYNN, M.: Some Computer Organizations and Their Effectiveness. In: *IEEE Trans. Comput.* Vol. C-21 (1972), S. 948, 1972

- [27] FOSSUM, Eric R.: Active Pixel Sensors: Are CCD's Dinosaurs? In: ED., Morley M. B. (Hrsg.): *Charge-Coupled Devices and Solid State Optical Sensors III, Proc. SPIE* Bd. Vol. 1900, 1993, S. 2–14
- [28] FREESCALE: *Quad Core Digital Signal Processor MSC8144*. MSC8144-Rev.12, 8/2008. Tempe, Arizona, USA: Freescale Semiconductor, Inc., August 2008
- [29] FUERSTENBERG, Kay C. ; DIETMAYER, Klaus C. J. ; LAGES, Ulrich: Laserscanner Innovations for Detection of Obstacles and Road. In: *AMAA 7th International Conference on Advanced Microsystems for Automotive Applications*, AMAA, 2003
- [30] GÖTTING, H.-H.: Automatisiertes Fahren mit Nutzfahrzeugen / Götting KG. Lehrte, January 2001. – Forschungsbericht
- [31] HENKEL, R.D.: Locking onto 3D-Structure by a Combined Vergence and Fusion System. In: *Second International Conference on 3-D Imaging and Modeling*, 3DIM '99, 1999
- [32] HENKEL, Rolf. D.: Fast stereovision by coherence detection. In: *Computer Analysis of Images and Patterns*, Springer Berlin / Heidelberg, 1997, S. (297–204)
- [33] HIRSCHMÜLLER, Heiko ; INNOCENT, Peter R. ; GARIBALDI, Jon: Real-Time Correlation-Based Stereo Vision with Reduced Border Errors. In: *International Journal of Computer Vision* Volume 47, no. 1 (2002), S. 229–246
- [34] HOEFFINGER, Bernd: High-Dynamic-Range (HDR) Vision. In: HOEFFINGER, Bernd (Hrsg.): *Advanced Microelectronics* Bd. 26. 2007
- [35] INC., TYZX: *www.tyzx.com*. TYZX Inc.
- [36] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 6. Heidelberg : Springer Verlag, 2005. – ISBN 3-540-24999-0
- [37] KANADE, T. ; OKUTOMI, M.: A stereo matching algorithm with an adaptive window: Theory and Experiment. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* Bd. 16, 1994, S. p. 920
- [38] KASZUBIAK, J. ; TORNOW, M. ; KUHN, R.W. ; MICHAELIS, B. ; BRESCH, H.: Hardware-software co-design for an optical, real-time object detection and tracking system. In: *Electronic Proceedings of the GSPx'04*. Santa Clara/USA : GSPx'04, September 2004
- [39] KASZUBIAK, Jens: *Automatisierte Hardware-Software Partitionierung am Beispiel eines eingebetteten, echtzeitfähigen Stereobildanalyse-systems in Kraftfahrzeugen*, Otto-von-Guericke Universität Magdeburg, Dissertation, April 2008
- [40] KASZUBIAK, Jens ; TORNOW, Michael ; KUHN, Robert W. ; MICHAELIS, Bernd: Realtime, 3-D-multi object position estimation and tracking. In: *Proceedings of ICPR'04*. Cambridge/UK, 2004, S. 785–788

## LITERATURVERZEICHNIS

- [41] KASZUBIAK, Jens ; TORNOW, Michael ; KUHN, Robert W. ; MICHAELIS, Bernd ; KNÖPPEL, Carsten: Real-Time Vehicle and Lane Detection with Embedded Hardware. In: *Proceedings of IEEE Intelligent Vehicle Symposium* Bd. Vol. 5 IEEE Intelligent Vehicle Symposium, 2005, S. 111–116
- [42] KLIER, Christian: *Entwurf und Implementierung einer adaptiven Helligkeitsregelung für ein Stereokamerasystem*, Otto-von-Guericke Universität Magdeburg, Studienarbeit, Juli 2006
- [43] KNOEPPPEL, C. ; SCHANZ, A. ; MICHAELIS, B.: Robust Vehicle Detection at Large Distance Using Low Resolution Cameras. In: *Proceedings of the International Conference on Intelligent Vehicles*, IEEE Industrial Electronics Society, 2000, S. 36–41
- [44] KNÖPPEL, Carsten: *Stereobasierte und spurgenaue Erkennung von Straßenfahrzeugen im Rückraum eines Straßenfahrzeuges*. Magdeburg, Otto-von-Guericke Universität, Dissertation, Dezember 2001
- [45] KRAFT, G. ; JONKER, P.P.: REal-time Stereo with Dense Output by a SIMD Computed Dynamic Programming Algorithm. In: *International Conference on Parallel and Distributed Computing, Processing Techniques and Applications* Bd. Vol 3. Las Vegas (USA), June 2002, S. 1031–1036
- [46] KRAUS, Karl ; WALDHÄUSL, Peter: *3. Bd. 1: Photogrammetrie*. Bonn : Ferd. Dümmler Verlag, 1990
- [47] KRIMMLING, Jana: *Bewertung von digitalen Signalprozessoren für optische 3D-Messverfahren und Umsetzung entsprechender Algorithmen*, Otto-von-Guericke Universität Magdeburg, Diplomarbeit, April 2008
- [48] KUHN, Robert ; KASZUBIAK, Jens ; TORNOW, Michael ; CALOW, Roman ; MICHAELIS, Bernd: Echtzeitfähiger Multipositionssensor. In: *Oldenburger 3D Tage'05*. Oldenburg/Deutschland, 2005, S. 13–17
- [49] KWEON, S. ; BAO, Y. ; FUJIWARA, N.: Motion Estimation from Sequential Image Using Correlation. In: *Proceedings of Intelligent Vehicles Symp. 1997*, 1997
- [50] KYLE, S.A.: A modification to the space resection. In: *Allgemeine Vermessungsnachrichten, International Edition* (1990), Nr. No.7, S. 17–25
- [51] LANGE, R. ; SEITZ, P.: Solid-State Time-of-Flight Range Camera. In: *IEEE Journal of Quantum Electronics* Vol. 37, no. 3 (2001), March
- [52] LASKY, T.A. ; YEN, K.S. ; RAVANI, B.: The Advanced Snowplow Driver Assistance System / AHMCT Research Report UCD-ARR-00-06-30-02. 1999. – [www.ahmct.ucdavis.edu/reports/pdf/itswc99/lasky.pdf](http://www.ahmct.ucdavis.edu/reports/pdf/itswc99/lasky.pdf)
- [53] LEE, Tai S.: Image representation using 2D Gabor wavelets. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18 (1996), Oct, Nr. No. 10, S. 959–971

- [54] LUCAS, Bruce D. ; KANADE, Takeo: An iterative image registration technique with an application to stereo vision. In: *Proceedings of Imaging Understanding Workshop*, 1981, S. 121–130
- [55] LUHMANN, Thomas: *Nahbereichsphotogrammetrie*. Heidelberg : Wichmann Verlag Heidelberg, 2000. – ISBN 3-87907-321-X
- [56] LUONG, Q. T. ; FAUGERAS, O. D.: The fundamental matrix: Theory algorithms and stability analysis. In: *Int. Journal Computer Vision* Bd. 17, 1996, S. 43–75
- [57] MAKI, A. ; BRETZNER, L. ; EKLUNDH, J.: Local Fourier Phase and Disparity Estimates: An Analytical Study. In: IN HLAVAC, Sara (. (Hrsg.): *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns*, Springer-Verlag, Berlin, Heidelberg, New York, September 1995, S. 868–873
- [58] MARK, Wannes van d. ; GARVILLA, Dariu M.: Real-Time Dense Stereo for Intelligent Vehicles. In: *IEEE Transaction on Intelligent Transportation Systems* Vol. 7 (2006), March, Nr. No. 1, S. 38–50
- [59] MARR, D. ; HILDRETH, E.: Theory of Edge Detection. In: *Proceedings of the Royal Society of London . Series B, Biological Sciences* Vol. 207 (1980), Feb, Nr. No. 1167, S. 187–217
- [60] MENGE, Matthias: *Moderne Prozessorarchitekturen*. Springer Verlag, 2005
- [61] MEYER-BAESE, Uwe: *Digital Signal Processing with Field programmable Gate Arrays*. Second Edition. Springer Verlag, 2004
- [62] MICHAELSEN, E. ; WANKMÜLLER, U. ; STILLA, U.: Wissenserwerb für Produktionsnetze zur 3D-Erkennung von Fahrzeugen. In: *Proceedings of DAGM Symposium*, 1998, S. 507–514
- [63] MOORE, C.J. ; TORNOW, M. ; SHARROCK, P.J. ; LILLEY, F. ; LALOR, M.J. ; BURTON, D.: Dynamic Background Correction in 3D-Body Surface Sensing Visualisation. In: *Proceedings of IEEE MedVis conference*. London, UK : IEEE Medical Society, July 2001, S. 444–448
- [64] MUHR, Hannes: *Einsatz von SystemC im Hardware/Software-Codesign*, Technischen Universität Wien, Diplomarbeit, November 2000
- [65] NAGEL, H. ; M-HAAG ; JEYAKUMAR, V. ; MUKERJEE, A.: Visualisation of Conceptful Description Derived from Sequences. In: *Proceedings of DAGM Symposium*, 1999, S. 346–353
- [66] NAIR, H. N. ; STEWARD, C. V.: Robust focus ranging. In: *Proc. IEEE. Conf. on Computer Vision*. Champaign, IL, 1992, S. 309–314

## LITERATURVERZEICHNIS

- [67] NGUYEN, Thien-Nghia ; MICHAELIS, Bernd ; MEINECKE, Marc-Michael ; TO, Thanh-Binh ; JERHOT, Jiri: A sensor fusion approach based on occupancy grid and fuzzy logic. In: *Proceedings of 5th International Workshop on Intelligent Transportation (WIT 2008)*. Hamburg, March 2008, S. 5
- [68] NOBLE, P.J.W.: Self-scanned silicon image detector arrays. In: *IEEE Transactions on Electron Devices* Bd. 15, 1968, S. 202–209
- [69] O., Faugeras: *Three-Dimensional Computer Vision*. MIT-Press, 1993
- [70] PHAM, D. [u. a.]: The design and implementation of a first-generation CELL processor. In: *Proceedings of International Solid-State Circuits Conference*, 2005, S. 184–185
- [71] RAPPORT, Inc.: KC256 Technical Overview / Rapport, Inc. 2007 ( v.1.9). – Forschungsbericht
- [72] REMBOLD, Ulrich ; LEVI, Paul: *Realzeitsysteme zur Prozeßautomatisierung*. Hanser Fachbuchverlag, 1994
- [73] RESEARCH, Point G.: *www.ptgrey.com*. Point Grey Research
- [74] RESEARCH, Point G.: *Bumblebee 2*. Point Grey Research, 2006
- [75] RIEGL: *Entfernungsmessgerät Distanzsensor Lasertape FG21-HA Bedienungsanleitung*. Version 7/01 CE, Rev. 19-07-2001. Horn, Austria: Riegl Laser Measurement Systems GmbH, July 2001
- [76] RYOO, SHANE: *Programm Optimization Strategies For Data-Parallel Many-Core Processors*. Urbana, Illinois, University of Illinois, Dissertation, 2008
- [77] SAITO, A.: Silhouette Vision™: New Vehicle Detection Field Proven Robust and Accurate. In: *Intelligent Transportation Symposium*, 1999, S. 48–54
- [78] SANAYOSHI, K. ; HANAWA, K. ; KISE, K. ; SOGAWA, Y.: 3-D Image Recognition System for Assist, Proceedings of the Intelligent Vehicles 1993 Symposium. In: *Proceedings of the Intelligent Vehicles 1993 Symposium (1993)*, S. 60–64
- [79] SCHENK, Toni: *Digital Photogrammetry*. Bd. 1. 1. TerraScience, 1999
- [80] SILC, Jurij ; ROBIC, Borut ; UNGERER, Theo: *Processor Architecture - From Dataflow to Superscalar and Beyond*. Berlin Heidelberg : Springer Verlag, 1999. – ISBN 3-540-64768-8
- [81] SUPPES, A.: *Stochastische Hindernisdetektion aus stereoskopischen Videosequenzen fuer fahrerlose Transportfahrzeuge*. VDI Verlag, 2004
- [82] SYSTEMS, IBM ; GROUP, Technology: IBM Blade Center QS22, PowerXCell 8i / IBM Corporation. Somers, NewYork, USA, October 2008 ( BLD03019-USEN-03). – Forschungsbericht



- [83] TAKAHASHI, H. ; TOMITA, F.: Self Calibration of Stereo Cameras. In: *Proc. 2nd Int. Conf. on Computer Vision*. Tarpon Springs, FL, 1988, S. 123–128
- [84] TEXAS INSTRUMENTS, Inc.: *TMS320C6418 Fixed-Point Digital Signal Processor*. No. SPRS241D August 2004 Rev. January 2006. Dallas, Texas, USA: Texas Instruments, Inc., January 2006
- [85] TIRUMALAI, A.P. ; SCHUNCK, B.G. ; JAIN, R.C.: Dynamic stereo with self-calibration. In: *Third International Conference on Computer Vision* (1990), S. 466–470
- [86] TISCHER, Michael ; JENNRICH, Bruno: *PC intern 5 : Systemprogrammierung*. Düsseldorf : Data-Becker, 1995, 1995
- [87] TORNOW, M. ; MICHAELIS, B. ; KUHN, R.W. ; CALOW, R. ; MECKE, R.: Hierarchical Method for Stereophotogrammetric Multi-objekt-Position Measurement. In: *Pattern Recognition, DAGM Symposium* (2003), S. 164–171
- [88] TORNOW, Michael: *Implementierung von 3D-Messalgorithmen*, Otto-von-Guericke Universität Magdeburg, Diplomarbeit, 2002
- [89] TORNOW, Michael ; KASZUBIAK, Jens ; KUHN, Robert W. ; MICHAELIS, Bernd ; SCHINDLER, Thomas: Hardware Approach for Realtime machine stereo vision. In: *Journal of Systemics Cybernetics and Informatics* Vol. 4 (2006), Nr. 24-36
- [90] TORNOW, Michael ; MICHAELIS, Bernd. *Verfahren zur photogrammetrischen Abstands- und/oder Positionsbestimmung*. Patentschrift DE 103 10 849 B4. Januar 2009
- [91] TYRREL, James: Low-cost sensor puts 3D-cameras in the picture. In: *Opto and Laser Europe* The European magazine for photonics professionals December 2004 (2004), Nr. 123, S. 20–21
- [92] UHLER, W. ; MATHONY, H.-J. ; KNOLL, P.M.: Driver Assistance Systems for Safety and Comfort / Robert Bosch GmbH, Driver Assistance Systems. Leonberg, 2003. – EU-Projekt EDEL im 5. Rahmenprogramm, edel-eu.org
- [93] ULMER, B.: VITA – An autonomous road vehicle (ARV) for collision avoidance in traffic. In: *Proceedings of the International Conference on Intelligent Vehicles*. Detroit, USA : IEEE Industrial Electronics Society, June 1992, S. 36–41
- [94] ULMER, B.: Autonomous Automated Driving in real Traffic. In: *Proceedings of the First World Congress on Applications of Transport, Telematics and Intelligent Vehicle Highway Systems*, 1994, S. 36–41
- [95] UNGERER, T. ; ZEHENDER, E.: A Parallel Programming Language Directed Towards Top-Down Software Development. In: *Proceedings of the International Conference on Parallel Processing*. St. Charles Illinois, Aug 1989, S. 122–125
- [96] UNGERER, Theo (Hrsg.): *Datenflussrechner*. Stuttgart : B.G.Teubner, 1993

## LITERATURVERZEICHNIS

- [97] UNSER, Michael: An improved least squares Laplacian pyramid for image compression. In: *Signal Process.* 27 (1992), Nr. 2, S. 187–203. – ISSN 0165–1684
- [98] WADA, T. ; UKIDA, H. ; MATSUYAMA, T.: Shape from shading with interreflections under a proximal light source. In: *Int. Journal Computer Vision* 24 (1997), Nr. 2, S. 125–135
- [99] WOLFSPERGER, Hans A.: *Elektromagnetische Schirmung: Theorie und Praxisbeispiele.* Berlin : Springer Verlag, 2008 ( ISBN 3540769129, 9783540769125)
- [100] WROBEL, B.P.: The evolution of digital photogrammetry from analytical photogrammetry. In: *Photogrammetric Record* 13 (1991), April, Nr. 77, S. 765–776
- [101] ZABIH, R. ; WOODFILL, J.: A variable window approach to early vision. In: *Proceedings of the European Conferenz of Computer Vision 94*, 1994, S. 151–158
- [102] ZITNICK, C. L. ; KANADE, Takeo: A volumetric iterative approach to stereo matching and occlusion detection / The Robotics Institue, Carnegie Mellon University. Pittsburg, Pennsylvania, Dec 1998 ( CMU-RI-TR-98-30). – Forschungsbericht
- [103] ZITNICK, C. L. ; KANADE, Takeo ; GOVERNMENT, States: A cooperative algorithm for stereo matching and occlusion detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), S. 675–684