

Julius-Maximilians-Universität Würzburg

Fakultät für Mathematik und Informatik

Informationstechnik für Luft- und Raumfahrt

Lehrstuhl für Informatik 8

Prof. Dr. Sergio Montenegro



Bachelorarbeit

Stereo-Optische Abstandsmessung für einen autonomen Quadrocopter

Vorgelegt von

Helge Andreas Lauterbach

Matr.-Nr.: 1740655

Prüfer: Prof. Dr. Sergio Montenegro

Betreuender wissenschaftlicher Mitarbeiter: Dipl.-Ing. Nils Gageik

Würzburg, 09.09.2013

Erklärung

Ich versichere, dass ich die vorliegende Arbeit einschließlich aller beigefügter Materialien selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Werken entnommen sind, sind in jedem Einzelfall unter Angabe der Quelle deutlich als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form noch nicht als Prüfungsarbeit eingereicht worden.

Mir ist bekannt, dass Zuwiderhandlungen gegen diese Erklärung und bewusste Täuschungen die Benotung der Arbeit mit der Note 5.0 zur Folge haben kann.

Würzburg, 09.09.2013

Helge Andreas Lauterbach

Thema

Die Fortschritte im Bereich Sensorik und Mikrotechnik ermöglichen heutzutage den kostengünstigen Bau kleiner unbemannter Luftfahrzeuge (UAV, unmanned aerial vehicle, Drohne) wie Quadrocopter sowie deren Ausstattung mit einer Vielzahl an Sensorik. Kameras sind heute für wenige Euros zu kaufen und stellen eine sehr mächtige Sensorik dar. Während es Systeme für die ferngesteuerte oder GPS-gestützte Verwendung schon zu kaufen gibt, ist die Forschung im Bereich autonomer Systeme noch in den Anfängen. Insbesondere der robuste autonome Betrieb, der eine kollisionsfreie Operation in unterschiedlichen Umgebungen ermöglicht, ist problematisch. Daher wird der Aufbau eines eigenen autonomen Systems am Lehrstuhl Aerospace Information Technology der Uni Würzburg erforscht und erprobt. Im Rahmen dieses Forschungsvorhabens ist ein System zu entwickeln, welches unter Verwendung von zwei Kamerabildern (stereo-optisch) eine Entfernungsmessung ausführt. Diese Entfernungsmessung soll verwendet werden, um die Hinderniserkennung mit Hilfe weiterer redundanter Sensorinformationen robuster zu gestalten.

Hauptaugenmerk dieser Arbeit ist die Bestimmung der Entfernung von Objekten durch die Bilder zweier Kameras und Korrelation der Bildinformationen (Bildverarbeitung). Dazu gehört die Realisierung und Parametrisierung einer Lösung zum Einsatz im Quadrocopter. Dabei soll auf bestehende Lösungen aufgebaut werden. Eine ausführliche Übersicht über den Stand der Technik ist erforderlich. Die entwickelte Software ist in das bestehende System einzubinden. Zur Aufgabe gehört sowohl die Kalibrierung des Systems als auch die Implementierung einer fortlaufenden Entfernungsmessung sowie die Repräsentation der Daten in einem geeigneten Format. Eine umfangreiche Evaluierung und eine ausführliche Dokumentation sind durchzuführen. Besonderes Merkmal des Themas ist, dass die Laufzeit ein kritischer Faktor ist, da nur begrenzt Rechenzeit zur Verfügung steht. Darüber hinaus sollte eine Angabe über die Verlässlichkeit der Abstandsrechnungen gemacht werden, da die Lösung zur Kollisionsvermeidung Anwendung findet.

Aufgabenstellung (Stichpunktartig):

- Stand der Technik: Abstandsmessung mit stereo-optischen Kamerasystemen
- Algorithmus und Konzept ausführlich erläutern
- Implementierung Abstandsmessung in QT mit zwei Web-Cams
- Anbindung an Quadrocopter Software
- Evaluierung (u.a.: Untersuchung/Optimierung von Laufzeit und Verlässlichkeit)
- Dokumentation

Zusammenfassung

Im Rahmen dieser Arbeit soll ein System zur optischen Entfernungsmessung für den Einsatz in einem autonomen Quadrocopter aufgebaut werden. Dabei soll es als Ergänzung bereits bestehender Sensorsysteme wie Infrarot und Ultraschall zum Einsatz kommen. Ziel ist es, dass dieses System später zur Hindernisvermeidung verwendet werden kann.

Als Grundlage für die Entfernungsmessung kommt der stereo-optische Ansatz zur Anwendung. Als Sensoren dienen zwei Webcams, welche nebeneinander auf einer Halterung an der Frontseite des Quadrocopters montiert sind. Dadurch betrachten die Kameras gleiche Objekte aus leicht unterschiedlicher Perspektive. Aus der Verschiebung zwischen den beiden Bildern wird per Triangulation die Tiefenentfernung eines Objektes bestimmt. Dazu werden Algorithmen verwendet, welche miteinander korrespondierende Bildpunkte zwischen den beiden Bildern findet und deren Verschiebung ermittelt. Die entstehende Tiefenkarte wird im letzten Schritt auf mögliche Hindernisse untersucht. Die Kalibrierung des Systems erfolgt über die Aufnahme von Schachbrettern mit unterschiedlicher Orientierung zum Ermitteln der Kamera-Eigenschaften und Geometrischen Zusammenhänge zwischen den beiden Kameras. Bei der Implementierung des Softwaremoduls kommen größtenteils Algorithmen der OpenCV Bibliothek zur Anwendung.

Das System wurde in Bezug auf Genauigkeit und Verlässlichkeit evaluiert. Dabei wurden die Parameter der beiden Stereo-Algorithmen der OpenCV Bibliothek optimiert, um einen guten Kompromis zwischen Genauigkeit und erforderlicher Rechenzeit zu finden. Hauptsächlich geschah die Evaluation unter Laborbedingungen. Ein abschließender Test des Systems erfolgte im Flug.

Die Evaluation zeigt, dass das aufgebaute System Entfernungen zwar mit akzeptabler Genauigkeit messen kann. Die Robustheit des Systems reicht aber für einen zuverlässigen Einsatz im Flug noch nicht aus.

Inhaltsverzeichnis

1. Einleitung	1
2. Grundlagen	2
2.1. Eigenschaften stereo-optischer Entfernungsmessung	2
2.1.1. Vorteile	2
2.1.2. Nachteile	2
2.2. Kameramodell	3
2.2.1. Lochkameramodell	3
2.2.2. Principal Point	4
2.2.3. Linsendistorsion	5
2.2.4. Stereokameramodell	6
2.3. Epipolargeometrie	7
2.4. Rektifizierung	8
2.4.1. Algorithmus von Bouguet	8
2.4.2. weitere Algorithmen zur Rektifizierung	9
2.5. Triangulation	9
3. Stand der Technik	13
3.1. Kalibrierung	13
3.1.1. Kalibrierung einer einzelnen Kamera	13
3.1.1.1. Ablauf eines Kalibrierungsvorganges	13
3.1.1.2. Analyse der Schachbretter	14
3.1.1.3. Lösung der Kamera Kalibrierung	14
3.1.1.4. Anzahl notwendiger Schachbrettaufnahmen	15
3.1.2. Stereo Kalibrierung	16
3.2. Stereokorrespondenzanalyse	16
3.2.1. Klassifizierung von Matching-Algorithmen	17
3.2.2. Typischer Aufbau korrelationsbasierter Matching-Algorithmen	17
3.2.3. Suche nach Korrespondenzen durch lokale Methoden	18
3.2.4. Block-Matching	18
3.2.5. Semi Global Block Matching	20
3.3. OpenCV	21
4. Konzept	23
4.1. StereoVision System	23
4.2. Kalibrierung	24
4.3. Matching	25
5. Implementierung	27

5.1. Software	27
5.1.1. StereoController	29
5.1.2. Kalibrierung	30
5.1.3. Matching	32
5.1.4. GUI/StereoWidget	35
5.2. Hardware - Kameras	37
6. Evaluierung	39
6.1. Überblick	39
6.2. Robustheit der Tiefenauswertung	39
6.2.1. Robustheit der ursprünglich verwendeten Methode	39
6.2.2. Vergleich der ursprünglichen und der korrigierten Methode	40
6.3. Erreichbare Genauigkeit der Stereo-Algorithmen	41
6.4. Genauigkeit in Abhängigkeit von der Auflösung	42
6.5. Auswirkung von Lichtverhältnissen	43
6.5.1. Versuchsaufbau	43
6.5.2. Ergebnisse	43
6.6. Auswirkung der Parameter der Stereo-Algorithmen und deren Optimierung	44
6.6.1. <i>NumberOfDisparities</i>	44
6.6.2. <i>MinDisparities</i>	46
6.6.3. <i>UniquenessRatio</i>	46
6.6.4. <i>SADWindow</i>	46
6.7. Zusammenfassung der Ergebnisse	47
6.8. Flug mit dem Quadrocopter	47
7. Diskussion und Ausblick	48
7.1. Diskussion der Ergebnisse	48
7.2. Ausblick	48
8. Literaturverzeichnis	50
A. Messergebnisse	52
A.1. Robustheit der Tiefenauswertung	52
A.2. Vergleich SGBM und BM	56
A.3. Genauigkeit in Abhängigkeit der Auflösung	57
A.4. Auswirkung unterschiedlicher Lichtverhältnisse	57
A.5. Auswirkung von Parametern der Stereo-Algorithmen	60
A.5.1. <i>NumberOfDisparities</i>	60
A.5.2. <i>UniquenessRatio</i>	61
A.5.3. <i>SADWindow</i>	62
B. Technische Daten	63
B.1. Kameras	63
B.2. Notebook	63

Abbildungsverzeichnis

2.1.	Modell einer Lochkamera	3
2.2.	mathematisch vereinfachtes Modell einer Lochkamera	4
2.3.	Auswirkung der tangentialen Distorsion	5
2.4.	Auswirkung der tangentialen Distorsion	6
2.5.	allgemeines Stereokameramodell (a), Standardstereogeometrie (b)	6
2.6.	Epipolargeometrie	7
2.7.	Rektifizierung nach Bouguet: a) nicht rektifizierte Kameraandordnung, b) parallele optische Achsen nach der ersten Rotation, c) rektifizierte Kameraanordnung mit parallelen optischen Achsen	8
2.8.	Geometrische Zusammenhänge der Triangulation (a) Erläuterung der Differenz $T - d$ (b)	10
3.1.	Beispiel Kalibrierungsobjekt mit einigen Objektkoordinaten	14
3.2.	Korrespondenzanalyse	18
3.3.	Beispiel für Pfade zum Erreichen eines Pixels p mit Disparität d	21
3.4.	OpenCV 2.4.0 Modulübersicht	22
4.1.	Konzept des Gesamtsystems	23
4.2.	Konzept der Kalibrierung	24
4.3.	Ablauf des Matching	25
5.1.	Überblick über die Module der Implementierung	27
5.2.	UML-Diagramm des Stereo-Vision Systems. public Methoden und wichtige private Methoden	28
5.3.	Übersicht über wichtige Funktionen des Stereo-Vision Systems zur Kalibrierung	30
5.4.	Übersicht über wichtige Funktionen des Stereo-Vision Systems zur Kalibrierung	32
5.5.	Beispiel für Konfiguration der Aufteilung der Tiefenkarte	34
5.6.	Übersicht über die grafische Oberfläche des Stereo-Vision Systems	35
5.7.	Übersicht über die weitere Reiter der Oberfläche des Stereo-Vision Systems	36
5.8.	Am Quadrocopter montierte Logitech C270 Webcams	38
6.1.	Versuchsaufbau zur Messung der Robustheit	39
6.2.	Versuchsaufbau aus Sicht der linken Kamera	40
6.3.	zur Tiefenkarte gehörende Bildausschnitte mit Einteilung in die Segmente in Abhängigkeit der Auflösung	42
6.4.	Flächen mit erkannter, kürzester Tiefenentfernung und deren Referenztiefe	43
6.5.	Auswirkung von <i>NumberOfDisparities</i> auf die Tiefenkarte	45
A.1.	Robustheit der Tiefenauswertung mit ursprünglicher Methode und SGBM	53
A.2.	Robustheit der Tiefenauswertung mit ursprünglicher Methode und BM	54

A.3. Vergleich der Tiefenauswertung mit ursprünglicher Methode und neuer Methode	55
A.4. Vergleich der Genauigkeit von BM und SGBM	56
A.5. Messreihe mit SGBM unter verschiedenen Lichtverhältnissen	58
A.6. Messreihe mit BM unter verschiedenen Lichtverhältnissen	58
A.7. Laufzeit der Stereo-Algorithmen in Abhängigkeit von <i>NumberOfDisparities</i> . . .	60
A.8. minimal messbare Entfernung in Abhängigkeit von <i>NumberOfDisparities</i>	60
A.9. Linkes Kamerabild zu den folgenden Tiefenbildern	61
A.10. Auswirkung der <i>UniquenessRatio</i> auf die Tiefenkarte beim SGBM	61
A.11. Auswirkung der <i>UniquenessRatio</i> auf die Tiefenkarte beim BM	61
A.12. Auswirkung des <i>SADWindow</i> auf die Tiefenkarte beim BM	62
A.13. Auswirkung des <i>SADWindow</i> auf die Tiefenkarte beim SGBM	62

Tabellenverzeichnis

2.1. Beispiel für die Tiefe Z und den Fehler ΔZ bei einer Auflösung der Disparität d von 1 px	11
2.2. Verbesserung der Tiefenauflösung bei einer Auflösung der Disparität d von 0.1 px	11
A.1. Vergleich der Tiefenauswertung mit ursprünglicher Methode und neuer Methode	55
A.2. Vergleich der Genauigkeit der beiden Stereo-Algorithmen	56
A.3. Tiefenmessung einer Szene mit Auflösung von 320 px	57
A.4. Tiefenmessung einer Szene mit Auflösung von 192 px	57
A.5. Entfernungsmessung mit dem SGBM unter verschiedenen Beleuchtungssituationen	58
A.6. Entfernungsmessung mit dem BM unter verschiedenen Beleuchtungssituationen	59
A.7. Auswirkungen des Parameters <i>NumberOfDisparities</i>	60
B.1. technische Daten der verwendeten Kameras [Logitech, 2010a] [Logitech, 2010b]	63

1. Einleitung

Spätestens seit der Affäre um den Ausstieg der Bundeswehr aus dem EuroHawk Projekt, sind Drohnen in aller Munde. Dabei wird der Begriff Drohne häufig mit Überwachung und militärischen Operationen assoziiert. Dass auch zivile Einsatzmöglichkeiten für unbemannte Luftfahrzeuge existieren, zeigt das Forschungsprojekt AQopterI8 des Lehrstuhl für Informatik VIII der Universität Würzburg. Ziel ist es, einen Quadrocopter zu entwickeln, welcher in der Lage ist: „... autonome Aufgaben aus den Bereichen Search & Rescue (z.B: Feuerwehreinsatz) sowie Überwachung und Kontrolle (Industrie- & Chemieanlage) durchzuführen.“ [AQopterI8, 2013]

Der Quadrocopter ist bereits mit Sensorsystemen zur Entfernungsmessung wie Ultraschall und Infrarot ausgestattet. Jedes dieser Systeme hat sowohl Vor- als auch Nachteile. Ziel dieser Arbeit ist es, ein System zur optischen Entfernungsmessung aufzubauen und den Einsatz in einem autonomen Quadrocopter zu erproben. Damit sollen die bestehenden Sensorsysteme in der Hindernisvermeidung unterstützt und ergänzt werden. Dabei kommen Kameras aus dem Low-Cost Bereich zum Einsatz.

Ermöglicht durch die Verfügbarkeit kleiner und leistungsstarker Computer finden Systeme zur stereo-optischen Entfernungsmessung heutzutage eine breite Verwendung. Das beginnt beispielsweise bei Anwendungen in der Robotik zur Navigation und Pfadplanung und geht über den Einsatz in Fahrzeugen, zur Messung des Abstandes zu vorausfahrenden Fahrzeugen, bis zur Verwendung als Eingabegerät für Spielekonsolen (z.B. Microsoft Kinect).

Der Mensch ist durch seine Augen bestens mit dem Stereo-Sehen vertraut. Dabei rekonstruiert das Gehirn aus den beiden zweidimensionalen Bildern ein dreidimensionales Bild der Umgebung. Ähnlich wird bei der Tiefenmessung durch ein Stereokamerassystem vorgegangen. Die aufgenommenen Kamerabilder werden auf Korrespondenzen hin untersucht. Aus der Verschiebung eines abgebildeten Objektes zwischen den beiden Bildern einer Szene kann mit Hilfe von Triangulation eine 3D-Rekonstruktion durchgeführt werden. Die Berechnung der Tiefeentfernung zu einem Objekt durch ein Stereokamerasystem ist dabei generell mit einem hohen Rechenaufwand verbunden. Trotz der Verfügbarkeit schneller Algorithmen ist eine Optimierung des Stereokamerasystems z.B. in Bezug auf Bildauflösung notwendig um mit den begrenzten Ressourcen des Quadrocopters zurecht zu kommen und die Entfernungsmessung in Echtzeit durchführen zu können.

2. Grundlagen

In diesem Kapitel werden zunächst einige Eigenschaften eines Stereo Vision Systems diskutiert. Anschließend wird der grundsätzliche Aufbau eines Stereo Vision Systems beschrieben. Ausgehend vom Lochkameramodell und notwendigen Modifikation welche z.B. der Einsatz von Linsen mit sich bringt, wird erläutert wie durch Epipolarometrie und Triangulation die Tiefenentfernung eines abgebildeten Objektes berechnet werden kann.

2.1. Eigenschaften stereo-optischer Entfernungsmessung

Beim stereo-optischen Verfahren handelt es sich um passives Verfahren zur Tiefenmessung. Dabei wird die Entfernung eines Objektes per Triangulation (siehe 2.5) ermittelt.

2.1.1. Vorteile

Ein großer Vorteil von optischen Sensoren wie Kameras ist, dass sie keinen Drift aufweisen. Die Genauigkeit der Entfernungsmessung ändert sich dadurch auch über einen langen Zeitraum nicht, sofern keine anderen äußeren Störeinflüsse vorhanden sind.

Zudem wird die Umgebung nicht nur punktuell erfasst, sondern großflächig abgebildet. Gegenüber anderen Sensorsystemen bietet das den Vorteil, dass nur zwei Sensoren (Kameras) zur Abdeckung eines großen Bereiches notwendig sind.

Zudem bietet ein Stereo-Vision System die Möglichkeit Bestimmung der relativen 3D-Position eines abgebildeten Objektes, im Gegensatz zur reinen Entfernungsmessung durch Infrarot und Ultraschallsensoren.

2.1.2. Nachteile

Systeme zur optischen Entfernungsmessung können zwar eine hohe Genauigkeit erreichen. Die Tiefenauflösung ist aber nur im nahen Umfeld hoch. (Siehe dazu Kapitel 2.5)

Die Berechnung von Korrespondenzen zwischen zwei Bilder geht immer mit einem hohen Berechnungsaufwand einher. In Abhängigkeit von gewählter Auflösung kann dabei schnell eine Laufzeit des Algorithmus entstehen, welche die Anforderungen an einen Echtzeitbetrieb nicht erfüllen kann.

Als passives Sensorsystem sind Kamera von den herrschenden Lichtverhältnissen abhängig. Insbesondere extreme Lichtverhältnisse wie starke Sonneneinstrahlung oder Dunkelheit können das Messergebnis stark beeinflussen bis unbrauchbar machen.

Probleme bei der Suche nach Übereinstimmungen durch einen Stereo-Algorithmus treten hauptsächlich bei glatten, homogenen Flächen mit wenig Textur wie z.B. einfarbige Wände auf,

da eine eindeutige Zuordnung zusammengehöriger Bildpunkte schwer bis unmöglich ist. Ähnlich verhält es sich bei sich häufig wiederholenden Muster, wie z.B. einem Netz.

2.2. Kameramodell

2.2.1. Lochkameramodell

Die Abbildungsgeometrie lässt sich am einfachsten anhand einer Lochkamera erläutern.

Bildgebend ist eine infinitesimal kleine und dünne Lochblende, welche genau einen Lichtstrahl passieren lässt, welcher entlang der Verbindungslinie zwischen einem Punkt auf dem abgebildeten Objekt und der Lochblende verläuft. Dadurch entsteht auf der Bildebene ein Bild, welches immer im Fokus ist und eine unendliche Schärfentiefe aufweist. Es erscheint auf der Bildebene horizontal und vertikal gespiegelt.

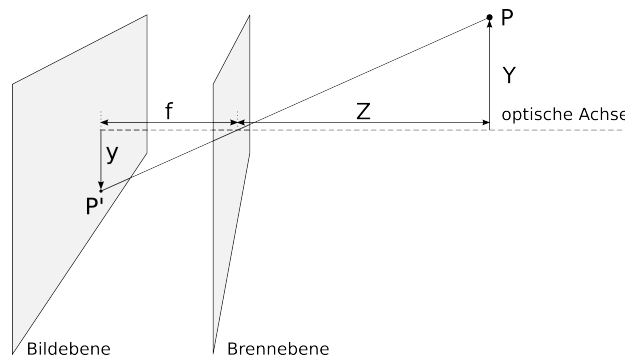


Abbildung 2.1.: Modell einer Lochkamera

Mathematisch lässt sich die Abbildung wie folgt ausdrücken

$$\begin{aligned} x_{Bild} &= -\frac{f}{Z} X_{Obj} \\ y_{Bild} &= -\frac{f}{Z} Y_{Obj} \end{aligned} \quad (2.1)$$

Dabei fällt auf, dass die Bildkoordinaten nur Verhältnisse der Objektkoordinaten enthalten, aus welchen sich weder die Entfernung eines Gegenstandes, noch seine Größe bestimmen lässt. Alle 3D-Punkte in Objektkoordinaten, die auf der gleichen Projektionsgeraden liegen werden auf denselben Bildpunkt abgebildet.

Das mathematische Modell der Lochkamera lässt sich vereinfachen, indem die Bildebene an der Lochblende punktgespiegelt wird. Die neue Bildebene befindet sich so nun aber zwischen Lochblende und abgebildetem Objekt. Die Lochblende wird zum Projektionszentrum. Dieses modifizierte Modell entspricht zwar nicht mehr dem physikalischen Modell ist aber mathematisch äquivalent. Es vereinfacht die Betrachtungsweise und ist vor allem für die Stereogeometrie (siehe 2.2.4) nützlich.

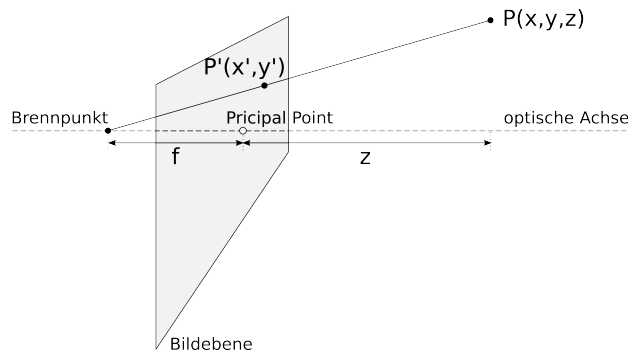


Abbildung 2.2.: mathematisch vereinfachtes Modell einer Lochkamera

$$\begin{aligned} x_{Bild} &= \frac{f}{Z} X_{Obj} \\ y_{Bild} &= \frac{f}{Z} Y_{Obj} \end{aligned} \quad (2.2)$$

Ein Lochkamera ist aus zwei Gründen nicht praxistauglich. Zum einen ist eine infinitesimal kleine Lochblende technisch nicht realisierbar. Hat die Lochblende eine Ausdehnung, führt das zu unscharfen Bildern. Ein viel gravierender Nachteil ist aber, dass die geringe Lichtmenge, welche die Lochblende passieren kann, sehr lange Verschlusszeiten erfordert, da es keine Filme oder Sensoren mit unendlich hoher Lichtempfindlichkeit gibt. Lange Verschlusszeiten sind für Echtzeitanwendungen wie z.B. die stereo-optische Entfernungsmessung inakzeptabel.

Aus diesem Grund werden statt der Lochblende Linsen eingesetzt, welche die Lichtstrahlen, die von einem Objekt ausgehen, bündeln und in einem Punkt auf der Bildebene abbilden.

Die Linsengleichung lautet:

$$\frac{1}{f} = \frac{1}{b} + \frac{1}{g} \quad (2.3)$$

Das Lochkameramodell kann aber weiterhin als Beschreibung verwendet werden. Unter der Annahme, dass die Gegenstandsweite g sehr viel größer als die Bildweite b ist, kann man davon ausgehen, dass sich die Bildebene in der Brennebene befindet. Zudem kann man sich der Einfachheit halber auf die Betrachtung des Projektionsstrahls durch das Zentrum der Linse beschränken, eine perfekte Linse vorausgesetzt.

2.2.2. Principal Point

Der sogenannte *Principal Point* befindet sich im Schnittpunkt der optischen Achse der Kamera mit der Bildebene. In der Theorie stimmt dieser Punkt mit dem Mittelpunkt des Sensors überein. In der Realität weicht dieser einige Pixel von der theoretischen Position ab, da der Sensor bei der Kameraproduktion nicht genau mit der optischen Achse ausgerichtet werden kann.

Das Lochkameramodell muss also noch um die beiden Parameter c_x und c_y erweitert werden, welche die Verschiebung des Sensormittelpunktes in Bezug auf die optische Achse modellieren.

$$\begin{aligned}x_{Bild} &= \frac{f}{Z} X_{Obj} + c_x \\y_{Bild} &= \frac{f}{Z} Y_{Obj} + c_y\end{aligned}\quad (2.4)$$

2.2.3. Linsendistorsion

In der Realität ist keine Linse perfekt. Das führt zu einer Reihe unterschiedlicher Abbildungsfehler. Bedeutend sind vor allem Verzeichnungen oder Distorsionen.

Zwei unterschiedliche Arten von Distorsionen werden durch Funktionen von OpenCV berücksichtigt:

1. radiale Distorsion entsteht durch die Form der Linse. Ausgeprägt ist sie vor allem bei Weitwinkel- und Fish-Eye-Objektiven, also Objektiven mit sehr kurzer Brennweite. Ebenso sind sie bei minderwertigen Linsen bemerkbar. Dabei hängt die Stärke der Verzeichnung vom Abstand zur optischen Achse ab. In der Nähe der Achse ist sie praktisch nicht vorhanden, in den Bildecken ist sie am stärksten.

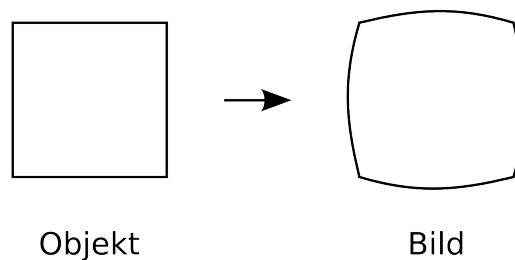


Abbildung 2.3.: Auswirkung der tangentialen Distorsion

Abbildung 2.3 verdeutlicht den Effekt radialer Distorsion: Gerade Linien werden verbogen.

Bei handelsüblichen Digitalkameras und mittlerweile auch bei Webcams ist die radiale Distorsion nur geringfügig oder gar nicht bemerkbar. Heutzutage wird bei der Herstellung von Linsen viel Aufwand betrieben, sie so gering wie möglich zu halten.

In der Praxis ist die radiale Distorsion klein und kann durch die ersten Terme einer Taylorentwicklung um $r = 0$ charakterisiert werden. [Bradski und Kaehler, 2008] Im Normalfall reichen die ersten beiden Terme k_1 und k_2 aus. Bei starker radialer Distorsion kann der dritte Term k_3 dazugenommen werden. Die Korrektur der Bildkoordinaten erfolgt durch:

$$\begin{aligned}x_{kor} &= x_{orig} (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\y_{kor} &= y_{orig} (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)\end{aligned}\quad (2.5)$$

2. tangentiale Distorsion entsteht, wenn die Linse und der Bildsensor nicht exakt koplanar angeordnet sind. Das tritt vor allem dann auf, wenn der Sensor bei der Produktion geklebt wird. In höherwertigen Kameras ist der Effekt vernachlässigbar.

Sie kann durch zwei zusätzliche Parameter p_1 und p_2 korrigiert werden.

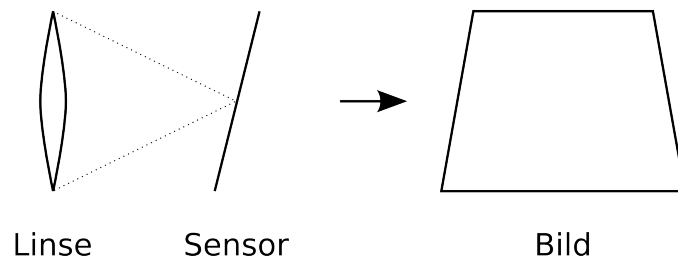


Abbildung 2.4.: Auswirkung der tangentialen Distorsion

$$\begin{aligned} x_{\text{kor}} &= x_{\text{orig}} + (2p_1 y_{\text{orig}} + p_2 (r^2 + 2x_{\text{orig}}^2)) \\ y_{\text{kor}} &= y_{\text{orig}} + (p_1 (r^2 + 2y_{\text{orig}}^2) + 2p_2 x_{\text{orig}}) \end{aligned} \quad (2.6)$$

Die Bestimmung der genannten fünf Distorsions-Koeffizienten erfolgt durch die Kalibrierung (siehe Kapitel 3.1)

2.2.4. Stereokameramodell

Aus dem Lochkameramodell ist ersichtlich, dass eine Kamera allein nicht ausreicht um die Position eines abgebildeten Objektes eindeutig zu bestimmen. (siehe (2.4)) Das wird erst durch die Hinzunahme einer zweiten Kamera möglich.

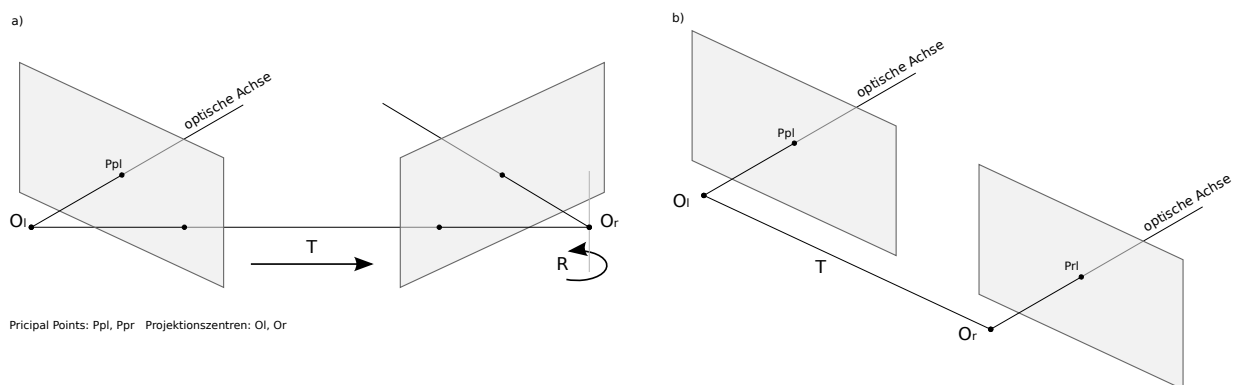


Abbildung 2.5.: allgemeines Stereokameramodell (a), Standardstereogeometrie (b)

Ein Stereokamerasystem ist wie in Abbildung 2.5 gezeigt aufgebaut. Die geometrischen Zusammenhänge zwischen den beiden Kameras werden durch die sogenannten extrinsischen Parameter beschrieben. Dabei handelt es sich um eine Rotationsmatrix R und einen Translationsvektor T , mit welchen die eine Kamera in die andere transformiert werden kann.

Für die Korrespondenzanalyse (siehe Kapitel 3.2) ist es von Vorteil, wenn es sich bei den Bildebenen der Kameras um Ausschnitte der selben Ebene handelt, welche lediglich um die x-Achse verschoben sind. Diese Kameraanordnung wird auch als Standardstereogeometrie bezeichnet. Da es in der Praxis schwer möglich ist, die Kameras so exakt auszurichten, dass sie die Standardstereogeometrie erfüllen, ist eine mathematische Korrektur, Rektifizierung (siehe 2.4) genannt notwendig.

Prinzipiell sind auch Kamera Anordnungen möglich, bei denen die Kameras wie in Abbildung 2.5 (a) zueinander ausgerichtet sind. Da in diesem Fall Stereokorrespondenzanalyse und Triangulation aufwändiger sind, wird im Regelfall die Standardstereogeometrie bevorzugt. Im Folgenden wird, wenn nicht explizit erwähnt, von einer Standardstereogeometrie ausgegangen.

2.3. Epipolargeometrie

Bei der Korrespondenzanalyse besteht grundsätzlich das Problem, dass sich der zu einem Punkt P_l im linken Bild gehörende Punkt P_r im rechten Bild zunächst an einer beliebigen Position befinden könnte. Durch die Suche in 2 Dimensionen würde ein hoher Rechenaufwand entstehen.

Eine wichtige Einschränkung des Suchbereichs stellt die Epipolargeometrie dar. Sie dient zur Beschreibung der geometrischen Beziehungen zwischen den Projektionen eines Objektes in die beiden Bildebenen der Kameras.

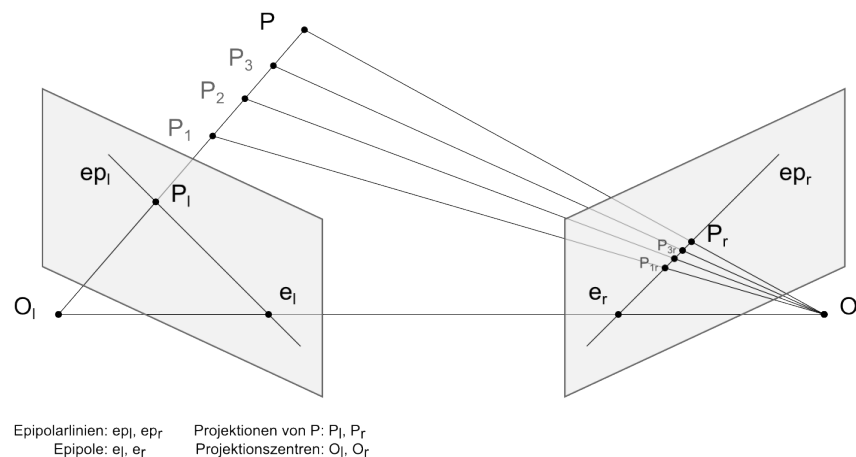


Abbildung 2.6.: Epipolargeometrie

Ausgegangen wird von einem Stereokamerasystem wie in Abbildung 2.6 gezeigt. Die Grafik veranschaulicht dabei die wichtigsten Eigenschaften der Epipolargeometrie.

Ein Epipol ist als die Projektion des Projektionszentrums der einen Kamera in die Bildebene der anderen Kamera definiert. In Abbildung 2.6 wird z.B. das Projektionszentrum der linken Kamera O_l auf die rechte Bildebene als Epipol e_r projiziert. Dabei kann sich der Epipol auch außerhalb des sichtbaren Bildbereiches befinden. Seine Position ist fix, solange die Anordnungen der Kameras nicht verändert wird.

Ein Punkt P , welcher in die Punkte P_l und P_r projiziert wird, spannt mit den beiden Projektionszentren der Kameras O_l und O_r eine Ebene auf, welche Epipolarebene genannt wird.

Die Schnittgerade der Epipolarebene mit der Bildebene einer Kamera wird als Epipolarlinie bezeichnet. Alle Epipolarlinien eines Kamerabildes schneiden sich im Epipol. In der Abbildung sind das die Linien ep_l und ep_r .

Aus dem Lochkameramodell (siehe 2.2.1) ist bekannt, dass sich ein in die Bildebene projektierter Punkt in Wirklichkeit überall auf der Verbindungsgeraden zwischen dem Projektionszentrum und dem Projektierten Punkt liegen kann. Alle diese möglichen Punkte werden auf die Epipolarlinie des anderen Kamerabildes projiziert.

Mit Hilfe der Eigenschaften der Epipolargeometrie lässt sich das Suchproblem der Stereokorrespondenzanalyse von einem zweidimensionalen Problem auf ein eindimensionales Problem reduziert werden. Der korrespondierende Punkt im rechten Bild muss nur auf der Epipolarlinie gesucht werden, welche zu dem Punkt des linken Bildes gehört.

2.4. Rektifizierung

Aus der Epipolargeometrie (Kapitel 2.3) ist bekannt, dass der korrespondierende Punkt P_r zu einem Punkt P_l im linken Bild auf der Epipolarlinie im rechten Bild zu finden ist. Durch die sogenannte Rektifizierung lässt sich das Suchproblem der Korrespondenzanalyse noch weiter vereinfachen.

Ziel ist es, die Kamerabilder so zu transformieren, dass alle Epipolarlinien horizontal, also parallel zu x-Achse verlaufen. Dadurch werden die Epipole ins unendliche verschoben und beide Kamerabilder liegen in derselben Ebene. Weiterhin ist es von Vorteil, wenn die Kamerabilder so angeordnet sind, dass sich korrespondierende Punkte in der jeweils gleichen Bildzeile befinden. Diese Transformation des Kameramodells ist Aufgabe der Rektifizierung.

2.4.1. Algorithmus von Bouguet

Jean-Yves Bouguet [Bouguet, 2010] [Bradski und Kaehler, 2008] hat einen Algorithmus entwickelt, welcher versucht, den notwendigen Reprojektionsaufwand und die dadurch entstehende Verzerrung der beiden Bilder zu minimieren und gleichzeitig die Überschneidung beider Bilder zu maximieren.

Als Voraussetzung gilt, dass das linke und rechte Bild bereits distortionsfrei, also entzerrt sind. Weiterhin werden die Rotationsmatrix R und der Translationsvektor T benötigt, mit welchen die rechte Kamera in die linke transformiert werden kann. Beide extrinsischen Parameter stammen aus der Stereokalibrierung (siehe Kapitel 3.1.2).

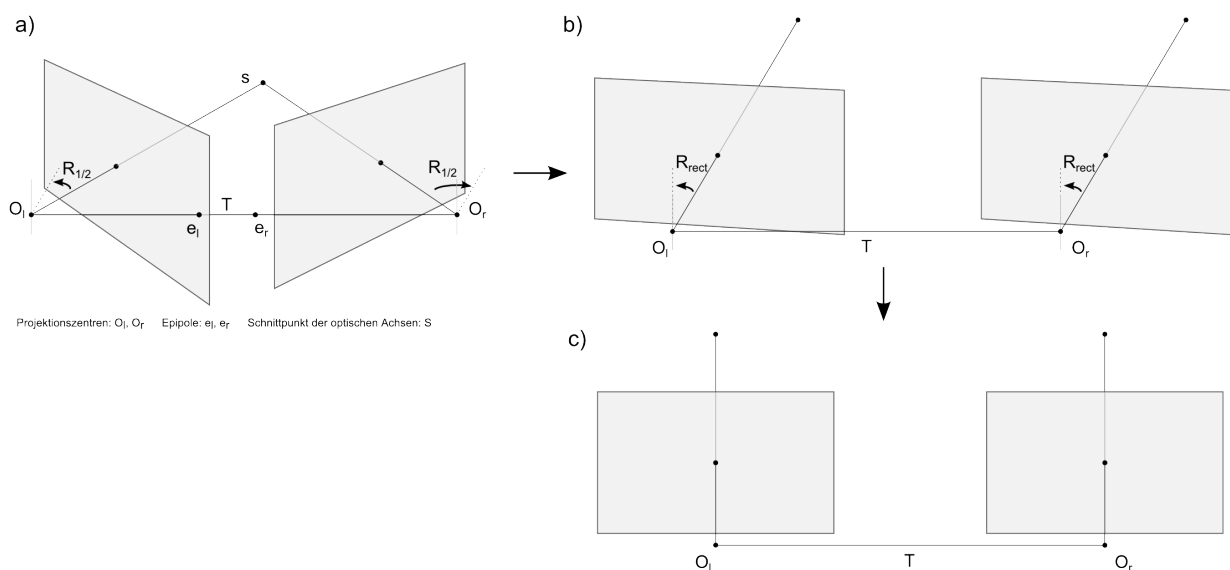


Abbildung 2.7.: Rektifizierung nach Bouguet: a) nicht rektifizierte Kameraanordnung, b) parallele optische Achsen nach der ersten Rotation, c) rektifizierte Kameraanordnung mit parallelen optischen Achsen

In Abbildung 2.7 ist der Ablauf der Rektifizierung schematisch dargestellt. Zunächst wird im ersten Schritt des Algorithmus jede Kamera um die halbe Rotationsmatrix R gedreht. Anschließend sind die Bildebenen beider Kameras koplanar. Die optischen Achsen verlaufen Parallel. Im zweiten Schritt erfolgt eine Rotation, die dafür sorgt, dass beide Bildebenen Teil derselben Ebene werden und die Zeilenanordnung so korrigiert wird, dass sich korrespondierende Punkte jeweils in der gleichen Bildzeile befinden. Die Epipole beider Bilder sind nun ins unendliche verschoben und die Kamerabilder rektifiziert.

Für die rektifizierten Bilder werden zudem noch zwei Projektionsmatrizen P_l und P_r berechnet, mit welchen sich ein 3D-Punkt in die linke oder rechte Kameraebene projizieren lässt. Im Zusammenhang mit der später beschriebenen Triangulation (siehe 2.5) ist aber vor allem die ebenso berechnete Reprojektionsmatrix Q wichtig, welche einen Punkt der linken Bildebene in 3D-Koordinaten konvertiert. Sie wird im Zusammenhang mit der Triangulation noch genauer betrachtet.

2.4.2. weitere Algorithmen zur Rektifizierung

Es gibt viele weitere Möglichkeiten um die für die Rektifizierung notwendigen Terme zu berechnen. Neben dem Algorithmus von Bouguet ist in OpenCV noch der Algorithmus von Hartley [Hartley, 1998][Bradski und Kaehler, 2008] implementiert. Dieser kann Bilder von nicht stereokalibrierten Kameras rektifizieren, erzeugt aber unter Umständen stärker verzerrte Bilder als der Algorithmus von Bouguet.

Der große Vorteil wäre, dass mit dem Algorithmus eine Online-Stereokalibrierung realisiert werden kann. Durch Beobachtung von Punkten in der abgebildeten Szene (Feature Detection und Matching) kann die Fundamentalmatrix zwischen beiden Bildern bestimmt werden. Mit deren Hilfe werden die Kamerabilder dann durch den Algorithmus von Hartley rektifiziert. Das würde einen eventuell aufwändigen Kalibrierungsvorgang überflüssig machen.

Ein sehr bedeutsamer Nachteil ist, dass mit dieser Methode die Größe eines Objektes nicht eindeutig bestimmt ist. Durch das Fehlen der intrinsischen Parameter werden z.B. unterschiedliche Brennweiten und optische Achsen der Kameras nicht berücksichtigt. Für die Entfernungsmessung ist der Algorithmus daher nicht geeignet.

2.5. Triangulation

Triangulation ist eine geometrische Methode der optischen Abstandsmessung. (vgl. [Wikipedia, 2008b]) Mit deren Hilfe wird aus der durch die Stereokorrespondenzanalyse (siehe Kapitel 3.2) ermittelten Disparität die Tiefenentfernung zwischen Kamera und Objekt berechnet. Bei der Disparität oder auch Parallaxe handelt es sich um die Verschiebung der korrespondierenden Punkte zwischen den beiden Bildern.

Abbildung 2.8 (a) zeigt die geometrischen Zusammenhänge der Triangulation. Der Abstand T zwischen den Projektionszentren O_l und O_r der beiden Kameras wird als Basislinie bezeichnet. Bei der Differenz $d = x_l - x_r$ handelt es sich um die durch die Korrespondenzanalyse berechnete Disparität.

Durch Betrachtung ähnlicher Dreiecke in Abbildung 2.8 lässt sich auf die Tiefenentfernung Z schließen. Wie dabei die Strecke $T - d$ zustande kommt ist in Abbildung 2.8 (b) dargestellt.

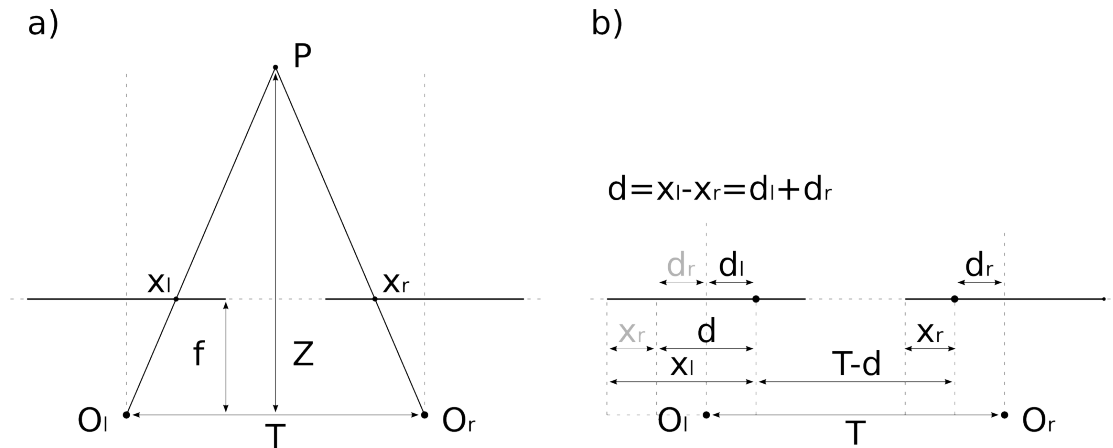


Abbildung 2.8.: Geometrische Zusammenhänge der Triangulation (a) Erläuterung der Differenz $T - d$ (b)

$$\frac{T}{Z} = \frac{T - d}{Z - f} \quad (2.7)$$

Nach Z aufgelöst ergibt sich

$$Z = \frac{fT}{d} \quad (2.8)$$

Reprojektionsmatrix Wie bereits in Kapitel 2.4 beschrieben, wird im Zuge der Rektifizierung die Reprojektionsmatrix Q berechnet. Sie erlaubt eine Konvertierung eines Bildpunktes der linken Kameraebene in einen 3D-Punkt. Sie hat für die Standardstereogeometrie folgenden Aufbau:

$$Q = \begin{pmatrix} 1 & 0 & 0 & -c_x^{left} \\ 0 & 1 & 0 & -c_y^{left} \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{1}{T_x} & 0 \end{pmatrix} \quad (2.9)$$

Bei f handelt es sich um die Brennweite, welche für das distortionsfreie, rektifizierte Bild der linken Kamera gilt, sie kann daher von der Brennweite der Kameramatrix abweichen. T_x ist der Abstand zwischen den beiden rektifizierten Kameras und gleich der x-Komponente des Translationsvektors. (siehe Kapitel 2.5)

Für die Reprojektion gilt:

$$Q \begin{pmatrix} x & y & d & 1 \end{pmatrix}^T = \begin{pmatrix} x' & y' & z' & w' \end{pmatrix}^T \quad (2.10)$$

$$\begin{pmatrix} x_{obj} & y_{obj} & z_{obj} \end{pmatrix}^T = \begin{pmatrix} \frac{x'}{w'} & \frac{y'}{w'} & \frac{z'}{w'} \end{pmatrix}^T \quad (2.11)$$

(vgl. [Bradski und Kaehler, 2008])

Durch ausmultiplizieren ergeben sich die Formeln (2.4) für x_{obj} und y_{obj} , sowie (2.8) für z_{obj} .

Tiefenauflösung Aus Formel (2.8) wird sofort die umgekehrt proportionale Abhängigkeit zwischen Tiefenentfernung und Disparität ersichtlich. Das hat zur Folge, dass kleine Änderungen bei Disparitäten nahe 0 große Auswirkungen auf die Tiefenentfernung haben. Eine hohe Tiefenauflösung wird daher nur im näheren Umfeld der Kameras erreicht.

Die Gaussche Fehlerrechnung erlaubt eine Einschätzung der zu erwarteten Tiefenauflösung. Unter der Annahme, dass die beiden Konstanten f und T des Stereokamerasystems fehlerfrei sind folgt für den Fehler von Z

$$\Delta Z = \sqrt{\frac{\delta Z}{\delta d}} \sigma_d = \frac{fT}{d^2} \sigma_d = \frac{Z^2}{fT} \sigma_d \quad (2.12)$$

Bei einer maximalen Auflösung der Disparität von 1 Pixel ergibt sich zum Beispiel eine Werteverteilung, wie sie in Tabelle 2.1 aufgeführt ist. Dafür wurde eine Basislinie $T = 10 \text{ cm}$ und eine Brennweite $f = 400 \text{ px}$ als Beispiel gewählt. Es fällt auf, dass die Genauigkeit bei kleinen Disparitäten sehr gering ist, bei gleichzeitig sehr großem Fehler. Beispielsweise können keine Entfernungen zwischen 438.8 cm und 493.6 cm aufgelöst werden. Der Wert 450 cm ist nicht messbar.

$d \text{ [px]}$	$Z \text{ [cm]}$	$\Delta Z \text{ [cm]}$	$d \text{ [px]}$	$Z \text{ [cm]}$	$\Delta Z \text{ [cm]}$
5	789.9	158.0	40	98.7	2.5
6	658.3	109.7	41	96.3	2.3
7	564.2	80.6	42	94.0	2.2
8	493.7	61.7	43	91.9	2.1
9	438.8	48.8	44	89.8	2.0
10	395.0	39.5	45	87.8	2.0
20	197.5	9.9	50	79.0	1.6
30	131.7	4.4	60	65.8	1.1

Tabelle 2.1.: Beispiel für die Tiefe Z und den Fehler ΔZ bei einer Auflösung der Disparität d von 1 px

Um dem Problem zu begegnen werden die Disparitäten bei der Korrespondenzanalyse in der Regel mit Subpixelgenauigkeit bestimmt. Tabelle 2.2 zeigt die Verbesserung, wenn mit einer Genauigkeit von 0.1 Pixel aufgelöst wird. Wie dies durch die Stereoalgorithmen erreicht wird, wird in Kapitel 3.2 gezeigt.

$d \text{ [px]}$	$Z \text{ [cm]}$	$\Delta Z \text{ [cm]}$	$d \text{ [px]}$	$Z \text{ [cm]}$	$\Delta Z \text{ [cm]}$
8.0	493.7	6.2	8.6	459.3	5.3
8.1	487.7	6.0	8.7	454.0	5.2
8.2	481.7	5.9	8.8	448.8	5.1
8.3	475.9	5.7	8.9	443.8	5.0
8.4	470.2	5.6	9.0	438.8	4.9
8.5	464.7	5.5	9.1	434.0	4.8

Tabelle 2.2.: Verbesserung der Tiefenauflösung bei einer Auflösung der Disparität d von 0.1 px

Eine weitere Möglichkeit, die Tiefenauflösung zu verbessern besteht darin, die Basislinie des Stereokamerasystems zu vergrößern. Dadurch steigt die Disparität bei gleicher Entfernung an.

Nachteilig wirkt sich dabei allerdings aus, dass die überlappenden Bereiche der Kamerabilder kleiner werden und somit die kleinste messbare Entfernung größer wird. Beim Design des Stereosystems ist ein Kompromiss zu finden, zwischen erforderlicher Tiefenauflösung und minimal messbarer Entfernung.

3. Stand der Technik

3.1. Kalibrierung

Die Kalibrierung hat folgende Aufgaben zu erledigen:

- Bestimmung der optischen Eigenschaften des Systems, die intrinsischen Parameter jeder Kamera
- mathematische Korrektur der Abweichungen vom Lochkameramodell, welche die Verwendung von Linsen mit sich bringt.
- Ermitteln des geometrischen Zusammenhanges zwischen den beiden Kameras des Stereosystems, die extrinsischen Parameter des Stereosystems
- Herstellen einer Verknüpfung zwischen der Kamera-Maßeinheit (Pixel) und den Maßeinheiten (z.B. cm) der realen 3D-Umgebung

3.1.1. Kalibrierung einer einzelnen Kamera

Die intrinsischen Parameter einer Kamera ließen sich zwar auch dem Datenblatt entnehmen, allerdings sind die Angaben dort in nahezu allen Fällen und gerade im Low-Cost Bereich zu ungenau oder unvollständig. Daher ist die Durchführung einer Kalibrierung der einzelnen Kamera notwendig.

Es gibt eine ganze Reihe von Ansätzen um das Problem der Kalibrierung zu lösen. Eine Möglichkeit ist z.B. , ein 3D-Objekt mit bekannten Abmessungen und festgelegten Kalibrierungspunkten in einer genau bekannten Lage und Position zu fotografieren. (z.B. ein Würfel mit Farbkreisen auf den Seitenflächen). Im Allgemeinen ist es schwierig, präzise 3D-Kalibrierungsobjekte herzustellen. Ein anderer Ansatz ist in OpenCV implementiert und wird im folgenden genauer beleuchtet. Er folgt im Wesentlichen den Arbeiten von Zhang [Zhang, 1999][Zhang, 2000] und Sturm [Sturm und Maybank, 1999].

3.1.1.1. Ablauf eines Kalibrierungsvorganges

Statt eines 3D-Objektes wird ein planares 2D-Kalibrierungsobjekt verwendet, welches viele eindeutig identifizierbare 3D-Punkte besitzt. Statt einer Aufnahme wird dieses Objekt aus verschiedenen Blickwinkeln und Entfernungen aufgenommen.

Aus der Projektion der 3D-Punkte in die Kamerabildebene kann die relative Position und Orientierung der Kamera zum Objekt berechnet werden. Zudem lässt sich auch auf die intrinsischen Parameter der Kamera, wie z.B. Brennweite, optische Achse und Distorsionsmodell schließen.

Als Kalibrierungsobjekt wird von OpenCV ein Schachbrett eingesetzt. Dabei müssen nicht notgedrungen 8 mal 8 Felder vorhanden sein. Im Gegenteil ist es sogar besser, ein Muster mit

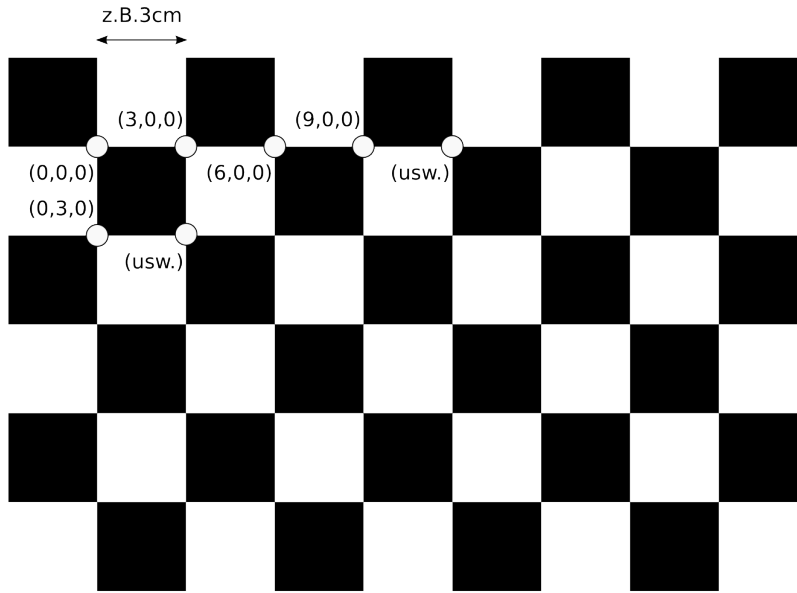


Abbildung 3.1.: Beispiel Kalibrierungsobjekt mit einigen Objektkoordinaten

unterschiedlicher Anzahl von Feldern in Höhe und Breite zu verwenden, da die Orientierung so eindeutig bestimmt ist. Die Herstellung ist einfach, das Muster kann ausgedruckt und auf einer ebenen Oberfläche wie z.B. eine Sperrholzplatte befestigt werden.

3.1.1.2. Analyse der Schachbretter

Als 3D-Punkte werden die inneren Ecken des Schachbrettes verwendet. Der Wechsel von Schwarzen und Weißen Feldern bietet den Vorteil, dass das entstehende Gitter eine gute Detektion der Ecken ermöglicht. Jeder Ecke werden 3D-Koordinaten wie in Abbildung 3.1 zugewiesen. Der Ursprung des Koordinatensystems befindet sich in der linken oberen Ecke.

Zunächst wird für ein Bild eine Detektion der Ecken durchgeführt. Dazu kann z.B. der Harris Corner Detector [Harris und Stephens, 1988] verwendet werden. Anschließend ist es notwendig die Position gefundenen Ecken mit Subpixelgenauigkeit zu optimieren, da andernfalls erhebliche Fehler in der Kalibrierung auftreten können. Die so bestimmten 2D-Bildkoordinaten werden später vom eigentlichen Kalibrierungsvorgang mit den realen 3D-Objektkoordinaten verknüpft.

3.1.1.3. Lösung der Kamera Kalibrierung

Die Projektion der 3D-Punkte des Schachbretts in die Bildebene lässt sich als Matrixmultiplikation darstellen. Die Matrix H ist aus der Kameramatrix M , welche die intrinsischen Parameter enthält und einer Transformationsmatrix, bestehend aus Rotationsmatrix und Translationsvektor, zusammengesetzt. In homogenen Koordinaten sieht die Gleichung wie folgt aus:

$$\begin{bmatrix} x_{Bild} \\ y_{Bild} \\ 1 \end{bmatrix} = H \begin{bmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_1 \\ r_{12} & r_{22} & r_{32} & t_1 \\ r_{13} & r_{23} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_{obj} \\ y_{obj} \\ z_{obj} \\ 1 \end{bmatrix} \quad (3.1)$$

Da die Z-Komponente der Objektkoordinaten in der Schachbrettebene gleich 0 gewählt wurde, vereinfacht sich die Gleichung.

$$\begin{bmatrix} x_{Bild} \\ y_{Bild} \\ 1 \end{bmatrix} = M \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_1 \\ r_{12} & r_{22} & r_{32} & t_1 \\ r_{13} & r_{23} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x_{obj} \\ y_{obj} \\ 0 \\ 1 \end{bmatrix} = M \begin{bmatrix} r_{11} & r_{21} & t_1 \\ r_{12} & r_{22} & t_1 \\ r_{13} & r_{23} & t_3 \end{bmatrix} \begin{bmatrix} x_{obj} \\ y_{obj} \\ 1 \end{bmatrix} \quad (3.2)$$

Aus den erkannten 2D-Bildkoordinaten einer Schachbrettaufnahme und den dazugehörigen 3D-Objektkoordinaten lässt sich aus die Matrix H schließen, ohne dass dafür die intrinsischen Parameter der Kamera bekannt sind.

Sind ausreichend Schachbrettaufnahmen vorhanden, kann dies ausgenutzt werden und auf die Zerlegung von H geschlossen werden. Anschließend werden die mit 0 initialisierten Distorsionskoeffizienten durch ein Gleichungssystem unter Zuhilfenahme der zuvor bestimmten intrinsischen Parameter und Gleichungen (2.5) und (2.6) gelöst. Das genaue mathematische Vorgehen der Kalibrierung ist in [Zhang, 2000], [Zhang, 1999] und [Bradski und Kaehler, 2008] beschrieben.

Es bleibt zu klären wie viele Schachbrettaufnahmen zur Lösung der Kalibrierung ausreichen.

3.1.1.4. Anzahl notwendiger Schachbrettaufnahmen

Durch die Kalibrierung sind folgende Parameter zu lösen:

- 5 Distorsionskoeffizienten
- 4 instrinsische Parameter
- 6 extrinsische Parameter je Schachbrett (3 Eulerwinkel und ein 3D-Translationsvektor zur Bestimmung der Position des Schachbrettes)

Die Projektion von 4 Ecken eines Schachbrettes in die Bildebene reicht aus, damit die Projektion der anderen Ecken eindeutig fest gelegt ist. Daher liefert die Projektion des Schachbrettes 8 Freiheitsgrade, für 4 Ecken je zwei Koordinaten. Zum Bestimmen der 4 instrinsischen Parameter sind daher mindestens 2 Aufnahmen von Schachbrettern aus unterschiedlichen Blickwinkeln erforderlich.

$$2 \cdot 8 \text{ Freiheitsgrade} = 4 \text{ instrinsische Parameter} + 2 \cdot 6 \text{ extrinsische Parameter}$$

Um gute Ergebnisse in der Kalibrierung erzielen zu können, sind eher 10 oder mehr Aufnahmen notwendig, aufgrund von Aspekten wie Rauschen und numerischer Stabilität.

Zur Bestimmung der 5 Distorsionskoeffizienten genügen theoretisch 3 Ecken eines Schachbrettes aus. Sie liefern als 2D-Punkte insgesamt 6 freie Parameter. Alle weiteren verfügbaren Punkte dienen ebenfalls der Optimierung.

(vgl. [Bradski und Kaehler, 2008])

3.1.2. Stereo Kalibrierung

„Stereo Kalibrierung ist die Berechnung des geometrischen Zusammenhanges der beiden Kameras im Raum.“ [Bradski und Kaehler, 2008] Gesucht werden die extrinsischen Parameter des Stereokamerasystems, die Rotationsmatrix R und der Translationsvektor T , mit welchen die rechte Kamera mathematisch in die linke überführt werden kann.

Angenommen, beide Kameras wurden bereits mit dem Verfahren aus 3.1.1 kalibriert. Weiterhin wurden die Schachbretter von beiden Kameras gleichzeitig aufgenommen. Bei der Einzelkalibrierung wurde für jedes Schachbrett eine Rotationsmatrix und ein Translationsvektor berechnet, welche die Objektkoordinaten P in Kamerakoordinaten P_l und P_r transformieren.

$$P_l = R_l P + T_l \quad (3.3)$$

$$P_r = R_r P + T_r \quad (3.4)$$

Die beiden Punkte P_l und P_r lassen sich wie folgt ineinander überführen

$$P_l = R^T (P_r - T) \quad (3.5)$$

Aus den drei Gleichungen können R und T berechnet werden.

$$R = R_r (R_l)^T \quad (3.6)$$

$$T = T_r - R T_l \quad (3.7)$$

R und T werden dann für jedes Schachbrett separat berechnet. Die so erhaltenen Matrizen und Vektoren unterscheiden sich in der Realität geringfügig voneinander, z.B. aufgrund von Bildrauschen oder Rundungsfehlern. Daher wird über alle Werte gemittelt. Die so erhaltenen Werte für R und T werden als Startwerte für eine iterative Approximation verwendet um die optimalen Werte zu finden. (vgl. [Bradski und Kaehler, 2008])

3.2. Stereokorrespondenzanalyse

Entfernungen werden beim stereo-optischen Ansatz indirekt, durch Vergleich der Projektionen eines Objektes auf zwei oder mehr Bilder gemessen. Die Aufgabe der Stereokorrespondenzanalyse ist es, korrespondierende Elemente in den Bildern zu finden und die Disparität zu bestimmen, welche per Triangulation (2.5) in die Tiefenentfernung umgerechnet werden kann.

Der Mensch kann intuitiv Muster erkennen und vergleichen. Daher ist es für ihn sehr leicht, Gemeinsamkeiten und Unterschiede zweier Bilder zu entdecken. Diese scheinbar einfache Aufgabe stellt für den Computer hingegen eine Herausforderung dar. Mittlerweile gibt es sehr viele Algorithmen, welche das Problem der Korrespondenzanalyse zu Lösen versuchen.

Auf der Middlebury Stereo Evaluation Website [Scharstein] findet sich ein Ranking von State-of-the-Art Matching-Algorithmen. Mit durchschnittlich ca. 4 - 5% falsch gematchten Pixeln in den Testbildern erzielen viele sehr gute Ergebnisse.

3.2.1. Klassifizierung von Matching-Algorithmen

Generell kann man die Matching-Algorithmen in zwei Klassen aufteilen.

- **lokale Methoden** beziehen die Korrelation der lokalen Umgebung des Pixels, dessen Disparität berechnet wird, mit ein.
- **globale Methoden** formulieren die Suche nach der richtigen Disparität für jedes Pixel als globales Energie-Minimierungsproblem. Sie matchen pixelweise, berücksichtigen dabei aber auch die Oberflächenkontinuität.

Globale Methoden basieren z.B. auf Graph-Cut oder Dynamischer Programmierung. Sie erzeugen sehr genaue Disparitätenkarten, benötigen dafür aber sehr viel Rechenzeit und Speicher. Für Echtzeitanwendungen sind sie nicht geeignet. Ein Großteil der auf [Scharstein] gelisteten Algorithmen fallen darunter.

Lokale Methoden sind weniger genau als Globale. Sie haben speziell Probleme bei der Verarbeitung von Objekträndern und Tiefendiskontinuitäten, da sie davon ausgehen, dass sich die lokale Umgebung des untersuchten Pixels in der gleichen Tiefe befindet. Das führt zu falschen Matchingergebnissen. Von großem großem Vorteil ist allerdings, dass sie eine vergleichsweise einfache Struktur haben und dadurch optimierte Implementierungen möglich sind. Sie sind schnell und robust und haben einen geringen Speicherverbrauch. Für Echtzeitanwendungen wie Entfernungsmessung kommen nur lokale Methoden in Frage.

(vgl. [Konolige, 1997] [Hirschmüller, 2005][De-Maeztu et al., 2011])

3.2.2. Typischer Aufbau korrelationsbasierter Matching-Algorithmen

Ein korrelationsbasierter Matching-Algorithmus besteht üblicherweise aus folgenden vier Schritten:

1. **Prefilter/Pre-Processing:** z.B. Normalisieren der Graustufen-Intensitäten, Reduzierung von Rauschen, Verbesserung der Textur, etc.
2. **Berechnung der Kostenfunktion:** Das eigentliche Matching. Es werden die Kosten für alle, für ein Pixel des Basis-Bildes möglichen, Disparitäten entlang der zugehörigen Epipolarlinie berechnet.
3. **Extremabestimmung:** Suche in der zuvor berechneten Kostenfunktion nach der Disparität mit den minimalen Matching-Kosten und anschließende Optimierung mit Subpixelgenauigkeit.
4. **Postfilter/Post-Processing:** Bereinigen der erstellten Disparitätenkarte, durch Herausfiltern falscher Matches

Im Folgenden werden die beiden im Rahmen dieser Arbeit verwendeten und in OpenCV implementierten Matching-Verfahren Block-Matching und Semi-Global-Block-Matching genauer betrachtet.

3.2.3. Suche nach Korrespondenzen durch lokale Methoden

Bei der Suche nach Korrespondenzen unterscheidet man zwischen Basisbild und Matchingbild. Für jedes Pixel des Basisbildes wird nach dem korrespondierenden Pixel im Matching-Bild gesucht. Die so entstehende Disparitätenkarte gehört zum Basisbild. In den meisten Fällen wird das linke Kamerabild als Basisbild verwendet und das rechte als Matching-Bild, es ist aber auch anders herum möglich.

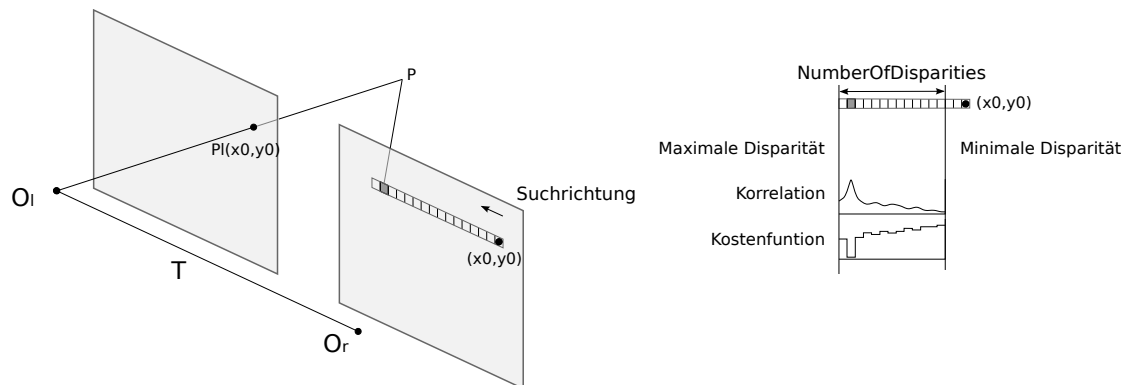


Abbildung 3.2.: Korrespondenzanalyse

Abbildung 3.2 zeigt das Vorgehen bei der Suche. Aufgrund der gewählten Standardstereogeometrie, verlaufen die Epipolarlinien entlang einer Bildzeile. Korrespondierende Pixel können sich bei den gleichen Bildkoordinaten befinden, in diesem Fall ist die Disparität null, die Projektionsgeraden verlaufen parallel und das abgebildete Objekt befindet sich in unendlicher Entfernung. Für Disparitäten größer null befinden sich die korrespondierenden Pixel im Matching-Bild in Richtung des Basis-Bildes verschoben.

Die Suche nach Korrespondenzen für einen Punkt $P_l = (x_0, y)$ im Basisbild beginnt daher im Punkt $P_r = (x_0 - d_{min}, y_0)$, wobei d_{min} die minimale Disparität ist, ab welcher gesucht werden soll. d_{min} wird häufig mit null gewählt.

Meist suchen lokale Stereo-Algorithmen nur einen begrenzten Bereich von Disparitäten *NumberOfDisparities* ab, um Rechenzeit zu sparen. Daraus ergibt sich die maximal erkennbare Disparität.

Üblicherweise setzen lokale Stereo-Methoden kleine Fenster mit der Umgebung eines Pixels zum Vergleichen ein. Je größer es gewählt wird, desto größer ist auch die Wahrscheinlichkeit für einen korrekten Match, allerdings entsteht eine weniger detaillierte Disparitätenkarte.

3.2.4. Block-Matching

Beim Block-Matching-Algorithmus handelt es sich um ein von Kurt Konolige [Konolige, 1997] entwickeltes, schnelles und effektives Verfahren. Bei diesem lokalen Verfahren werden Korrespondenzen gefunden, indem kleine Fenster entlang der Epipolarlinie miteinander verglichen werden. Zum Matchen benötigt der Algorithmus Punkte mit starker Textur. Homogene Bereiche der Umgebung, wie z.B. glatte Wände, werden nicht erkannt und die Disparitätenkarte bleibt dort leer.

Als Eingabe für den Algorithmus werden rektifizierte (2.4) Graustufenbilder erwartet. Die einzelnen Schritte sehen in der OpenCV Implementierung wie folgt aus:

Pre-Processing: In diesem Schritt werden beim Block-Matching die Graustufenbilder normalisiert, um Unterschiede in der Belichtung zu reduzieren. Dazu wird die Intensität jedes Pixel I_c durch folgende Funktion ersetzt: $I_c = \min(\max(I_c - \bar{I}, -I_{cap}), I_{cap})$. \bar{I} ist dabei die mittlere Intensität in einem Fenster um das zu normalisierende Pixel I_c . Bei I_{cap} handelt es sich um eine positive Grenze mit Standardwert 30.

Berechnung der Kostenfunktion Wie bereits erwähnt, werden beim Block-Matching Korrespondenzen durch das Verschieben kleiner Fenster entlang der epipolaren Linie berechnet. Dabei werden Pixel des Matching-Bildes und deren Umgebung mit dem Pixel des Basisbildes durch die Summe absoluter Differenzen (SAD) verglichen.

$$C(x, y, d) = \sum_{i=-k}^k \sum_{j=-k}^k |I_b(x+i, y+j) - I_m(x+i-d, y-j)| \quad (3.8)$$

Die Größe des Fensters wird durch $n = m = 2k + 1$ festgelegt. Bei I_b handelt es sich um die Intensität im Basisbild, I_m ist dementsprechend die Intensität im Matching-Bild. Bei d handelt es sich um die Disparität, für welche die Kosten berechnet werden. Die Kosten werden für alle Disparitäten bis zur maximalen Disparität berechnet. Dabei gilt, je kleiner C desto größer die Korrelation.

Extremabestimmung Ist die Kostenfunktion für ein Pixel des Basis-Bildes berechnet, werden deren Minima bestimmt. Die Position des Globalen Minimums wird als Kandidat für die Disparität festgelegt. Da es sich um eine diskrete Funktion handelt, liegt das wahre Minimum der Kosten mit hoher Wahrscheinlichkeit zwischen 2 Disparitäten. Daher wird eine quadratische Interpolation durchgeführt, indem eine Kurve durch das Minimum und dessen Nachbarn gelegt wird. Die so gefundene Disparität wird von OpenCV mit einer Genauigkeit von $\frac{1}{16}$ px aufgelöst.

Post-Processing Zum bereinigen der Disparitätenkarte kommen in der OpenCV- Implementierung des Block-Matching-Algorithmus drei Filter zum Einsatz:

- Mit der *UniquenessRatio* wird überprüft, um wie viel Prozent das gefundene globale Minimum der Kostenfunktion besser ist als das zweitbeste Minimum. Liegen die Kosten der Minima zu dicht beieinander, wird der Match verworfen.
- Über einen *TextureThreshold* wird sichergestellt, dass gematchte Pixel genügend Textur aufweisen, um zufälliges Bildrauschen auszuschließen
- Der sogenannte *SpeckleFilter* bereinigt die Disparitätenkarte an den Rändern von Objekten. Dort entstehen aufgrund der Verwendung von SAD-Fenstern kleine Bereiche, welche sich sehr stark in der Disparität unterscheiden. Der Speckle-Filter überprüft, ob sich innerhalb eines über die Disparitätenkarte verschobenen Fensters, die minimale und maximale Disparität um höchstens den eingestellten *SpeckleRange* unterscheidet. Dieser liegt normalerweise bei einem Wert zwischen 1 und 4.

3.2.5. Semi Global Block Matching

Bei dem von OpenCV implementierten Semi-Global-Block-Matching (SGBM) Algorithmus handelt es sich um eine Abwandlung des Semi Global Matching Algorithmus (SGM) von Heiko Hirschmüller [Hirschmüller, 2005][Hirschmüller, 2008]. Dieser verbindet Konzepte von lokalen und globalen Methoden für genaues, pixelweises Matching bei niedriger Laufzeit. (vgl. [Hirschmüller, 2011]). Neben der Kostenfunktion wird ein globales Glattheitskriterium verwendet, welches den Zusammenhang zwischen den Pixeln modelliert. Im Vergleich zum Block-Matching sollten SGM und SGBM theoretisch besser mit Objekträndern und wenig strukturierten Oberflächen zurecht kommen.

Im folgenden werden die wichtigsten Schritte zur Berechnung der optimalen Disparitätenkarte durch SGM und SGBM beschrieben.

Berechnung der Kostenfunktion Im Gegensatz zum SGM, welcher in seiner ursprünglichen Implementierung zur Berechnung der Kosten das Konzept der Mutual Information (siehe dazu [Hirschmüller, 2005]) verwendet, kommt beim SGBM eine einfachere Kostenfunktion von Birchfeld und Tomasi [Birchfield und Tomasi, 1998] zum Einsatz. Diese baut auf dem Konzept von absoluten Differenzen zum Vergleichen von Pixeln oder Bildern auf.

Sei p ein Pixel im Basisbild und $q = (x_p - d, y_p)^T$ das Pixel im Matching-Bild mit Disparität d , dessen Kosten berechnet werden. Diese berechnen sich als die absolute minimale Differenz der Intensitäten zwischen p und q im Bereich von einem halben Pixel in jeder Richtung entlang der epipolaren Linie (vgl. [Hirschmüller, 2005])

$$C_{BT}(p, q) = \min_{x_q - \frac{1}{2} \leq x_p \leq x_q + \frac{1}{2}} |I_b(p) - I_m(q)| \quad (3.9)$$

Beim SGM werden zur Berechnung der Kosten einzelne Pixel-Intensitäten verglichen. Beim SGBM werden im Unterschied dazu, genau wie beim Block-Matching, kleine Fenster eingesetzt.

Minimierung der Globalen Energiefunktion Wie bereits erwähnt, haben lokale Methoden Probleme bei der Bestimmung der korrekten Disparitäten an Objekträndern mit starken Tiefenunterschieden. Dem begegnen SGM und SGBM, indem wie bei vielen globalen Methoden üblich, ein sogenanntes Glattheitskriterium (smoothness constraint) verwendet wird. Es wird also nach einer Disparitätenkarte D gesucht, welche die globale Energiefunktion $E(D)$ minimiert.

$$E(D) = \sum_p \left(C(p, d_p) + \sum_{q \in N_p} P_1 T[|d_p - d_q| = 1] + \sum_{q \in N_p} P_2 T[|d_p - d_q| > 1] \right) \quad (3.10)$$

Beim ersten Term der Funktion handelt es sich um die Kosten für ein Pixel mit Disparität d_p . Der Zweite Term betrachtet die direkten Nachbarn von q und „bestraft“ kleine Änderungen der Disparitäten mit dem Wert P_1 . Der dritte Term bestraft analog zum zweiten Term größere Änderungen in den Disparitäten der Nachbarschaft mit dem Wert P_2 . T gibt dabei 0 oder 1 zurück, je nachdem ob das Argument zutrifft oder nicht. (vgl. [Hirschmüller, 2005]) Bei dem zweiten und dritten Term handelt es sich um das Glattheitskriterium.

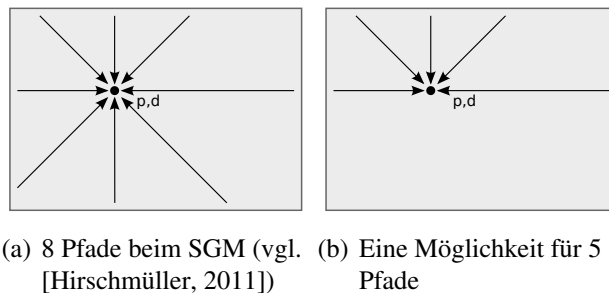


Abbildung 3.3.: Beispiel für Pfade zum Erreichen eines Pixels p mit Disparität d

Das Globale Minimum für $E(D)$ für das ganze 2D Bild zu finden, ist ein NP-Vollständiges Problem. SGM und SGBM berechnen dagegen $E(D)$ effizient entlang von Pfaden zum Pixel p aus mehreren Richtungen. (vgl. [Hirschmüller, 2006]) Beim SGM sind es 8 bzw. 16 Pfade. Beim SGBM werden standardmäßig nur 5 Pfade verwendet, um Speicherplatz zu sparen. Für jedes Pixel werden die Kosten der Pfade aufsum-

miert und die Disparität mit den geringsten Kosten ausgewählt. Wie beim BM erfolgt die anschließende Bestimmung der Subpixeldisparität wieder durch Quadratische Interpolation.

Das Ergebnis ist die Disparitätenkarte D_b welche zum Basis-Bild gehört. Okklusionen wurden bis dahin nicht betrachtet. Sie werden durch den sogenannten Left/Right-Check herausgefiltert. Dazu wird auf gleiche Weise die Disparitätenkarte D_m berechnet, welche zum Matching-Bild gehört. Beide Disparitätenkarten werden dann miteinander verglichen und voneinander abweichende Disparitäten entfernt.

Post-Processing Als weitere Filter zum Bereinigen der Disparitätenkarte kommen *UniquenessRatio* und *Speckle-Filter* aus dem Block-Matching zum Einsatz.

Desweiteren wird ein Median-Filter auf die Disparitätenkarte angewendet. Dadurch werden nochmals Ausreißer mit stark unterschiedlichen Disparitäten im Vergleich zur Umgebung entfernt und die Disparitätenkarte somit geglättet.

3.3. OpenCV

OpenCV ist eine Open-Source Softwarebibliothek für Computer Vision und Machine Learning. Die Entwicklungssprachen sind C und C++. Es sind zudem Interfaces für Python und Java verfügbar. OpenCV kann unter Windows-, Linux- und MacOS sowie iOS und Android verwendet werden.

Eines der Hauptziele von OpenCV ist es, eine einfach zu benutzende Computer Vision Infrastruktur zu schaffen, die es erlaubt mit wenig Zeitaufwand anspruchsvolle Anwendungen zu entwickeln. Es stehen mittlerweile mehr als 2500 optimierte Algorithmen zu Verfügung, sowohl etablierte als auch State-of-the-Art.

Beim Design wurde auf eine effiziente Nutzung der Rechenzeit mit einem starken Fokus auf Echtzeitanwendungen Wert gelegt. OpenCV kann unter anderem die Möglichkeiten von Mehrkern-Systemen ausnutzen.

Das Projekt wurde 1999 initiiert, die erste finale Version erschien 2006. Mittlerweile ist die Version 2.4.6 verfügbar. Mit der Version 2.0 wurde 2009 eine neue API eingeführt. Die Bibliothek ist nun in verschiedene Module aufgeteilt.

Abbildung 3.4 zeigt eine Übersicht aller Module von OpenCV 2.4.0.:

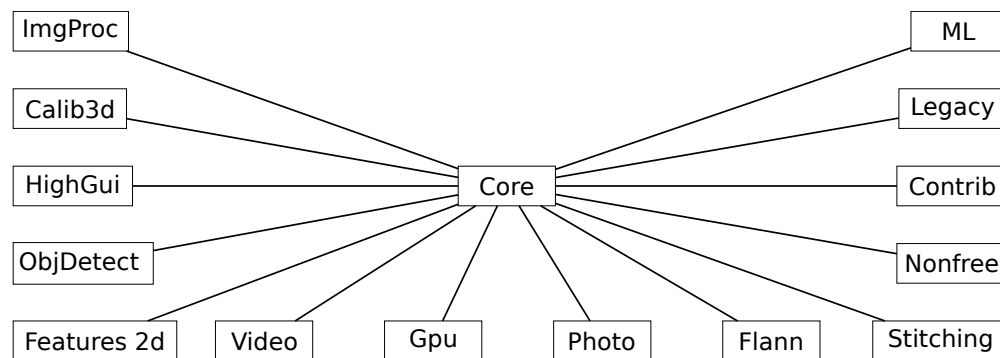


Abbildung 3.4.: OpenCV 2.4.0 Modulübersicht

- **Core:** stellt die Basisfunktionalität zur Verfügung. Das Modul definiert Datenstrukturen und Funktionen, welche von allen anderen Modulen verwendet werden
- **ImgProc:** Bildverarbeitungsfunktionen wie lineare und nichtlineare Filter, geometrische Bildtransformationen, Farbraumkonvertierung, Histogramme u. a.
- **HighGui:** einfache GUI-Funktionalität, Interface für Kameras, Bild- und Video-Codecs
- **Calib3d:** Stereoalgorithmen, Kamerakalibrierung, 3D-Rekonstruktion, u.a.
- **Features2d:** Feature Detektoren und Feature Matching Algorithmen
- **ObjDetect:** Algorithmen für Objekterkennung, Klassifizierung von z.B. Gesichtern, Augen, Autos, Menschen, u.a.
- **Video:** Algorithmen für die Videoanalyse, wie z.B. Tracking, Hintergrund entfernen, Bewegungserkennung
- **ML:** Algorithmen für Maschinelles Lernen.
- **Gpu:** Gpu-Implementierung von Algorithmen verschiedener Module
- **Contrib:** enthält experimentelle Algorithmen
- **Nonfree:** beinhaltet nicht freie Algorithmen z.B. den SURF Feature Detektor
- **Legacy:** als veraltet eingestufte Algorithmen, sie können in folgenden Versionen wegfallen.
- **Photo:** Fotoverarbeitung und Analyse
- **Stitching:** Zusammensetzen von Fotos zu Panoramen
- **Flann:** Clustering und Suche in Multi-Dimensionalen Räumen

(vgl. [OpenCV, 2013] [OpenCV, 2012])

4. Konzept

Ziel dieser Arbeit ist es, ein System zur optischen Entfernungsmessung aufzubauen. Dabei kommen zwei handelsübliche Webcams aus dem Low-Cost Bereich zur Anwendung. Eingesetzt werden soll dieses System zur Unterstützung bereits bestehender Sensorsysteme des Quadrocopters. Dazu wird ein stereoskopischer Ansatz verfolgt. Das bedeutet, dass ein Objekt von zwei Kameras aus leicht unterschiedlichen Blickwinkeln beobachtet wird. Dadurch erscheint das Objekt in den beiden Bildern verschoben. Aus dieser Verschiebung wird per Triangulation (2.5) eine 3D Rekonstruktion des gezeigten Objektes möglich.

Im folgenden wird das Konzept, welches der Softwareimplementierung aus Kapitel 5 zugrunde liegt erläutert. Dabei wird das System fortan als StereoVision System bezeichnet.

4.1. StereoVision System

Das StereoVision System kann grob in zwei Arbeitsschritte unterteilt werden.

1. **Die Kalibrierung** bereitet das System auf die Berechnung der Tiefeninformation vor und sorgt für die Verknüpfung von physikalischen Einheiten (z.B. cm) und den Bildeinheiten (Pixel)
2. **Das Matching** führt die Messung der Tiefenentfernung von Objekten durch.

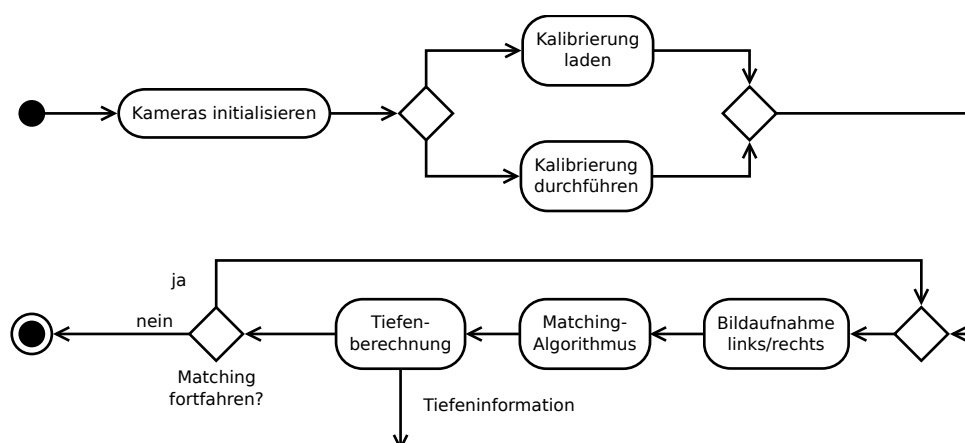


Abbildung 4.1.: Konzept des Gesamtsystems

In Abbildung 4.1 ist der Ablauf des Stereo Vision Systems dargestellt. Zu Beginn werden die Kameras initialisiert. Dabei wird die Auflösung eingestellt und festgelegt, welche Kamera die jeweils linke und rechte ist. Im nächsten Schritt wird eine Kalibrierung (siehe 4.2) der beiden Kameras durchgeführt. Alternativ kann eine bestehende Kalibrierung aus einer Datei geladen

werden. Das spart Zeit und ist sinnvoll, wenn die Anordnung der Kameras in der Zwischenzeit nicht modifiziert wurde. Ist die Kalibrierung abgeschlossen, kann das Matching (siehe 4.3) durchgeführt werden. Dazu werden fortlaufend Bilder von beiden Kameras aufgenommen und die deren Korrelation durch den Matching-Algorithmus berechnet. Aus den Verschiebungen von Objekten zwischen beiden Bildern kann anschließend die Tiefenentfernung berechnet werden. Dieser Vorgang wird solange wiederholt, wie es vom Benutzer gewünscht wird.

4.2. Kalibrierung

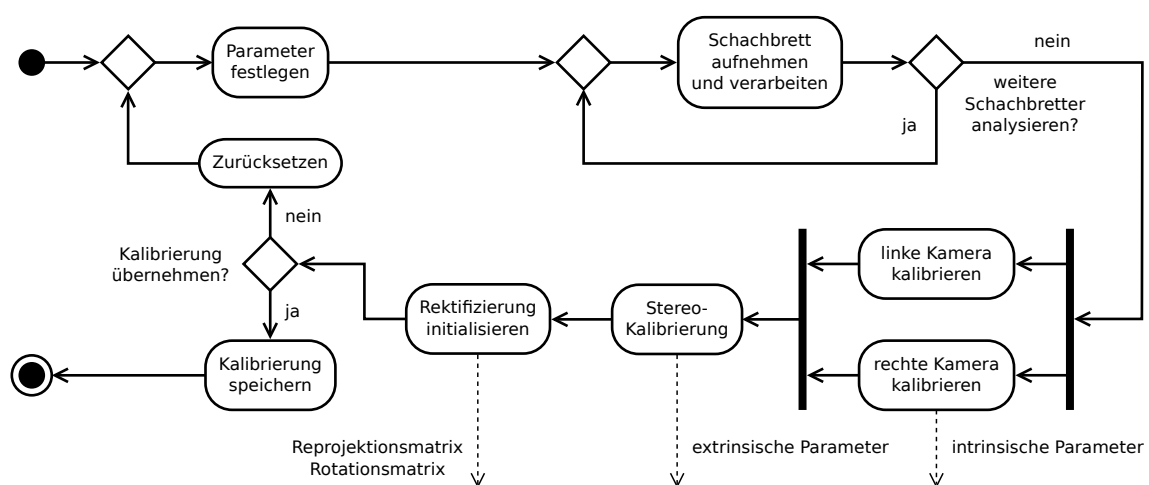


Abbildung 4.2.: Konzept der Kalibrierung

Zu Beginn des Kalibrierungsvorganges müssen zunächst einige Parameter festgelegt werden. Das sind zum einen die gewünschte Auflösung der Kameras, zum anderen muss angegeben werden, wie viele innere Ecken das verwendete Schachbrett in der Höhe und Länge hat. Zudem ist die Angabe der Seitenlänge eines Schachbrettfeldes essentiell. Über sie wird die Größe des Schachbrettes im Bild definiert und der Zusammenhang zwischen einer Strecke im Bild in Pixeln und dem wahren Abstand auf dem Schachbrett hergestellt. (Siehe Kapitel 3.1.1.2) Sind die Parameter festgelegt, können mit beiden Kameras gleichzeitig Bilder von einem Schachbrett in unterschiedlichen Positionen und Ausrichtungen gemacht werden. Diese werden jeweils analysiert, das heißt die Bildkoordinaten der inneren Ecken des Schachbrettes werden ermittelt und zwischengespeichert.

Wurden ausreichend viele Aufnahmen des Schachbrettes durchgeführt (siehe Kapitel 3.1.1.4), kann der eigentliche Kalibrierungsvorgang gestartet werden. Mit Hilfe der zwischengespeicherten Koordinaten werden sowohl die linke als auch die rechte Kamera zunächst einzeln kalibriert. Das dient dazu die sogenannten intrinsischen Parameter zu bestimmen. Das sind die Brennweite der Kamera in Pixel, der Principal Point (2.2.2) sowie die Distorsionskoeffizienten (2.2.3) zum Entfernen von Abbildungsfehlern, welche durch die Kamera verursacht werden.

Im Anschluss wird die Stereokalibrierung durchgeführt. Sie ermittelt die extrinsischen Parameter des StereoVision Systems, welche beschreiben, wie die rechte Kamera durch Rotation und Translation in die linke Kamera umgerechnet werden kann.

Das Matching (4.3) erfordert rektifizierte Bilder. Das bedeutet, dass sich ein Merkmal des abgebildeten Objektes im linken und rechten Bild in der gleichen Bildzeile befinden müssen. Die

Kamerabilder werden mit Hilfe der extrinsischen Parameter aus der Stereokalibrierung durch die Rektifizierung (2.4) so transformiert, dass die eben genannte Bedingung erfüllt ist. Bei der Initialisierung der Rektifizierung werden die dazu erforderlichen Rotationsmatrizen berechnet. Zudem wird die Reprojektionsmatrix Q erstellt, welche alle für die Triangulation (4.3) notwendigen Informationen wie z.B. die Brennweite des rektifizierten Kamerabildes und den Abstand zwischen den beiden Kameras enthält.

Nach Abschluss des Kalibrierungsvorganges wird entschieden, ob die Kalibrierung übernommen wird. Das geschieht zum einen objektiv Anhand des Reprojektionsfehlers, welcher während der Stereokalibrierung ermittelt wird. Ist dieser größer als 1, sind die Ergebnisse der Kalibrierung zu ungenau und daher unbrauchbar. Die Kalibrierung wird in diesem Fall automatisch zurückgesetzt. Zum anderen wird subjektiv vom Benutzer entschieden, ob sich die rektifizierten Bilder ausreichend überschneiden, nicht zu stark verzerrt wurden und die durch das Matching berechneten Entfernungen im Rahmen der gewünschten Messungenauigkeit liegen. Ist das der Fall, so können die Ergebnisse abgespeichert werden, damit beim nächsten Start des Stereo Vision Systems nicht von neuem kalibriert werden muss. Anderenfalls kann die Kalibrierung manuell zurückgesetzt werden. Dabei werden alle zwischengespeicherten Bildkoordinaten der Schachbrettecken gelöscht und die Ausgabematrizen der Kalibrierung zurückgesetzt.

Im Fall einer erfolglosen Kalibrierung sollte die Anordnung der Kameras überprüft und gegebenenfalls korrigiert werden. (siehe Kapitel 3.1 und 2.2.4)

4.3. Matching

Das Matching bildet das Kernmodul des Stereo Vision Systemes. Es erledigt die fortlaufende Entfernungsmessung.

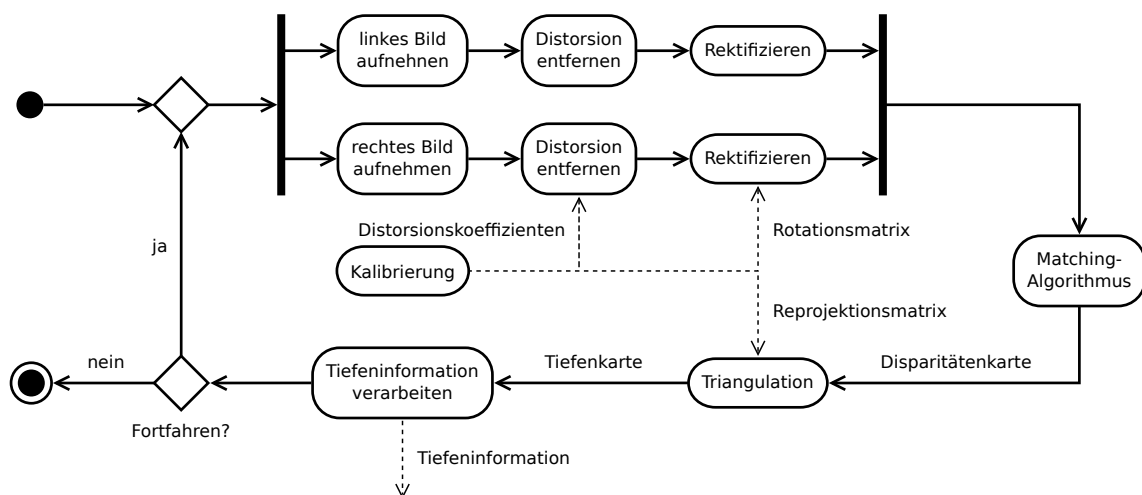


Abbildung 4.3.: Ablauf des Matching

Zunächst werden von der linken und der rechten Kamera Bilder aufgenommen. Dies sollte so gleichzeitig wie möglich geschehen, damit sich bewegende Objekte in beiden Bildern an der gleichen realen Position befinden und die Tiefeninformation nicht verfälscht wird. Die aufgenommenen Bilder werden dann für den Matching-Algorithmus vorbereitet. Dazu wird die Distorsion mit Hilfe der Distorsionskoeffizienten aus der Kalibrierung entfernt. Das bedeutet, dass sie entzerrt

werden und Abbildungsfehler korrigiert werden. Beispielsweise wird eine gerade Linie, welche durch die Distorsion gebogen abgebildet war, durch diesen Schritt wieder begradigt. Weiterhin müssen die entzerrten Bilder, wie bereits im Abschnitt Kalibrierung (4.2) erwähnt, rektifiziert werden, gleiche Merkmale müssen sich also in der gleichen Bildzeile befinden. Dazu wurde bereits in der Kalibrierung Rotationsmatrizen für jede Kamera berechnet. Mit diesen wird jedes Bild so transformiert, dass sie schließlich rektifiziert sind.

Die Hauptarbeit des StereoVision Systems wird vom Matching-Algorithmus geleistet. Er berechnet für jeden Bildpunkt des linken Bildes die Verschiebung bzw. Disparität des korrespondierenden Punktes im rechten Bild. Ausgegeben wird eine sogenannte Disparitätenkarte, welche alle berechneten Verschiebungen enthält. Für das Matching stehen die in Kapitel 3.2 genannten Algorithmen BM und SGBM zur Verfügung.

Die Disparitäten lassen sich per Triangulation (2.5) in die reale Tiefenentfernung zwischen Kamera und Objekt umrechnen. Im nächsten Schritt wird daher mit Hilfe der Reprojektionsmatrix aus der Kalibrierung, welche alle notwendigen Parameter enthält eine Tiefenkarte erstellt, welche für jeden Bildpunkt des linken Bildes die wahre Tiefenentfernung enthält.

Im letzten Schritt muss die erstellte Tiefenkarte interpretiert werden. Es muss z.B. entschieden werden, wo sich ein Hindernis befindet und in welcher Richtung genügend Raum für den Flug ist. Diese Informationen soll von anderen Systemen des Quadrocopters verwendet werden können.

Dazu wird die Tiefenkarte in mehrere Rechtecke aufgeteilt. Innerhalb eines Rechteckes wird die minimale Entfernung bestimmt. Dazu wird ein Histogramm über die Tiefeninformation des Rechteckes erstellt. Aus dem Histogramm wird anschließend der Abschnitt mit den kleinsten Entfernungen bestimmt, welcher einen festgelegten Mindestanteil der Bildpunkte im Segment enthält. Innerhalb dieses Histogramm abschnittes wird die Klasse mit der größten Häufigkeit ausgewählt und als Tiefe für den untersuchten Bereich der Tiefenkarte angenommen.

5. Implementierung

Im folgenden wird auf die Implementierung des in Kapitel 4 vorgestellten Konzeptes eingegangen. Der Hauptaugenmerk liegt dabei auf der Implementierung eines Softwaremoduls zur stereooptischen Entfernungsmessung. Dieses Modul dient schließlich als Erweiterung des bestehenden Steuerprogramms AQopterI8 für den Quadrocopter des Lehrstuhls Informatik VIII der Universität Würzburg.

Die Software ist vollständig in C++ geschrieben, unter Verwendung der OpenCV Bibliothek in der Version 2.4.0. Funktionen die von OpenCV stammen sind im Text durch Voranstellung des Namespaces `cv::` kenntlich gemacht. Für die Benutzeroberfläche wurde das QT-Framework (vormals von Nokia) in der Version 4.7.4 verwendet.

5.1. Software

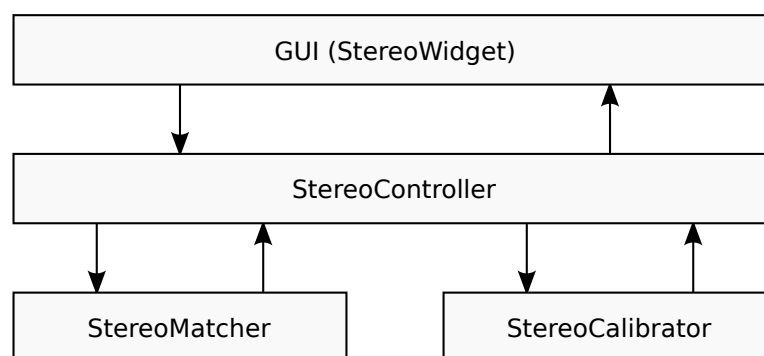


Abbildung 5.1.: Überblick über die Module der Implementierung

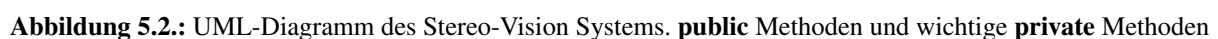
Das implementierte Softwaremodul, fortan StereoVision-System genannt, besteht aus vier Teilmodulen, deren Zusammenhang in Abbildung 5.1 dargestellt ist.

Das StereoWidget übernimmt mit seiner grafischen Benutzeroberfläche die Interaktion mit dem Benutzer und erlaubt eine Konfiguration des Stereo-Controllers. Gleichzeitig ermöglicht es die Kommunikation mit dem Quadrocopter über das Nachrichtensystem des bestehenden Steuerprogrammes AQopterI8.

Der Stereo-Controller ist für die Steuerung und Datenverwaltung des Stereo-Vision Systems verantwortlich. Er kümmert sich unter anderem um die Bildaufnahme und geometrische Korrektur. Seine wichtigste Aufgabe ist die Steuerung der Teilmodule StereoMatcher und StereoCalibrator.

Die für die Kalibrierung notwendige Funktionalität ist im StereoCalibrator zusammengefasst. Das Matching und die Verarbeitung der gewonnenen Tiefeninformation wird vom Teilmodul StereoMatcher übernommen.

Abbildung 5.2 zeigt ein UML-Diagramm des Stereo-Vision Systems.



5.1.1. StereoController

Die Implementierung des StereoController orientiert sich an dem in Kapitel 4.1 erläuterten Konzept. Nachfolgend wird auf die einzelnen Schritte des Programmablaufs näher eingegangen.

Repräsentation der Kameras

Als Bildquelle werden vom StereoController zwei Webcams verwendet, welche wie in Kapitel 2.2.4 beschrieben angeordnet sein müssen. OpenCV stellt mit der Klasse *cv::VideoCapture* einen Treiber für Webcams bereit. Dadurch gestaltet sich die Kommunikation mit diesen sehr einfach.

Alle vom Stereo-Vision System benötigten kameraspezifischen Parameter sind im Struct *Camera* zusammengefasst (siehe UML-Diagramm in Abbildung 5.2)

Initialisieren der Kameras

Zu Beginn des Programmablaufs müssen die Kameras initialisiert werden. Dazu gehört zunächst die Auswahl der richtigen Kameras. OpenCV weist jeder an den Computer angeschlossenen Webcam eine Ziffer zu. Mit der Funktion *setCameraPorts(int portCamLeft, int portCamRight)* des Stereocontrollers lassen sich so die richtigen Kameras auswählen. Das ist insbesondere wichtig, da auch beispielsweise das Objekterkennungsmodul von AQopterI8 eine Webcam verwendet.

Das Starten der Kameras geschieht durch die Funktion *openCams()*. Sie ruft für jede der beiden Kameras die Funktion *cv::VideoCapture::open(int)* auf. Als Parameter wird dabei die Nummer der Kamera übergeben, welche durch *setCameraPorts()* festgelegt wurde. Die standardmäßige Auflösung der verwendeten Logitech C270 Webcams beträgt 640×480 px. Da diese zu hoch für einen flüssigen Ablauf des Stereo-Vision Systems ist, wird sie sofort nach dem Öffnen der Kameras auf eine Auflösung von 320×240 px herabgesetzt.

Prinzipiell ist auch die Verwendung niedrigerer Auflösungen durch das Stereo-Vision System möglich. Durch die Funktion *configureCams(cv::Size)* können diese eingestellt werden. Dazu stehen folgende Auflösungen zur Verfügung: 320×240 px, 256×192 px, 240×180 , 192×144 .

Im Zusammenhang mit der Initialisierung der Kameras muss der StereoController auf die Anordnung der Kameras eingestellt werden. Dazu überprüft man, ob das als linkes Bild gespeicherte auch tatsächlich von der Linken Kamera kommt und nicht von der rechten. Andernfalls können die Kamerabilder vom StereoController mittels *setSwitchCams(bool)* direkt bei der Bildaufnahme getauscht werden. Ähnlich verhält es sich mit der Funktion *RotateCams(bool rotateLeft, bool rotateRight)*. Da die Webcams am Quadrocopter durchaus auch auf dem Kopf montiert werden können, kann mit der Funktion festgelegt werden, welches Kamerabild später um 180 Grad gedreht werden soll.

Kalibrierung

Im nächsten Schritt des Stereo-Vision System muss eine Kalibrierung durchgeführt werden. Deren Implementierung ist in Kapitel 5.1.2 beschrieben. Alternativ kann eine früher durchgeführte Kalibrierung mit der Funktion *loadCalibParams(String Filename)* aus einer Datei geladen werden.

Dabei handelt es sich um eine XML-Datei in welcher die intrinsischen Parameter und die Distorsionskoeffizienten der beiden Kameras sowie die extrinsischen Parameter des Stereo-Vision Systems als Matrix abgelegt sind.

Zudem wird auch die Auflösung gespeichert, da einige Parameter wie z.B. die effektive Brennweite der Kameras in Pixeln von ihr abhängen.

Bildaufnahme und Bildkorrektur

Die Bildaufnahme erfolgt durch die Funktion *updateImages()*. Sie sorgt gleichzeitig für die Korrektur der Distorsionen (siehe 2.2.3) und die Rektifizierung (siehe 2.4).

Für die Qualität des später folgenden Matchings (5.1.3) ist es wichtig, die Bilder beider Kameras so gleichzeitig wie möglich aufzunehmen, um mögliche Bewegungen von Objekten einzufrieren. Das ist prinzipiell nur mit synchronisierten Kameras machbar. Um den zeitlichen Versatz zwischen der Aufnahme beider Bilder gering zu halten, werden die beiden Webcams zunächst unmittelbar nacheinander mit Hilfe der Funktion *cv::VideoCapture::grab()* angewiesen ein Bild zu machen. Im Anschluss werden diese dann mit *cv::VideoCapture::retrieve(Mat)* von den Kameras heruntergeladen und abgespeichert.

Da beide Stereo-Algorithmen später beim Matching 8-Bit Graustufen-Bilder erwarten, erfolgt bereits hier eine Konvertierung in dieses Format. Zudem wird durch die Farbkonvertierung der vorhandene unterschiedliche Weißabgleich der verwendeten Kameras kompensiert. Ebenfalls werden, falls notwendig, die Bilder um 180 Grad gedreht.

Ist das Stereo-Vision System bereits kalibriert, wird zum Schluss der Funktion *updateImages()* die Bildkorrektur durchgeführt. Dazu werden die Bilder mit Hilfe von Look-up-Maps, welche die Korrektur der Distorsionen und die Rektifizierung modellieren auf ein neues korrigiertes Bild gemappt. Die Look-up-Maps stammen aus der Funktion *initRectification()*, auf welche im Zuge der Kalibrierung (5.1.2) genauer eingegangen wird.

5.1.2. Kalibrierung

Die Klasse *StereoCalibrator* stellt die Funktionalität der Kalibrierung zur Verfügung. Der Ablauf eines Kalibrierungsvorganges folgt dabei dem Konzept aus Kapitel 4.2. Die wichtigsten Funktionen sind dabei in Abbildung 5.3 aufgelistet.

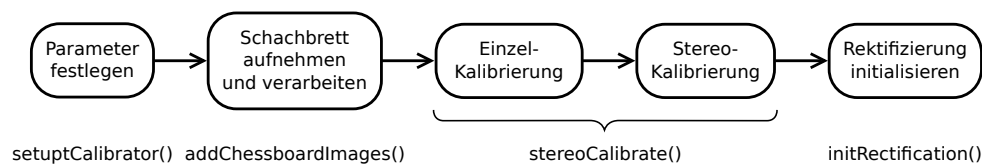


Abbildung 5.3.: Übersicht über wichtige Funktionen des Stereo-Vision Systems zur Kalibrierung

Das Festlegen der Parameter erfolgt durch die Funktion *setupCalibrator(cv::float fieldSize, cv::Size boardSize)*. Dabei handelt es sich um Parameter zur Beschreibung des Kalibrierungsobjektes, in diesem Fall eines Schachbrettes. *boardSize* bestimmt dabei die Anzahl der inneren Schachbrettecken in Höhe und Breite. (vgl. Abbildung 3.1) Mit *fieldSize* wird angegeben, welche Seitenlänge ein Feld des Schachbrettes hat. Die später beim Matching berechneten Tiefenentfernungen haben dabei die gleiche Einheit wie *fieldSize* (z.B. cm).

Verarbeitung der Schachbretter

Im nächsten Schritt der Kalibrierung werden Aufnahmen von Schachbretter aus verschiedenen Blickwinkeln gemacht. Die Aufnahmen werden vom *StereoController* zur Analyse an die Funktion *addChessboardImages(cv::Mat imageLeft, cv::Mat imageRight)* übergeben.

Diese führt zunächst eine Detektion der Schachbrettecken mit Hilfe der Funktion *cv::findChessboardCorners(...)* für beide Eingabebilder durch. Diese führt eine Eckendetektion durch und versucht die inneren Ecken des Schachbrettes zu finden. Dabei müssen genauso viele Ecken pro Schachbrettzeile und -Spalte gefunden werden wie durch *boardSize* angegeben. Die Bildkoordinaten der gefundenen Ecken werden in einem *std::vector<cv::Point2f>* zwischengespeichert.

Ist die Suche nach den Ecken in einem der beiden Bilder nicht erfolgreich, bricht die Funktion *addChessboardImages(...)* ab. Ansonsten werden die Koordinaten der Ecken auf Subpixelgenauigkeit optimiert, mit der Funktion *cv::cornerSubPix(...)*. Andernfalls wäre keine Kalibrierung möglich, deren Ergebnisse eine genaue Messung der Tiefenentfernung zuließen.

Abschließend werden durch die Funktion *addChessboardImages(..)* 3D-Objektkoordinaten für das reale Schachbrett generiert und in einem *std::vector<Point3f>* gespeichert. Wie bereits in Kapitel 3.1.1.2 erwähnt, handelt es sich dabei um Punkte in einem Koordinatensystem in der Schachbrettebene mit Ursprung in der linken oberen Ecke. Die z-Koordinate ist für alle Punkte des realen Schachbrettes 0. Die Koordinaten werden wie in Abbildung 3.1 gezeigt festgelegt.

Der Rückgabewert der Funktion gibt an, ob die Analyse erfolgreich war. Damit die Kalibrierung der Kameras durchgeführt werden kann, muss die Funktion *addChessboardImages(..)* mindestens zweimal erfolgreich ausgeführt werden, wie in Kapitel 3.1 erläutert. Besser ist es jedoch, wenn mindestens 10 oder mehr Schachbretter analysiert werden, um die Genauigkeit der Kalibrierung zu erhöhen.

Einzel- und Stereo-Kalibrierung

Die beiden Schritte des Konzeptes (siehe 4.2) werden durch die Funktion *stereoCalibrate(Camera left, Camera right, cv::Size imageSize, cv::Mat R, cv::Mat T)* erledigt. Sie bestimmt einerseits die intrinsischen Parameter und die Distorsionskoeffizienten der Kameras, welche im Struct *Camera* gespeichert sind. Desweiteren werden durch sie die extrinsischen Parameter des Stereo-Vision Systems wie die Rotationsmatrix R und der Translationsvektor T berechnet.

Dazu wird von ihr zunächst jede Kamera einzeln kalibriert. Das geschieht durch die Funktion *cv::calibrateCamera(...)* welche aus den gespeicherten Bildkoordinaten der Schachbretter und den zugehörigen, generierten Objektkoordinaten die intrinsischen Parameter bestimmt. Das geschieht wie in Kapitel 3.1 beschrieben.

Nach der Kalibrierung der linken und rechten Kamera wird die Funktion *cv::stereoCalibrate(...)* aufgerufen. Die erwartet als Eingabe die Bildkoordinaten beider Bilder, die dazugehörigen Objektkoordinaten und die intrinsischen Parameter aus der Einzelkalibrierung. Mit diesen Parametern werden analog zu Kapitel 3.1 die extrinsischen Parameter des Stereo-Vision Systems ermittelt. Sie gibt gleichzeitig einen Reprojektionsfehler zurück, mit welchem sich einschätzen lässt, wie gut die Kalibrierung ist. Hat dieser einen Wert größer 1, so ist die Kalibrierung in jedem Fall unbrauchbar und muss wiederholt werden.

Initialisierung der Rektifizierung

Wurden die Kameras erfolgreich kalibriert, wird beim ersten Aufruf der Funktion *updateImages()* die Funktion *initRectification()* aus dem StereoController ausgeführt, welche die Berechnung der Distorsionskorrektur und Rektifizierung durchführt.

Diese ruft zunächst die Funktion *cv::stereoRectify()* auf, welche den Algorithmus von Bouguet zur Berechnung der Rektifizierung verwendet. (siehe 2.4) Mit Hilfe der Kameraparameter beider Kameras und den extrinsischen Parametern werden von ihr die Rotationsmatrizen zur Rektifizierung mit den dazugehörigen Projektionsmatrizen berechnet. Ebenso wird die Reprojektionsmatrix *Q* erstellt, welche später für die Triangulation benötigt wird.

Im Anschluss daran wird mit der Funktion *cv::initUndistortRectifyMap(..)* aus den Rotationsmatrizen und Projektionsmatrizen von *cv::stereoRectify()* für jede Kamera eine Look-up-Map erstellt. In ihr ist hinterlegt, auf welche Bildkoordinaten ein Pixel aus dem originalen Kamerabild abgebildet werden muss, damit ein distorsionsfreies, rektifiziertes Bild entsteht. Das erspart die ansonsten bei jedem Korrekturvorgang in *updateImages()* notwendige und aufwendige Berechnung der Zielkoordinaten für jedes Pixel.

Zurücksetzen der Kalibrierung

Ein Zurücksetzen der Kalibrierung wird notwendig, wenn zum einen die Kalibrierung unbrauchbar ist, oder neu kalibriert werden muss, weil z.B. die Auflösung verändert werden soll. Das Zurücksetzen der Kalibrierung übernimmt die Funktion *resetCalib()* aus dem StereoController. Dabei wird der Kalibrierungsstatus des StereoControllers auf **false** gesetzt. Die Matrizen mit den intrinsischen Parametern und Distorsionskoeffizienten sowie die extrinsischen Parameter *R* und *T* mit einer leeren Matrix überschrieben, damit sie nicht bei der folgenden Kalibrierung von den OpenCV Funktionen als geschätzte Eingabeparameter verwendet werden.

Die Funktion *clearChessboardList()* ermöglicht ein kleines Reset, falls mit der Aufnahme von Schachbrettern bereits begonnen wurde, der Vorgang aber wiederholt werden soll weil beispielsweise nochmals die Auflösung verändert wird. Sie ruft dabei die Funktion *clearPointList()* des StereoCalibrator auf, welche die Vektoren mit den Bild- und Objektkoordinaten leert.

5.1.3. Matching

Wie aus dem UML-Diagramm in Abbildung 5.2 ersichtlich wird, handelt es sich bei der Klasse *StereoMatcher* um eine abstrakte Klasse. Sie stellt ein Interface für das Matching bereit.

Die von *StereoMatcher* erbenenden Klassen *StereoMatcherBM* und *StereoMatcherSGBM* implementieren dieses Interface mit dem Block-Matching Algorithmus (BM) bzw. mit dem Semi-Global-Block-Matching Algorithmus (SGBM). Die Funktionsweise beider Algorithmen ist in Kapitel 3.2.4 und 3.2.5 beschrieben.

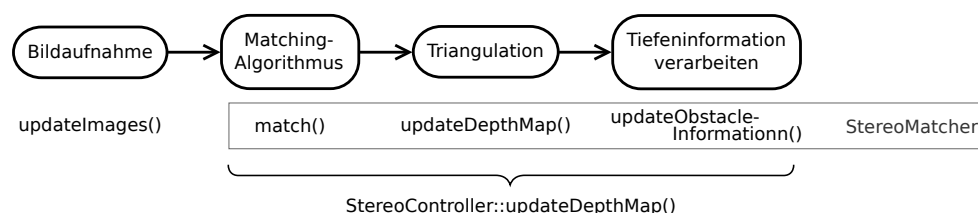


Abbildung 5.4.: Übersicht über wichtige Funktionen des Stereo-Vision Systems zur Kalibrierung

Die wichtigsten Funktionen der Klasse *StereoMatcher* und die dazugehörigen Blöcke des Konzeptes (siehe Kapitel 4.3) sind in Abbildung 5.4 dargestellt.

Ausführen des Matching

Eine zentrale Rolle spielt die abstrakte Funktion *match(Mat leftImage, Mat rightImage)*. Sie berechnet die Disparitätenkarte mittels Korrespondenzanalyse aus dem linken und rechten Kamerabild. Diese wird für die spätere Verarbeitung als Variable in *StereoMatcher* abgespeichert. Im Falle der Klasse *StereoMatcherBM* kommt in *match()* eine Instanz der OpenCV-Klasse *cv::StereoBM* zur Anwendung, mit deren Operator *StereoBM::operator()* das Matching ausgeführt wird. Die Funktion *match()* der Klasse *StereoMatcherSGBM* verwendet analog dazu eine Instanz der OpenCV-Klasse *cv::StereoSGBM()*.

Die erzeugte Disparitätenkarte ist in beiden Fällen eine *cv::Mat* Matrix vom Typ *CV_16S*. Das bedeutet, dass die Disparitäten nicht als Gleitkommazahl, sondern als 16Bit *short* Werte. Zudem sind die Disparitäten mit dem Faktor 16 skaliert. Um die wahren Disparitäten als Gleitkommazahl zu erhalten, müssen diese folglich durch 16 geteilt werden. Daher kommt auch die maximale Auflösung der Disparitäten von $\frac{1}{16} px$.

Erstellung der Tiefenkarte

Aus der Disparitätenkarte wird im nächsten Schritt die Tiefenkarte berechnet. Das geschieht per Triangulation (siehe Kapitel 2.5) mit der Formel

$$Z = \frac{fT}{d}$$

mit der Tiefe Z , der Brennweite f , der Basislinie T und der Disparität d aus der Disparitätenkarte. Die beiden Parameter f und T werden der Reprojektionsmatrix Q entnommen, welche aus der Rektifizierung (siehe Kapitel 2.4) stammt. Q hat dabei folgenden Aufbau

$$Q = \begin{pmatrix} 1 & 0 & 0 & -c_{lx} \\ 0 & 1 & 0 & -c_{ly} \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{1}{T_x} & 0 \end{pmatrix}$$

Mittels der Funktion *setupTriangulation(Mat Q)* werden f und $T = \frac{1}{T_x}$ nach der Kalibrierung als Klassenvariablen von *StereoMatcher* gesetzt, damit ein einfacher, von Instanzen unabhängiger Zugriff erfolgen kann.

Die Berechnung der Tiefenkarte erfolgt mit der in *StereoMatcher* implementierten Funktion *updateDepthMap()*. Diese legt zunächst eine leere Tiefenkarte an, mit der Größe der Disparitätenkarte. Als Typ der *cv::Mat* Matrix wird *CV_32FC1* verwendet, das bedeutet eine 1-Kanal Matrix mit 32Bit Gleitkommazahlen als Datentyp. Dann wird für jedes Pixel die Tiefe mit obiger Formel berechnet und in die Tiefenkarte geschrieben. Dabei müssen die Disparitäten vorher durch den Faktor 16 geteilt werden. (siehe oben)

Darstellung der Tiefeninformation

Als letzter Schritt des Matchings erfolgt die Verarbeitung der Tiefeninformation. Dazu wird die Tiefenkarte in Zeilen aufgeteilt und jede Zeile separat in verschiedene viele Spalten. Innerhalb

eines jeden, auf diese Weise entstehenden, rechteckigen Bereichs der Tiefenkarte wird die geringste, plausible Entfernung bestimmt.

Um die Aufteilung der Karte flexibel zu gestalten, die Repräsentation der Daten dabei aber möglichst kompakt zu halten, wurde folgende Datenstruktur verwendet:

Die Konfiguration des Gitters ist in einem Vektor `std::vektor<unsigned int> rowConfigObstacles` gespeichert. Die Anzahl der Werte im Vektor (`rowConfigObstacles.size()`) gibt an, in wie viele Zeilen die Tiefenkarte aufgeteilt wird. Jeder Wert, welcher sich im Vektor `rowConfigObstacles` befindet, definiert die Anzahl der Spalten in der entsprechenden Zeile.

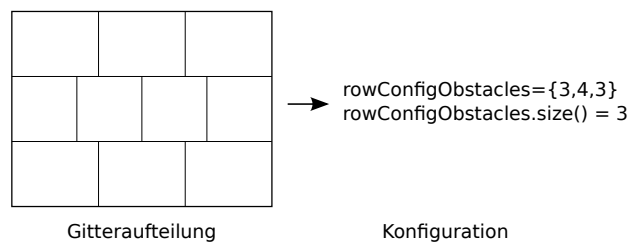


Abbildung 5.5.: Beispiel für Konfiguration der Aufteilung der Tiefenkarte

Die Tiefeninformation für jeden Bereich wird als Gleitkommazahl fortlaufend in einem Vektor `std::vector<double> obstacles` geschrieben. Mit Hilfe der Werte in `rowConfigObstacles` kann auf die Tiefe des gewünschten Bereichs in `obstacles` zugegriffen werden.

Verarbeitung der Tiefenkarte

Die Verarbeitung der Tiefenkarte wird durch die Funktion `getObstacleInformation()` durchgeführt.

Zunächst wird von ihr der Bereich der Tiefenkarte ermittelt, welcher keine Information enthält. Beide verwendeten Stereo-Algorithmen verursachen einen Balken auf der Tiefenkarte mit der Breite von *NumberOfDisparities* ohne Tiefeninformation. Bei der Verarbeitung der Tiefenkarte wird dieser Bereich nicht berücksichtigt.

Die Ursprüngliche Implementierung teilte dann wie oben beschrieben, die Tiefenkarte in mehrere Segmente. Innerhalb jedes Segmentes wurde zunächst die minimale und maximale Entfernung bestimmt. Lagen die minimale und maximale Entfernung in diesem Bereich dicht beieinander (< 20 cm) wurde der Mittelwert der Tiefen des gesamten Segmentes berechnet. Anderenfalls wurde ein Histogramm mit 30 Klassen für das Segment erstellt, wobei nur die kleinsten 5% der vorkommenden Entfernungen berücksichtigt wurden. Aus den 30 Klassen wurde diejenige mit der größten Häufigkeit ausgewählt und als Tiefenentfernung für das untersuchte Segment angenommen. Leider hat sich bei der Evaluierung herausgestellt, dass das beschriebene Verfahren zur Verarbeitung der Tiefenkarte nicht robust gegenüber Ausreißern ist. Ein einzelner sehr niedriger falscher Messwert führt dazu, dass eine viel zu geringe Entfernung angenommen wird.

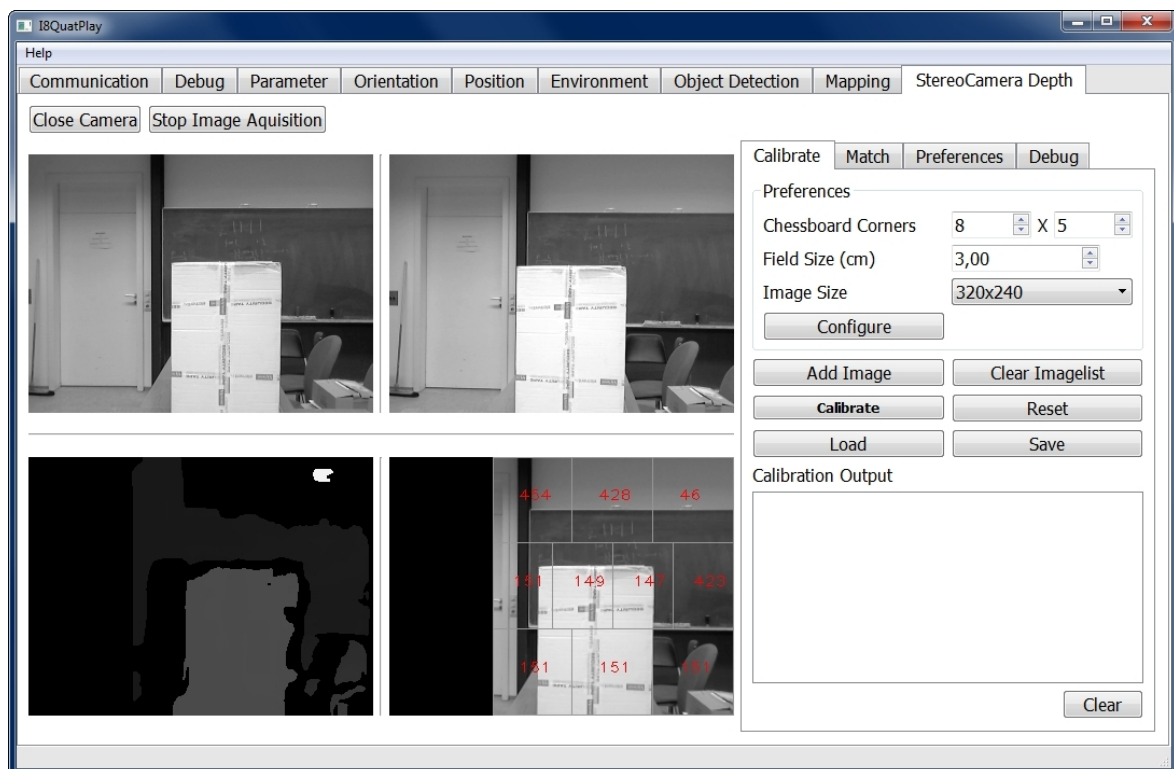
Stattdessen wird nun für jedes Segment der Tiefenkarte ein Histogramm mit 600 Klassen über den gesamten Messbereich erstellt. Die 600 Klassen sind dabei in 20 Oberklassen mit je 30 Unterklassen unterteilt. Aus den Großklassen wird im nächsten Schritt die Oberklasse mit den kleinsten Entfernungen ausgewählt, welche einen Mindestanteil an Bildpunkten des Segmentes enthält. Dieser Schwellwert ist als Konstante in der Klasse *StereoMatcher* gespeichert und liegt standardmäßig bei 10 Prozent. Aus der So gewählten Oberklasse wird die Unterklasse mit der größten Häufigkeit heraus gesucht, welche dann die Tiefenentfernung für das Segment darstellt. Ebenso

als Konstanten abgelegt sind die minimale und die maximale durch das Histogramm berücksichtigte Tiefe, damit der Messbereich, wenn erforderlich, eingegrenzt werden kann.

Im Rahmen der Evaluierung in Kapitel 6 wird gezeigt, inwieweit die sich Verbesserungen im Vergleich zur alten Methode ergaben.

5.1.4. GUI/StereoWidget

Übersicht



(a) Oberfläche mit Reiter Calib

Abbildung 5.6.: Übersicht über die grafische Oberfläche des Stereo-Vision Systems

Abbildung 5.6 zeigt die Bestandteile der Grafischen Oberfläche des Stereo-Vision Systems. Sie ist in der Klasse *StereoWidget* implementiert.

Über die Schaltfläche Open Camera / Close Camera werden die Kameras gestartet. Über die Schaltfläche Start Image Aquisition / Stop Image Aquisition wird vom *StereoWidget* ein Timer gestartet, dessen Funktion weiter unten beschrieben wird.

Die vier grafischen Anzeigen in Abbildung 5.6(a) haben folgende Bedeutung: In der Oberen Zeile werden die Kamerabilder der linken und rechten Kamera angezeigt. Wahlweise können statt der unkorrigierten Kamerabilder auch die rektifizierten Bilder angezeigt werden (siehe Abbildung 5.7(b) *show rectified Images*). Die Bilder werden vom StereoController mit der Funktion *getImage(Mat image, flagCamera leftOrRight, bool rectified)* abgerufen. Bei *flagCamera* handelt es sich um ein **enum** mit den möglichen Werten *STEREO_CAM_LEFT* und *STEREO_CAM_RIGHT* mit welcher vom StereoController die beiden Kameras unterschieden werden.

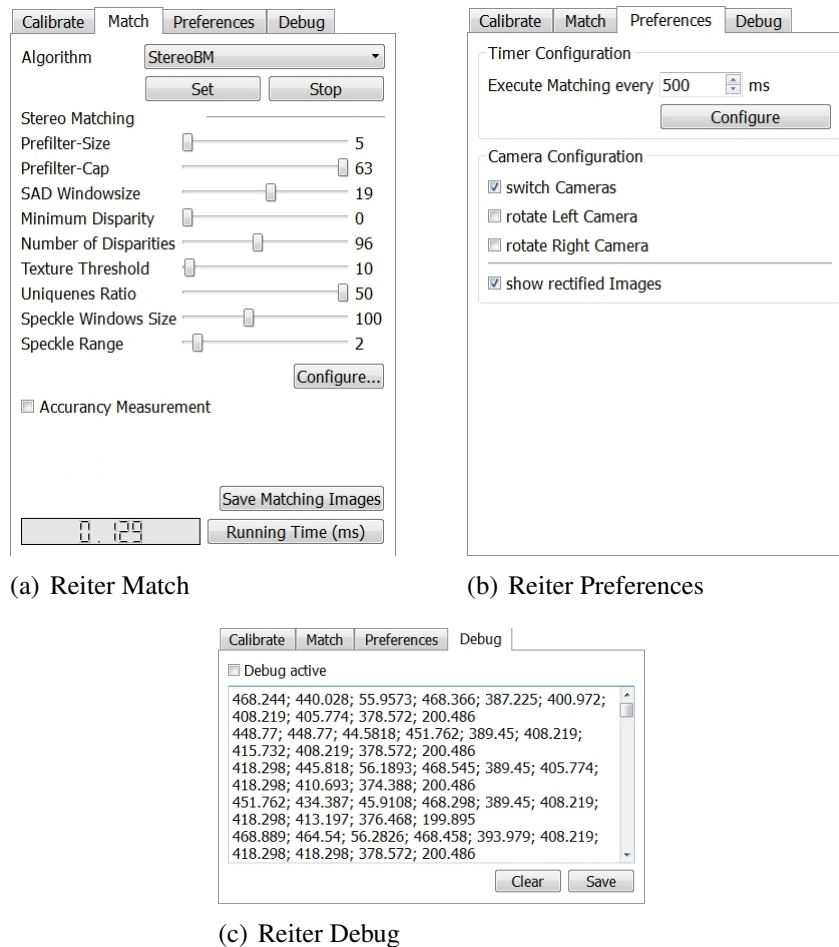


Abbildung 5.7.: Übersicht über die weiteren Reiter der Oberfläche des Stereo-Vision Systems

In der zweiten Zeile wird links eine grafische Darstellung der Tiefenkarte angezeigt. Dazu verwendet die Funktion *getGrayscaleDepthImage(..)* des *StereoMatchers* die durch *match()* erzeugte Disparitätenkarte und normalisiert sie auf einen Wertebereich zwischen 0 und 255. So entsteht ein anzeigefähiges Graustufenbild der Disparitätenkarte bzw. Tiefenkarte. Je heller ein Bereich auf dieser Anzeige, desto näher befindet er sich an der Kamera, je dunkler, desto weiter ist er weg. Schwarz bedeutet, dass die Tiefe in diesem Bereich nicht bestimmt werden konnte.

Das Bild rechts daneben visualisiert die verarbeitete Tiefeninformation. Dazu werden die Rechtecke in das rektifizierte, linke Kamerabild eingezeichnet und jedem Rechteck die zugehörige Tiefeninformation aus *getObstacleInformation()* zugeordnet. Dieses Bild stammt aus der Funktion *getMapWithText()* des *StereoControllers*.

Reiter Calib

Über den Reiter Calib (Abbildung 5.6(a)) lassen sich alle Funktionen der Kalibrierung ausführen, welche bereits in Abschnitt 5.1.2 beschrieben wurden. In das Textfeld Calibration Output werden Rückmeldungen z.B. über den (Miss-)Erfolg einer Schachbrettanalyse geschrieben. So lässt sich abschätzen, wie viele Schachbretter noch für eine Kalibrierung analysiert werden müssen.

Reiter Match

Die Wahl des zum Matchen verwendeten Algorithmus kann über den Reiter Match erfolgen. (Abbildung 5.7(a)) Die Schaltfläche Set setzt über die Funktion *setAlgorithm()* den Stereo-Algorithmus.

Gestartet bzw. gestoppt wird das Matching über die Schaltfläche Start / Stop. Dazu wird die **bool** Variable *isMatchingActive* des *StereoWidget* gesetzt, in Abhängigkeit welcher die Funktion *updateImageAndGui()* (siehe unten) das Matching veranlasst.

Um möglichst optimale Matching-Ergebnisse zu erhalten lassen sich die Stereo-Algorithmen mit der Funktion *configAlgorithm(flagMatcherProperties property, int value)* konfigurieren. *flagMatcherProperties* ist dabei ein **enum** zur Identifikation der richtigen Eigenschaft (siehe UML-Diagramm 5.2). Mit der Schaltfläche werden so die über die Schieberegler eingestellten Parameter an den *StereoMatcher* übermittelt.

Reiter Preferences

Der Reiter Preferences (Abbildung 5.7(b)) erlaubt zum einen die Einstellung des Timers zum aktualisieren der Bilder.

Zum anderen kann die Geometrische Anordnung der Kameras korrigiert werden. Die dazugehörigen Funktionen des *StereoControllers* wurden bereit in Abschnitt 5.1.1 beschrieben.

Steuerung des zeitlichen Programmablauf und Kommunikation mit Quadrocopter

Die Steuerung des zeitlichen Programmablaufes wird von der Klasse *StereoWidget* übernommen. Dazu wird von einem Timer die Funktion *updateImageAndGui()* aufgerufen. Diese gibt den Befehl zum aktualisieren der Kamerabilder mit *updateImages()* und zum Ausführen des Matchings mit *updateDepthMap()* an den *StereoController*. Anschließend konvertiert sie für die Anzeige die neuen Bilder des *StereoControllers* ins *QImage*-Format.

Die Kommunikation mit dem Quadrocopter und der Steuerungssoftware wird auch von dieser Funktion übernommen. Dazu wird die Tiefeninformation mittels *getObstacleInformation()* vom *StereoController* abgerufen und über das Nachrichtensystem der AQopterI8 Software an deren Telemetrie-Display gesendet. Alternativ können die Tiefeninformation auch direkt an den Quadrocopter gesendet werden. Zudem werden die Entfernungen auch über das Textfeld des Reiters Debug angezeigt. (Abbildung 5.7(c))

5.2. Hardware - Kameras

Als Bildquelle für das Stereo-Vision System dienen zwei Logitech C270 Webcams. Diese sind auf einer Halterung am Quadrocopter montiert. (Siehe Abbildung ...). Das Gehäuse wurde entfernt, um eine stabile Montage zu ermöglichen. Der Abstand zwischen den Sensoren beträgt in etwa 10 cm. Der genaue Wert wird durch die Kalibrierung ermittelt.

Die Kameras werden über USB an den Computer des Quadrocopters angeschlossen und per plug and play vom Betriebssystem erkannt. Spezielle Treiber für die Kameras sind nicht notwendig. Einige Technische Daten sind in Anhang B.1 aufgelistet.

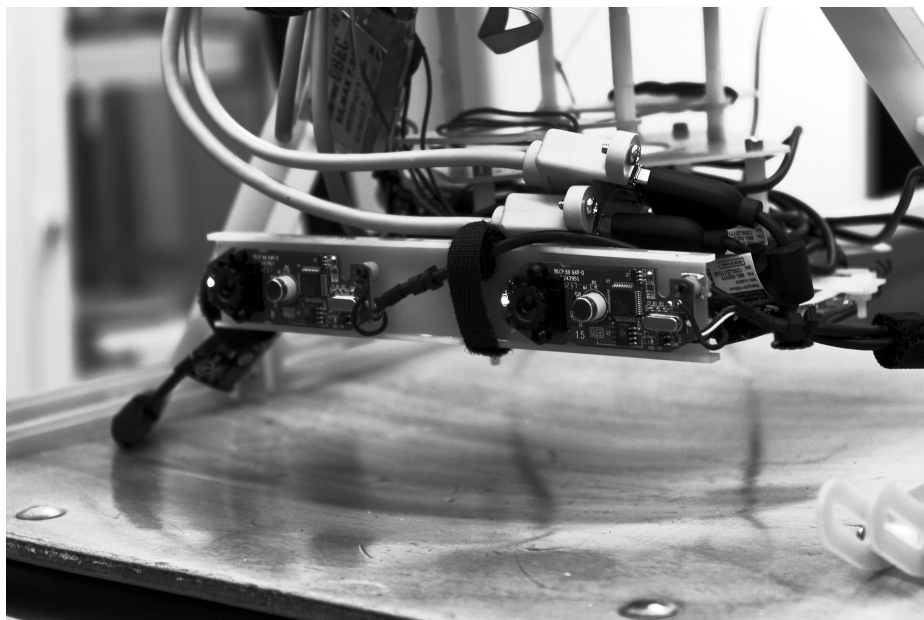


Abbildung 5.8.: Am Quadrocopter montierte Logitech C270 Webcams

6. Evaluierung

6.1. Überblick

6.2. Robustheit der Tiefenauswertung

Gerade im Hinblick auf die Verwendung der Tiefeninformation zur Hindernisvermeidung ist es wichtig, dass deren Auswertung robuste Ergebnisse liefert. In der folgenden Messreihe treten die Probleme der ursprünglich verwendeten Methode zur Tiefenauswertung zu Tage. Daher wurde diese nochmals überarbeitet. Die Verbesserung im Vergleich zur alten Methode werden in der zweiten Messreihe des Abschnittes gezeigt.

6.2.1. Robustheit der ursprünglich verwendeten Methode

Versuchsaufbau

Um die Robustheit der Tiefenauswertung einschätzen zu können wurde folgender Versuch durchgeführt. Ein Karton wurde auf einem Tisch, beginnend bei 400 cm Entfernung, langsam in Richtung der Kameras verschoben. Dabei wurde die Verschiebung alle 10 cm für einige Sekunden angehalten. Die Entfernung des Kartons zur Kamera wurde dabei kontinuierlich vom Stereo-Vision System erfasst und in einem Graphen aufgetragen. (Siehe Abbildung A.1 und A.2) Er zeigt den zeitlichen Verlauf der Messung. Der Versuch wurde je einmal für den SGBM und den BM durchgeführt. Ausgewertet wurde ein Segment der Tiefenkarte, auf welchem der Karton durchgängig abgebildet war.



Abbildung 6.1.: Versuchsaufbau zur Messung der Robustheit

Ergebnisse

Abbildung A.1 zeigt den Verlauf der Messung mit dem SGBM. Es lässt sich erkennen, dass die Entfernung zum Karton grundsätzlich richtig gemessen wurde. Die horizontalen Abschnitte zeigen dabei die Tiefe, bei welcher jeweils der Karton einige Zeit still stand. Abweichungen treten einmal im Bereich größer 300 cm auf. (Zeitbereich bis 4300 s) Hier ist ein deutliches Rauschen der Messwerte zu beobachten. Der Grund dafür ist in der verwendeten Methode zur Tiefenauswertung zu suchen, ebenso wie bei den starken Ausreißern im Zeitbereich um 4500 s. Da mit der ursprünglichen Methode nur die 5% kleinsten im Segment vorkommenden Entfernungen ausgewertet wurden, führt bereits ein einziger falscher Match zu einer falschen Entfernung für das Segment. Die Methode ist also nicht robust.

Beim BM besteht das gleiche Problem, macht sich aber weniger bemerkbar, da die Disparitätenkarte des BM durch Filterung grundsätzlich weniger Information enthält als die des SGBM. Der Graph in Abbildung A.2 zeigt, dass das Rauschen der Messwerte im oberen Tiefenbereich geringer ausfällt als beim SGBM. Mit Ausnahme einiger Werte im Bereich um 500 s treten keine großen Ausreißer auf. Auch diese lassen sich aber auf die mangelnde Robustheit der Tiefenauswertung zurückführen.

6.2.2. Vergleich der ursprünglichen und der korrigierten Methode

In einer Messreihe wurde untersucht, inwieweit sich eine Verbesserung der Robustheit durch die korrigierte Tiefenauswertung im Vergleich zur ursprünglich verwendeten einstellt.

Versuchsaufbau

Eine große Sperrholzplatte wurde mit Werbeprospekten beklebt um gute Voraussetzungen für das Matching zu schaffen. Die Entfernung zwischen Kamera und Holzplatte wurde dann in einer Entfernung von 400 cm aufgestellt und 100 mal die Tiefe gemessen. Bis zu einer Entfernung von 200 cm wurde die Messung in 10 cm Schritten wiederholt, jeweils mit der alten und der neuen Methode zur Tiefenauswertung. Für den Test wurde der SGBM ausgewählt, da er im vorangegangenen Versuch deutlich mehr Probleme mit der Robustheit verursacht hat. Zum Einsatz kam in dieser Messung zwei Logitech C310 Kameras mit einer Auflösung von $320 \times 240 \text{ px}$ einer Brennweite $f = 476.15 \text{ px}$ und einer Basislinie $T = 9.97 \text{ cm}$.



Abbildung 6.2.: Versuchsaufbau aus Sicht der linken Kamera

Ergebnisse

Der Graph in Abbildung A.3 zeigt ein Vergleich der alten und der neuen Methode. Die dazugehörigen Messwerte sind in Tabelle A.1 aufgeführt. Es ist deutlich erkennbar, dass die Kurve der neuen Methode im Bereich größer 250 cm näher an der Referenzmessung verläuft. Die Abweichung vom wahren Wert lag in diesem Bereich mit einer Ausnahme immer unterhalb von 10

cm. Im Vergleich dazu lag diese bei der ursprünglichen Methode im Bereich größer 300 cm fast immer zwischen 15 und 20 cm.

Im Bereich zwischen 200 und 250 cm erzielt die alte Methode etwas bessere Ergebnisse als die neue. Das lässt sich zum einen dadurch erklären, dass alle Tiefenwerte im Segment sehr nah beieinander liegen und deshalb von der alten Methode der Mittelwert aus allen Tiefenwerten des Segmentes berechnet wurde. Zum anderen hat die neue Methode aufgrund der Histogrammberechnung eine geringere Tiefenauflösung. Bei 600 Klassen im Bereich 50 cm bis 600 cm (verwendete Obergrenze für die Messung) ergibt sich eine Breite von 1.1 cm pro Klasse. Insgesamt ist die Abweichung der neuen Methode vom wahren Wert zwischen 200 und 250 cm mit Werten zwischen 2 und 5 cm immer noch sehr gering.

Durch Betrachten der Standardabweichungen der Messwerte von alter und neuer Methode zeigt sich, dass die Messwerte der neuen Methode weniger streuen als die alte. Das zeigt, dass die Robustheit zugenommen hat.

Zusammenfassend lässt sich sagen, dass die neue Methode zur Tiefenauswertung Verbesserungen in Bezug auf Abweichungen vom wahren Wert in größeren Entfernungen sowie bei der Robustheit aufweist.

6.3. Erreichbare Genauigkeit der Stereo-Algorithmen

Versuchsaufbau

In dieser Messreihe wurde der gleiche Versuchsaufbau wie in 6.2.2 verwendet. Der Messbereich erstreckte sich in diesem Fall auf einen Bereich zwischen 400 cm und 60 cm. Für die beiden Stereo-Algorithmen SGBM und BM wurde je eine Messreihe mit 100 Messungen pro Schritt durchgeführt.

Ergebnisse

Die Verwendung beider Stereo-Algorithmen erlaubt eine gute Tiefenauswertung im Bereich zwischen 60 cm und 400 cm. Wie in Abbildung A.4 zu sehen und aus Tabelle A.2 ersichtlich, ist die Abweichung zur manuellen Messung mit einem Maßband gering. Im Fall des SGBM liegt sie bei unter 2 cm im Bereich bis 200 cm, im Bereich bis 300 cm unter 5 cm und im Bereich bis 400 cm mit einer Ausnahme unter 10 cm. Der BM erlaubt noch genauere Messungen, hier liegt die maximale Abweichung mit zwei Ausnahmen immer unterhalb von 2 cm im Vergleich zur manuellen Messung.

Auffallend ist, dass die Messwerte für beide Algorithmen wenig streuen, wie die Standardabweichung in Tabelle A.2 zeigt. Das lässt sich mit der verwendeten Methode zur Tiefenauswertung und der Breite einer Klasse im Histogramm erklären. Durch die Einteilung der Werte der Tiefenkarte in Klassen entsteht bereits eine Glättung der Ergebnisse. Außerdem ist über das Histogramm nichts weiter bekannt, als dass die gemessene Tiefe der Klasse mit der maximalen Häufigkeit im Histogrammabschnitt entspricht. Die wirkliche Verteilung kann sich durchaus von Messung zu Messung stärker unterscheiden.

Wie die Messreihe insgesamt zeigt, sind mit dem Stereo-Vision System sehr genaue Entfernungsmessungen möglich, vorausgesetzt die abgebildeten Objekte besitzen genügend Textur.

6.4. Genauigkeit in Abhängigkeit von der Auflösung

Versuchsaufbau

Für den Vergleich zweier unterschiedlicher Auflösungen wurde eine typische Situation in einem Innenraum ausgewählt. (siehe Abbildung 6.3) Die Entfernung der einzelnen abgebildeten Objekte zur Kamera wurde mit einem Maßband gemessen. Anschließend wurde für 2 verschiedene Auflösungen 100 Messwerte mit dem BM aufgenommen und die einzelnen Segmente ausgewertet. Die beiden Auflösungen $320 \times 240 \text{ px}$ und $192 \times 144 \text{ px}$ wurden gewählt, da sie durch Kalibrierung und Rektifizierung einen nahezu identischen Ausschnitt der Situation zeigen. Eine Kalibrierung und Rektifizierung weiterer Auflösungen mit vergleichbarem Bildausschnitt schlug leider fehl.

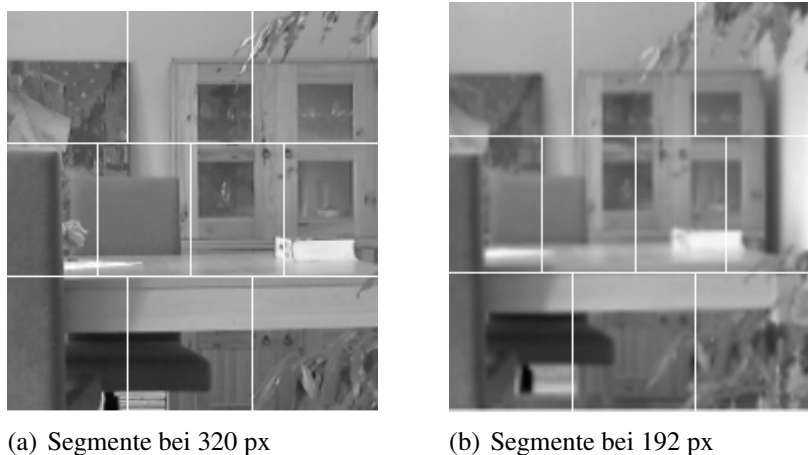


Abbildung 6.3.: zur Tiefenkarte gehörende Bildausschnitte mit Einteilung in die Segmente in Abhängigkeit der Auflösung

Ergebnisse

Unter Verwendung beider Auflösungen werden die Tiefen der abgebildeten Objekte von der Größenordnung her gut erkannt. Während bei einer Auflösung von 320 px relativ gut mit der Referenzmessung übereinstimmen, wird bei einer Auflösung von 192 px die Tiefe von entfernteren Objekten überschätzt, z. B. der Stuhl hinter dem Tisch um 20 cm. Die in den Tabellen A.3 und A.4 ebenfalls aufgeführte Standardabweichung weist darauf hin, dass bei beiden Auflösungen die Tiefenentfernung der Pflanze nicht immer erkannt wurde und deshalb vereinzelt die Entfernung des dahinter befindlichen Schrankes gemessen wurde. Ähnlich verhält es sich mit dem Stuhl im vorderen Bildbereich bei einer Auflösung von 192 px .

Zusammenfassend lässt sich sagen, dass unter beiden Auflösungen Messungen der Tiefe möglich sind welche von der Größenordnung her in Ordnung sind. Ist nur die ungefähre Tiefenentfernung interessant, bietet sich eine Auflösung von 192 px an, da eine geringere Laufzeit des Matchings zu erwarten ist. Zu bemerken ist allerdings, dass die Durchführung einer Kalibrierung der Kameras bei einer Auflösung von 192 px schwer ist, da das Schachbrett häufig nicht erkannt wird.

6.5. Auswirkung von Lichtverhältnissen

Werden Kameras als Sensorsystem verwendet, besteht immer eine Abhängigkeit von herrschenden Lichtverhältnissen und Messergebnis. Nachfolgend wird untersucht, wie weit sich die Helligkeit in einem Raum reduzieren lässt um noch aussagekräftige Messwerte zu erhalten.

6.5.1. Versuchsaufbau

Dazu wurde eine, in Abbildung 6.4 gezeigte, statische Szene, mit Objekten in unterschiedlichen Entfernungen zur Kamera aufgebaut. Um einschätzen zu können, ob später die Messwerte für jedes Segment der Tiefenkarte plausibel sind, wurde eine Referenzmessung mit einem Maßband durchgeführt. Dabei wurde die Tiefe solcher Flächen in der Szene gemessen, welche später vom Algorithmus bei der Tiefenverarbeitung ausgewählt werden sollte.

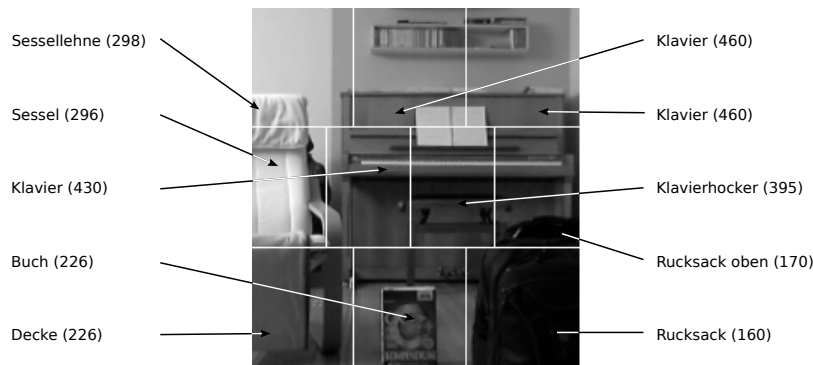


Abbildung 6.4.: Flächen mit erkannter, kürzester Tiefenentfernung und deren Referenztiefe

Die unterschiedlichen Beleuchtungssituationen wurden durch eine 300 Watt Dimmlampe erzeugt.

Es wurden vier Messungen pro Stereo-Algorithmus durchgeführt, zunächst bei voller Leistung der Dimmlampe. Je Messung wurden 100 Messwerte für jedes der 10 Segmente der Tiefenkarte aufgenommen und Mittelwert \bar{z} und Standardabweichung σ_z berechnet. Da nur die Größenordnung der Abweichung von der Referenz wichtig ist, wurde auf 0 Nachkommastellen gerundet. Die Messergebnisse sind in den Tabellen A.5 und A.6 aufgeführt. Die zur jeweiligen Messung gehörenden Bilder der linken Kamera finden sich in Abbildung A.5 und A.6.

6.5.2. Ergebnisse

Insgesamt wurden die Entfernungen von beiden Algorithmen in den meisten Messungen korrekt bestimmt. Einige Auffälligkeiten müssen allerdings näher betrachtet werden.

Schwankungen in der Beleuchtung werden bereits durch die Kameras bis zu einem gewissen Grad kompensiert. Zwischen den Bildern in Abbildung A.6(a) und A.6(b) sind fast keine Unterschiede in der Helligkeit zu erkennen, obwohl die Lampe bereits deutlich zurück gedimmt wurde. Das spiegelt sich auch in den Messwerten zu den beiden Bildern in Tabelle A.6 im Anhang wieder.

Auffallend ist auch, dass auch geringe Intensitäten für gute Messungen ausreichen können. Das Buch wird durch den SGBM beispielsweise selbst in Bild A.5(d) noch korrekt erkannt, wie die Messwerte zeigen.

Dagegen werden andere, vor allem dunkelfarbige Objekte nicht mehr erkannt. Das betrifft z.B. den schwarzen Rucksack und den Klavierhocker in Abbildung A.5(c) und A.5(d), welchen durch den SGBM fälschlicherweise die Tiefenentfernung des Klaviers zugewiesen wird. Auch für den BM stellt der Rucksack ein Problem dar. Der obere Teil des Rucksackes wird dabei im Bild A.6(d) nur manchmal erkannt. Dadurch lässt sich die hohe Standardabweichung erklären, da in einem Großteil der Fälle stattdessen der Klavierhocker die geringste Tiefe in diesem Segment aufwies.

Aus einem ähnlichen Grund kommt auch die hohe Standardabweichung der Sessellehne in der Messung mit dem SGBM zustande. Einerseits wird die Sessellehne in einigen Fällen nicht erkannt, weshalb die Entfernung zur dahinter liegenden Wand bei der Tiefenverarbeitung ausgewählt wird. Andererseits stellt die Wand eine homogene Fläche dar, weshalb hin und wieder durch den SGBM großflächig falsche Matches berechnet werden.

Zusammenfassend lässt sich sagen, dass durch den Helligkeitsausgleich der Kameras nicht optimalen Lichtverhältnissen gute Ergebnisse in der Entfernungsmessung in statischen Szenen erzielt werden können. Aussagen über Auswirkung von Lichtverhältnissen im Flug lassen sich ohne einen separaten Test allerdings nicht machen.

6.6. Auswirkung der Parameter der Stereo-Algorithmen und deren Optimierung

Die Wahl der Parameter der beiden Stereo-Algorithmen beeinflusst unter anderem die Genauigkeit der Messwerte, die Laufzeit und die minimal messbare Entfernung. Im Folgenden wird die Auswirkung der einzelnen Parameter untersucht, um anschließend eine Optimierung durchführen zu können.

Bei allen Messreihen dieses Abschnittes wurde dieselbe Kalibrierung verwendet. Sie berechnete bei einer Auflösung von $320 \times 240 \text{ px}$ eine Basislinie $T = 9.97 \text{ cm}$ und eine Brennweite $f = 451.87 \text{ px}$.

6.6.1. *NumberOfDisparities*

Durch den Parameter *NumberOfDisparities* (Im Programm abgekürzt mit *nDisparities*) wird die Breite des Suchbereichs festgelegt. Er bestimmt wie viele Pixel des rechten Bildes auf Korrespondenz zu einem Pixel des linken Bildes untersucht werden. Dadurch wird die maximal bestimmbar Disparität nach oben beschränkt und damit die minimale noch messbare Tiefenentfernung beeinflusst. Zudem hat die Größe des Suchbereichs Auswirkung auf die Laufzeit der Algorithmen. Des weiteren geht aufgrund der Implementierung der Algorithmen am linken Rand der Tiefenkarte ein Bereich verloren, wodurch das Sichtfeld eingeschränkt wird.

Einfluss auf die minimal messbare Entfernung

Versuchsaufbau Ein Objekt, in diesem Fall ein großer Karton, wird solange in Richtung der Kameras verschoben, bis es gerade nicht mehr erkannt wird. Dann wird durch kleine Verschiebungen versucht die letzte noch messbare Entfernung zu finden. Das geschieht mit Hilfe des ausgegebenen Tiefenbildes. Wird dieses im Bereich des Objektes schwarz oder weist eine große Anzahl stark unterschiedlicher Tiefen auf (Flecken im Bild), so befindet sich das Objekt zu nahe an der Kamera und kann nicht mehr erkannt werden.

Ergebnisse Die Ergebnisse der Messung der minimalen Entfernung in Abhängigkeit von *NumberOfDisparities* sind in Tabelle A.7 aufgeführt. Abbildung A.8 stellt diese Grafisch dar. Werte für *NumberOfDisparities* kleiner 64 px sind in der Praxis nicht sinnvoll, da die minimale Entfernung mit etwas einem Meter und darüber hinaus zu groß ist, für einen Einsatz zur Hindernisvermeidung. Ähnlich verhält es sich mit Werten größer 96 px. Aufgrund der umgekehrt proportionalen Abhängigkeit von Disparität und Tiefenentfernung (siehe 2.5) ist der Zugewinn an Tiefe im Vergleich zum weiter unten betrachteten Bildverlust gering. Die minimal messbare Entfernung ist abhängig von Brennweite und Basislinie des Stereokamerassystems. Die minimal messbare Entfernung kann daher je nach Kalibrierung und Rektifizierung etwas variieren.

Einfluss auf die Laufzeit

Für die einzelnen Werte von *NumberOfDisparities* wurde die Laufzeit für beide Algorithmen über 20 Messungen gemittelt. Dabei ergaben sich die in Tabelle A.7 aufgeführten und in Abbildung A.7 dargestellten Ergebnisse.

Dabei zeigt sich ganz deutlich, dass die Laufzeit des SGBM bereits ab einer maximalen Disparität von 32 px hundert Millisekunden übersteigt, während der BM selbst bei 128 px noch unterhalb dieser Marke bleibt. Die Laufzeit des SGBM beträgt in etwa das 2.5-fache des BM, was dadurch zu erklären ist, dass der SGBM mehr Informationen pro Pixel verarbeitet als der BM.

In Bezug auf die Laufzeit hat der BM deutliche Vorteile gegenüber dem BM

Verlust von Bildfläche

Sowohl der BM als auch der SGBM sind in OpenCV so implementiert, dass Pixel aus dem linken Bild erst ab einer x-Koordinate von *NumberOfDisparities* gematcht werden. Das kommt daher, dass korrespondierende Pixel im rechten Bild links von der gleichen x-Koordinate gesucht werden.

Das führt dazu, dass im Bereich zwischen $x = 0$ und $x = \text{NumberOfDisparities}$ keine Entfernungen gemessen werden. Dieser Abschnitt der Tiefenkarte enthält demzufolge keine Information. Dadurch reduziert sich der effektiv nutzbare, horizontale Öffnungswinkel der Kameras.

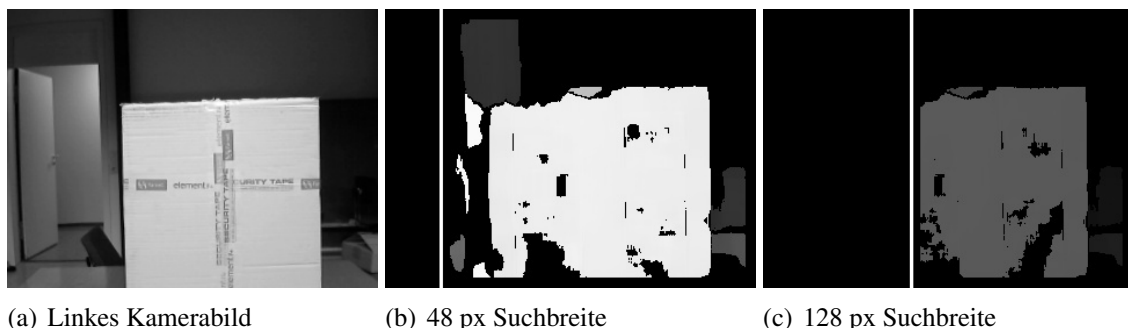


Abbildung 6.5.: Auswirkung von *NumberOfDisparities* auf die Tiefenkarte

Abbildung 6.5 veranschaulicht dieses Problem. Der abgebildete Karton befindet sich dabei in einer Entfernung von 100 cm vor der Kamera. Die weiße Linie in den Tiefenbildern zeigt den Einflussbereich von *NumberOfDisparities*. Bei einer Auflösung von 320×240 px entspricht eine

Stufenbreite von *NumberOfDisparities* von 16 px etwa 5% der Bildbreite. Mit jeder Erhöhung von *NumberOfDisparities* gehen weitere 5% des Bildes und somit auch des Öffnungswinkels verloren. Bei 128 px ergibt sich bereits ein Verlust von 40%, weshalb ein noch höherer Wert für *NumberOfDisparities* bei dieser Auflösung nicht sinnvoll erscheint.

Zusammenfassung der Auswirkungen von *NumberOfDisparities*

Zusammenfassend lässt sich sagen, je höher der Wert für *NumberOfDisparities*, desto besser in Bezug auf die minimale Entfernung, jedoch desto nachteiliger in Bezug auf Laufzeit und Verkleinerung des Öffnungswinkels. Als Kompromiss bieten sich die Werte 80 px oder 96 px an. Dem Block-Matching Algorithmus ist aufgrund der Laufzeit der Vorzug zu geben.

6.6.2. *MinDisparities*

Der Parameter *MinDisparities* (im Programm mit *MinDisp* abgekürzt) hat ähnliche Auswirkungen wie der Parameter *NumberOfDisparities*. Da durch eine Erhöhung von *MinDisparities* der messbare Tiefenbereich nach oben begrenzt wird, wurde er konstant bei 0 gehalten und nicht näher untersucht. Interessant ist er nur, falls lediglich ein stark eingegrenzter Tiefenbereich gemessen werden soll.

6.6.3. *UniquenessRatio*

Die Aufgabe der *UniquenessRatio* wurde bereit in Kapitel 3.2.4 beschrieben.

Beim SGBM treten, verursacht durch homogene Flächen, wie weiße Wände, gelegentlich großflächig falsche Matches auf. Bei dem in Abbildung A.9 gezeigten Versuchsaufbau traten falsche Matches vor im Bereich der Wand oberhalb der Tafel auf. Mit Hilfe der *UniquenessRatio* konnten die falsch gematchten Bereiche entfernt werden. Abbildung A.10 zeigt Die Tiefenkarte für eine *UniquenessRatio* von 0, 10, 20 und 50. Bei einem Wert von 50 treten die falsch gematchten Bereiche zwar nicht mehr auf, dafür gehen allerdings auch korrekt gematchte Details der Tiefenkarte verloren.

Da der BM von vorne herein nur dort im Bild Korrespondenzen findet, wo genügend Textur vorhanden ist, treten bei ihm die Probleme mit großflächig falschen Matches selten auf. Daher ist eine wesentlich geringere *UniquenessRatio* erforderlich als beim SGBM um alle falschen Matches zu entfernen. Im Versuch reicht bereits ein Wert von 21 aus. (Siehe Abbildung A.11)

6.6.4. *SADWindow*

Generell gilt, je größer das *SADWindow*, desto größer ist die Wahrscheinlichkeit, dass ein korrekter Match erfolgt. Dafür gehen mehr Details der Disparitätenkarte verloren. Abbildung A.12 zeigt die Auswirkungen des Parameters *SADWindow* auf die Disparitätenkarte. Dabei profitiert der BM deutlich sichtbar von größeren Werten für *SADWindow*. Nachteilig ist, dass beim BM die Disparitätenkarte am Rand um die Größe des *SADWindow* verkleinert wird und somit Tiefeninformation verloren geht. Ein Wert von 19 px stellt einen guten Kompromiss dar.

Da der SGBM nicht allein auf dem Vergleich von *SADWindows* basiert (siehe 3.2.5, reichen kleinere Werte für *SADWindow* aus. Mit größeren Fenstern gehen mehr Details verloren als dazugewonnen werden kann. Abbildung A.13(d) zeigt, dass ein Wert für *SADWindow* sogar zusätzlich

Fehler verursacht. (senkrechter grauer Balken) Für den SGBM bietet sich ein *SADWindow* von 11 bis 13 px an.

6.7. Zusammenfassung der Ergebnisse

Durch die verschiedenen Messreihen unter Laborbedingungen hat sich gezeigt, dass das aufgebaute Stereo-Vision System in der Lage ist, Entfernungen bis 400 cm mit guter Genauigkeit zu messen. Dabei hat sich herausgestellt, dass der Block-Matching Algorithmus aufgrund von Laufzeitvorteilen und geringfügig höherer Genauigkeit besser für das Stereo-Vision System geeignet. Dazu kommt, dass er zudem weniger Probleme mit falschen Matches bei homogenen Flächen aufweist.

Als Parameter haben sich folgende untersuchte Werte als gut geeignet heraus gestellt:

- NumberOfDisparities: 96 px
- MinDisparities: 0 px
- SADWindowSize: 19 px
- UniquenessRatio: 21 px
- Auflösung: 320x240 px

6.8. Flug mit dem Quadrocopter

Ein abschließender Test des Stereo-Vision Systems erfolgte im Flug. Ziel dabei war es zu überprüfen, ob das System auch unter Flugbedingungen arbeiten kann.

Es zeigte sich, dass das System im Flug grundsätzlich Entfernungen messen kann, welche von der Größenordnung her stimmen. Probleme traten in der Flughalle des Lehrstuhl Informatik VIII der Universität Würzburg vor allem im Bereich einer weißen Wand mit davor hängendem Netz auf. Dagegen wurden durch das Bild laufende Personen gut erkannt, wobei über die Genauigkeit der ermittelten Tiefe keine zuverlässige Aussage gemacht werden kann.

Unabhängig von der Umgebung gibt es auch einige Aspekte, welche Probleme bereiten. Es macht sich bemerkbar, dass die beiden Kameras nicht synchronisiert Bilder aufnehmen. Aufgrund der starken Bewegung des Quadrocopters im Flug macht sich dieser zeitliche Versatz bemerkbar. Das führt dazu, dass sich die Disparitäten im besten Falle nur leicht verändern. Insgesamt wird dadurch das Messergebnis verfälscht.

Ein weiteres Problem liegt in der Laufzeit der Algorithmen. Die berechnete Tiefenkarte enthält die Entfernungen zum Zeitpunkt der Bildaufnahme. Nach der Berechnung hat sich die Position des Quadrocopters allerdings bereits geändert.

7. Diskussion und Ausblick

7.1. Diskussion der Ergebnisse

Die Versuche haben gezeigt, dass das aufgebaute System zur stereo-optischen Entfernungsmessung in dieser Form in der Lage ist, Entfernungen mit einer Genauigkeit zwischen 2 und 5 cm im Bereich bis 4 m zu messen. Der Erfolg einer Messung ist dabei von der Textur des gemessenen Objektes abhängig. Die Arbeit hat auch gezeigt, dass durch die automatische Helligkeitsanpassung der Kameras auch bei wenig Licht korrekte Messungen durchgeführt werden.

Ein Problem besteht noch in der Robustheit der Tiefenauswertung. Im Vergleich zur ursprünglich verwendeten Methode bei der Tiefenauswertung haben sich mit der Auswertung durch ein Histogramm bereits Verbesserungen gezeigt, insbesondere konnte auch der Fehler im Bereich zwischen 350 cm und 400 cm reduziert werden.

Ein weiteres großes Problem besteht in den großflächig auftretenden falschen Matches, wie sie durch Objekte mit zu geringer Textur wie z.B. eine weiße Wand oder sich gleichmäßig wiederholende Muster wie, z.B. ein Netz in der Flughalle, hervorgerufen werden

7.2. Ausblick

Im Rahmen dieser Arbeit wurde ein System zur Entfernungsmessung aufgebaut. Dabei wurde bis jetzt nur die Basisfunktionalität implementiert. Insbesondere fehlt bisher eine Integration der berechneten Tiefeninformationen in die Hindernisvermeidung des Quadrocopters.

Eine differenziertere Auswertung der Tiefenkarte wäre wünschenswert. Nachteil des jetzigen Verfahrens ist, dass ein Großteil des Informationsgehaltes der Tiefenkarte verloren geht, durch eine Reduzierung auf wenige, gemittelte Messwerte. Denkbar wäre z.B. die Tiefenkarte auf zusammenhängende Bereiche gleicher Tiefe hin zu untersuchen. Dazu werden aus der Tiefenkarte verschiedene Wertebereiche durch Thresholding extrahiert, z.B. 50-60 cm, 60-70 cm, ..., 200-250 cm und weitere. So entstehen Kopien der Tiefenkarte, welche nur bestimmte Entfernungen beinhalten. Als nächstes werden die Tiefenkarten nach Konturen abgesucht. Anhand dieser Konturen kann auf Größe und 3D-Position eines Hindernisbereichs geschlossen werden und entschieden werden ob ein Ausweichmanöver gestartet werden muss. Ebenso ließen sich diese 3D-Positionen für Navigation und Pfadplanung weiterverwenden.

Die aufgenommenen Kamerabilder können auch für andere Zwecke weiterverwendet werden. Interessant wäre eine Kombination mit der Objekterkennung, so dass Objekte nicht nur unterhalb des Quadrocopters, sondern auch vor diesem erkannt und genau lokalisiert werden können.

Im Bereich der Implementierung gibt es ebenso einige Verbesserungsmöglichkeiten. Objekte, welche sich in der Nähe der Kamera befinden, weisen eine hohe Disparität auf. Soll die Tiefenentfernung von sehr nahen Objekten messbar sein, muss der Suchbereich entsprechend groß gewählt werden. Dadurch geht ein Teil des linken Bildes verloren in welchem keine Disparitäten gemessen werden. Um diesen Bereich wenigstens zur Messung von größeren Entfernungen nutzbar zu ma-

chen, wäre eine Abwandlung der OpenCV-Implementierung der Stereo-Algorithmen denkbar, so dass im Randbereich der Bilder dynamisch auch kleinere Suchbereiche als *NumberOfDisparities* abgesucht werden.

Ein sehr wichtiger Aspekt ist die Optimierung der Laufzeit der Stereoalgorithmen. Generell ist das Berechnen von Tiefenkarten durch Korrespondenzanalyse sehr rechenintensiv. Eine Möglichkeit wäre, das Matching auf die Grafikkarte auszulagern, da GPUs für solche Aufgaben prädestiniert sind. Durch ihre Struktur sind sowohl der BM-Algorithmus als auch der SGBM-Algorithmus dafür geeignet. Das würde zudem zu einer Entlastung der CPU führen. Die eingesparte Rechenzeit käme anderen Systemen des Quadrocopters zu gute.

Im Hinblick auf mögliche Einsatzszenarios des Quadrocopters ist über den Einsatz anderer Kamerasysteme nachzudenken. Gerade bei schlechten Lichtverhältnissen oder Dunkelheit wären Infrarotkameras eine interessante Option. Fraglich ist, inwieweit eine eventuell notwendige Ausleuchtung durch Infraot-LEDs andere Sensorsysteme des Quadrocopters beeinflusst. Andererseits wäre auch der Einsatz ultraheller LEDs zur Ausleuchtung der Umgebung denkbar um die Matchingergebnisse bei schlechten Lichtverhältnissen weiter zu verbessern.

8. Literaturverzeichnis

- [AQopterI8 2013] AQOPTERI8: *Autonomous Quadrotor for Indoor Exploration*. Website. März 2013. – URL <http://www8.informatik.uni-wuerzburg.de/wissenschaftsforschung/aqopteri8/>
- [Birchfield und Tomasi 1998] BIRCHFIELD, Stan. ; TOMASI, Carlo: Depth Discontinuities by Pixel-to-Pixel Stereo. In: *Proc. Sixth IEEE Int'l Conf. Computer Vision* (1998), Januar, S. 1073–1080
- [Bouguet 2010] BOUGUET, Jean-Yves: *The calibration toolbox for Matlab, example 5: Stereo rectification algorithm*. Website. Juli 2010. – URL http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example5.html. – aufgerufen am 10.08.2013
- [Bradski und Kaehler 2008] BRADSKI, Gary ; KAEHLER, Adrian: *Learning OpenCV - Computer Vision with the OpenCV library*. First Edition. Cambridge, MA : O'Reilly, September 2008. – ISBN 978-0-596-51613-0
- [Davies 2012] DAVIES, E.R.: *Computer and Machine Vision: Theory, Algorithms, Practicalities*. Elsevier, 2012 (Academic Press). – ISBN 978-0-123-86908-1
- [De-Maeztu et al. 2011] DE-MAEZTU, L. ; MATTOCCIA, S. ; VILLANUEVA, A. ; CABEZA, R.: Linear stereo matching. (2011), November 6-13
- [Harris und Stephens 1988] HARRIS, Christopher ; STEPHENS, Mike: A Combined Corner and Edge Detector. (1988)
- [Hartley 1998] HARTLEY, R. I.: Theory and practice of projective rectification. In: *International Journal of Computer Vision* 35 (1998), S. 115–127
- [Hirschmüller 2005] HIRSCHMÜLLER, Heiko: Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 20-26 June 2005, San Diego, CA, USA* 2 (2005), S. 807–814
- [Hirschmüller 2006] HIRSCHMÜLLER, Heiko: Stereo Vision in Structured Environments by Consistent Semi-Global-Matching. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR, New York, NY, USA* (2006), Juni
- [Hirschmüller 2008] HIRSCHMÜLLER, Heiko: Stereo Processing by Semiglobal Matching and Mutual Information. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2008), Nr. 2, S. 328–341. – ISSN 0162-8828
- [Hirschmüller 2011] HIRSCHMÜLLER, Heiko: Semi-Global Matching - Motivation, Developments and Applications. In: *Invited Paper at the Photogrammetric Week in Stuttgart, Germany* (2011), September, S. 173–184

- [Jähne 2012] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 7. Auflage. Springer, 2012. – ISBN 978-3-642-04951-4
- [Konolige 1997] KONOLIGE, Kurt: Small Vision Systems: Hardware and Implementation. In: *Eighth International Symposium on Robotics Research* (1997), S. 111–116
- [Laganière 2011] LAGANIÈRE, Robert: *OpenCV 2 Computer Vision Application Programming Cookbook*. Birmingham, UK : Packt Publishing, Mai 2011. – ISBN 978-1-849-51325-8
- [Logitech 2010a] LOGITECH: *Logitech HD Webcam C270 Technical Specifications*. Website. Oktober 2010. – URL http://logitech-en-amr.custhelp.com/app/answers/detail/a_id/17556/~c270-technical-specifications
- [Logitech 2010b] LOGITECH: *Logitech HD Webcam C310 Technical Specifications*. Website. Januar 2010. – URL http://logitech-en-amr.custhelp.com/app/answers/detail/a_id/17181/~c310-technical-specifications
- [OpenCV 2012] OPENCV, Dev Team: *OpenCV v2.4.0 Documentation*. Website. April 2012. – URL <http://docs.opencv.org/2.4.0/>
- [OpenCV 2013] OPENCV, Dev Team: *OpenCV About*. Website. 2013. – URL <http://opencv.org/about.html>
- [Scharstein] SCHARSTEIN, Daniel: *Middlebury Stereo Evaluation*. Website. – URL <http://vision.middlebury.edu/stereo/eval/>
- [Sturm und Maybank 1999] STURM, P. F. ; MAYBANK, S. J.: On plane-based camera calibration: A general algorithm, singularities, applications. In: *IEEE Conference on Computer Vision and Pattern Recognition* (1999)
- [Wikipedia 2008a] WIKIPEDIA: *Epipolar geometry*. Website. März 2008. – URL http://de.wikipedia.org/wiki/Epipolar_geometry
- [Wikipedia 2008b] WIKIPEDIA: *Triangulation (Messtechnik)*. Website. März 2008. – URL [http://de.wikipedia.org/wiki/Triangulation_\(Messtechnik\)](http://de.wikipedia.org/wiki/Triangulation_(Messtechnik))
- [Zhang 1999] ZHANG, Zhengyou: Flexible camera calibration by viewing a plane from unknown orientations. In: *Proceedings of the 7th International Conference on Computer Vision* (1999), S. 666–673
- [Zhang 2000] ZHANG, Zhengyou: A flexible new technique for camera calibration. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (2000), S. 1330–1334

A. Messergebnisse

A.1. Robustheit der Tiefenauswertung

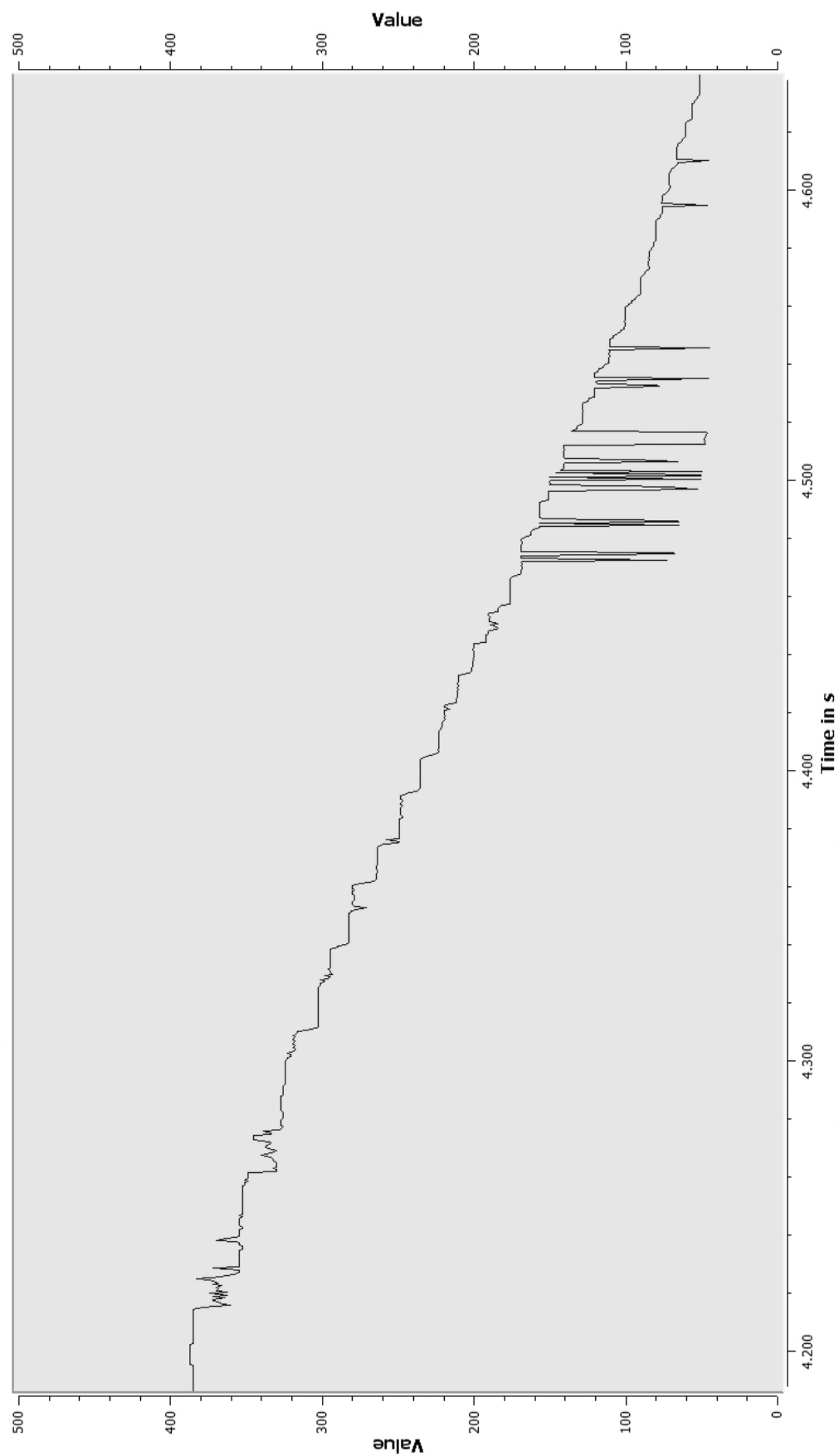


Abbildung A.1.: Robustheit der Tiefenauswertung mit ursprünglicher Methode und SGBM

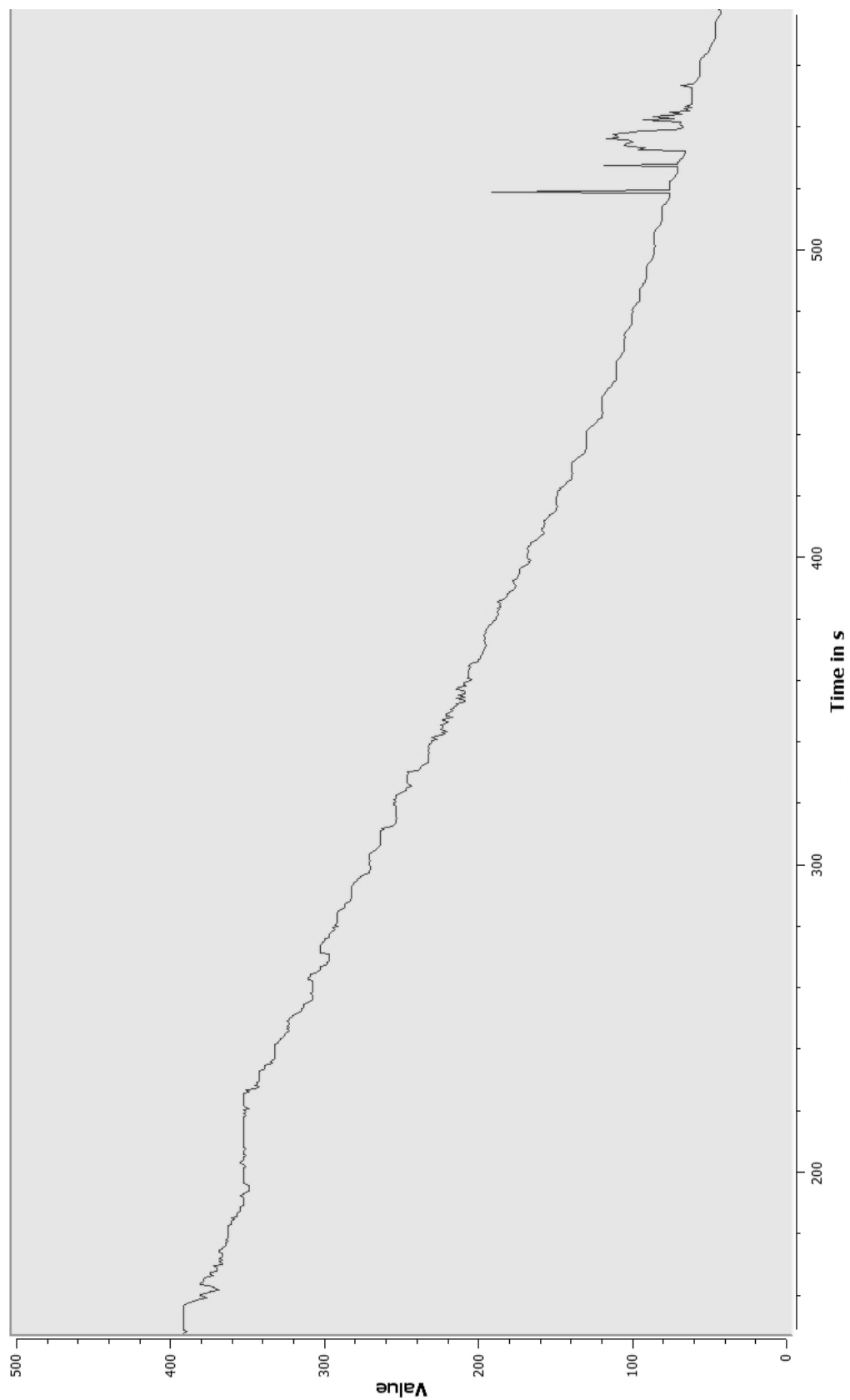


Abbildung A.2.: Robustheit der Tiefenauswertung mit ursprünglicher Methode und BM

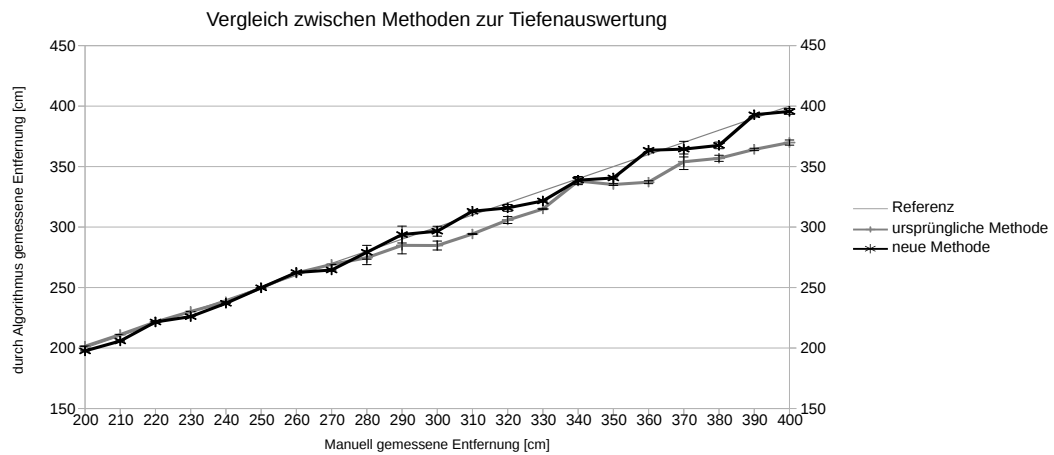


Abbildung A.3.: Vergleich der Tiefenauswertung mit ursprünglicher Methode und neuer Methode

manuelle Messung in cm	Messung mit ursprünglicher Methode in cm			Messung mit neuer Methode in cm		
	Mittelwert	Standardabw.	Abw. von Referenz	Mittelwert	Standard- abw.	Abw. von Referenz
200	201,46	0,06	30,21	197,58	0,00	2,42
210	211,15	0,10	28,19	205,83	0,00	4,17
220	221,57	0,10	29,66	221,55	3,33	1,55
230	230,21	0,10	32,03	226,00	0,00	4,00
240	238,19	0,03	29,12	237,00	0,00	3,00
250	249,66	0,04	24,66	249,83	0,00	0,17
260	262,03	0,08	24,87	262,44	0,40	2,44
270	269,12	0,13	14,75	264,51	0,27	5,49
280	274,66	5,69	14,30	279,17	0,00	0,83
290	284,87	6,98	15,82	293,82	0,09	3,82
300	284,75	3,69	14,96	296,58	0,00	3,42
310	294,30	0,49	28,04	313,08	0,00	3,08
320	305,82	2,87	15,16	315,83	0,00	4,17
330	314,96	0,57	7,11	321,60	1,12	8,40
340	338,04	2,95	13,99	338,75	0,00	1,25
350	335,16	0,73	6,84	340,58	0,00	9,42
360	337,11	1,21	22,89	363,50	0,00	3,50
370	353,99	6,41	16,01	364,42	0,00	5,58
380	356,84	2,53	23,16	367,51	0,85	12,49
390	364,18	0,97	25,82	392,83	0,00	2,83
400	369,84	2,12	30,16	395,58	0,00	4,42

Tabelle A.1.: Vergleich der Tiefenauswertung mit ursprünglicher Methode und neuer Methode

A.2. Vergleich SGBM und BM

manuelle Mes- sung in cm	Messung mit SGBM in cm			Messung mit BM in cm		
	Mittelwert	Standardabw.	Abw. von Referenz	Mittelwert	Standardabw.	Abw. von Referenz
60,00	59,17	0,00	0,83	59,17	0,00	0,83
70,00	68,33	0,00	1,67	70,17	0,00	0,17
80,00	79,33	0,00	0,67	79,33	0,00	0,67
90,00	89,42	0,00	0,58	89,42	0,00	0,58
100,00	98,58	0,00	1,42	100,42	0,00	0,42
110,00	109,58	0,00	0,42	110,50	0,00	0,50
120,00	118,75	0,00	1,25	118,75	0,00	1,25
130,00	130,05	1,15	0,05	129,75	0,00	0,25
140,00	139,13	0,39	0,87	139,83	0,00	0,17
150,00	148,08	0,00	1,92	149,92	0,00	0,08
160,00	158,17	0,00	1,83	159,08	0,00	0,92
170,00	169,17	0,00	0,83	169,40	0,40	0,60
180,00	181,08	0,00	1,08	180,17	0,00	0,17
190,00	189,33	0,00	0,67	190,25	0,00	0,25
200,00	197,58	0,00	2,42	200,21	0,47	0,21
210,00	205,83	0,00	4,17	211,22	0,30	1,22
220,00	221,55	3,33	1,55	221,42	0,00	1,42
230,00	226,00	0,00	4,00	231,50	0,00	1,50
240,00	237,00	0,00	3,00	240,39	0,49	0,39
250,00	249,83	0,00	0,17	249,83	0,00	0,17
260,00	262,44	0,40	2,44	259,93	0,09	0,07
270,00	264,51	0,27	5,49	269,08	0,00	0,92
280,00	279,17	0,00	0,83	281,92	0,00	1,92
290,00	293,82	0,09	3,82	292,02	0,13	2,02
300,00	296,58	0,00	3,42	301,17	0,00	1,17
310,00	313,08	0,00	3,08	309,42	0,00	0,58
320,00	315,83	0,00	4,17	321,30	0,18	1,30
330,00	321,60	1,12	8,40	333,84	0,89	3,84
340,00	338,75	0,00	1,25	343,33	0,00	3,33
350,00	340,58	0,00	9,42	350,29	0,84	0,29
360,00	363,50	0,00	3,50	361,67	0,00	1,67
370,00	364,42	0,00	5,58	371,75	0,00	1,75
380,00	367,51	0,85	12,49	379,08	0,00	0,92
390,00	392,83	0,00	2,83	391,11	0,44	1,11
400,00	395,58	0,00	4,42	401,39	2,15	1,39

Tabelle A.2.: Vergleich der Genauigkeit der beiden Stereo-Algorithmen

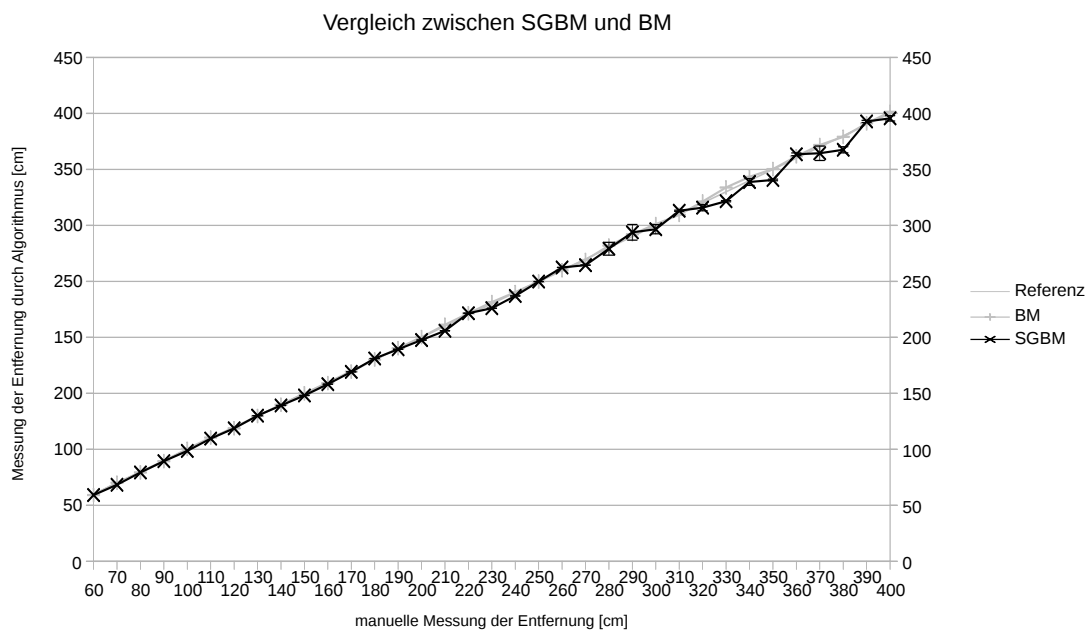


Abbildung A.4.: Vergleich der Genauigkeit von BM und SGBM

A.3. Genauigkeit in Abhängigkeit der Auflösung

Auflösung in px	Algorithmus	<i>Bild</i>	<i>Schrank</i>	<i>Pflanze oben</i>	<i>Stuhl</i>	<i>Stuhl hinten</i>
Referenztiefe in cm		365	325	96	157	270
320x240	BM	364 1	331 1	100 27	159 0	273 3

Auflösung	Algorithmus	<i>Schrank</i>	<i>Buch</i>	<i>Stuhl unten</i>	<i>Tisch</i>	<i>Pflanze</i>
Referenztiefe in cm		325	215	160	170	84
320x240	BM	326 3	216 0	164 0	170 1	85 0

Tabelle A.3.: Tiefenmessung einer Szene mit Auflösung von 320 px

Auflösung in px	Algorithmus	<i>Bild</i>	<i>Pflanze oben</i>	<i>Schrank</i>	<i>Stuhl</i>	<i>Stuhl hinten</i>
Referenztiefe in cm		365	96	325	157	270
192x144	BM	363 2	103 39	333 1	166 22	289 1

Auflösung	Algorithmus	<i>Buch</i>	<i>Schrank</i>	<i>Stuhl unten</i>	<i>Pflanze</i>	<i>Pflanze</i>
Referenztiefe in cm		215	325	160	84	84
192x144	BM	232 2	334 3	169 1	83 27	83 0

Tabelle A.4.: Tiefenmessung einer Szene mit Auflösung von 192 px

A.4. Auswirkung unterschiedlicher Lichtverhältnisse

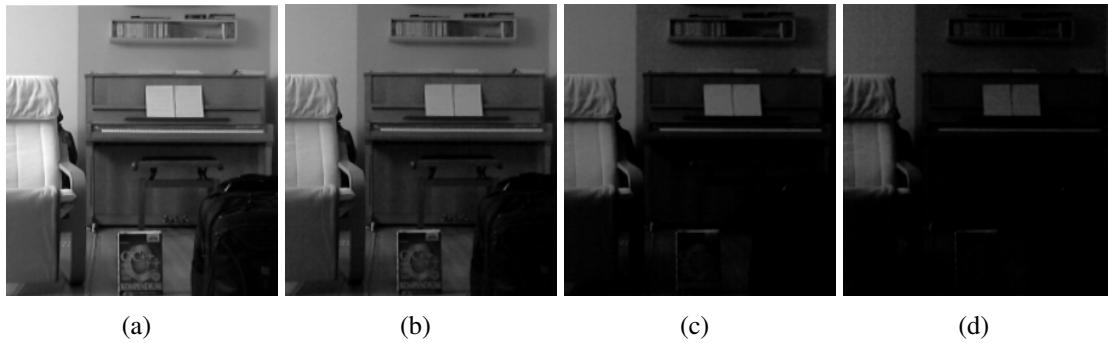


Abbildung A.5.: Messreihe mit SGBM unter verschiedenen Lichtverhältnissen

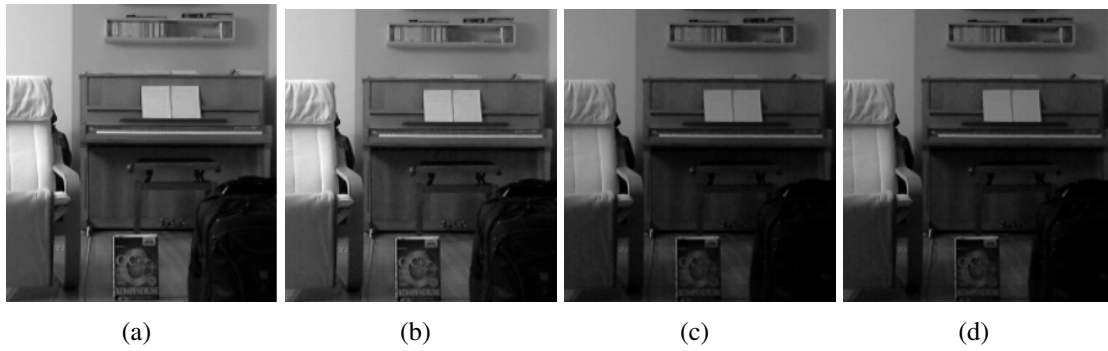


Abbildung A.6.: Messreihe mit BM unter verschiedenen Lichtverhältnissen

Objekt	Sessellehne		Klavier		Klavier		Sessel		Klavier	
	\bar{z}	σ_z	\bar{z}	σ_z	\bar{z}	σ_z	\bar{z}	σ_z	\bar{z}	σ_z
Referenztiefe	298		460		464		296		430	
Abb A.5(a)	289	28	465	7	468	8	297	0	431	0
Abb A.5(b)	286	44	465	7	458	9	297	0	431	0
Abb A.5(c)	304	33	459	7	451	12	297	0	431	1
Abb A.5(d)	302	31	458	8	454	9	297	0	432	1

Objekt	Klavierhocker		Rucksack oben		Decke		Buch		Rucksack	
	\bar{z}	σ_z	\bar{z}	σ_z	\bar{z}	σ_z	\bar{z}	σ_z	\bar{z}	σ_z
Referenztiefe	395		170		226		226		160	
Abb A.5(a)	396	1	169	0	226	0	226	0	158	0
Abb A.5(b)	396	1	169	0	226	0	226	0	159	0
Abb A.5(c)	431	1	431	0	226	1	226	0	176	0
Abb A.5(d)	432	1	431	1	228	2	226	0	424	57

Tabelle A.5.: Entfernungsmessung mit dem SGBM unter verschiedenen Beleuchtungssituationen

Objekt	<i>Sessellehne</i>	<i>Klavier</i>	<i>Klavier</i>	<i>Sessel</i>	<i>Klavier</i>
	\bar{z} σ_z	\bar{z} σ_z	\bar{z} σ_z	\bar{z} σ_z	\bar{z} σ_z
Referenztiefe	298	460	464	296	430
Abb A.6(a)	301 1	453 2	455 3	287 14	435 3
Abb A.6(b)	302 1	452 3	454 4	293 2	436 10
Abb A.6(c)	302 1	452 3	453 6	295 2	439 9
Abb A.6(d)	300 2	454 2	450 8	293 1	439 7

Objekt	<i>Klavierhocker</i>	<i>Rucksack oben</i>	<i>Decke</i>	<i>Buch</i>	<i>Rucksack</i>
	\bar{z} σ_z	\bar{z} σ_z	\bar{z} σ_z	\bar{z} σ_z	\bar{z} σ_z
Referenztiefe	395	170	226	226	160
Abb A.6(a)	403 1	167 1	229 0	226 0	159 0
Abb A.6(b)	403 2	167 1	229 0	226 0	159 0
Abb A.6(c)	400 2	178 48	229 0	226 0	163 3
Abb A.6(d)	405 17	219 404	91 0	226 0	174 1

Tabelle A.6.: Entfernungsmessung mit dem BM unter verschiedenen Beleuchtungssituationen

A.5. Auswirkung von Parametern der Stereo-Algorithmen

Die Messungen dieses Abschnittes erfolgten alle bei einer Auflösung von 320×240 mit einer Basislinie $T = 9.97 \text{ cm}$ und einer Brennweite $f = 451.87 \text{ px}$.

A.5.1. *NumberOfDisparities*

NumberOfDisparities in px	minimal messbare Entfernung in cm	Laufzeit in ms		Bildverlust in %
		BM	SGBM	
16	300	30	70	5
32	145	46	104	10
48	96	56	157	15
64	72	67	176	20
80	57	76	190	25
96	48	84	213	30
112	40	89	225	35
128	35	90	232	40

Tabelle A.7.: Auswirkungen des Parameters *NumberOfDisparities*

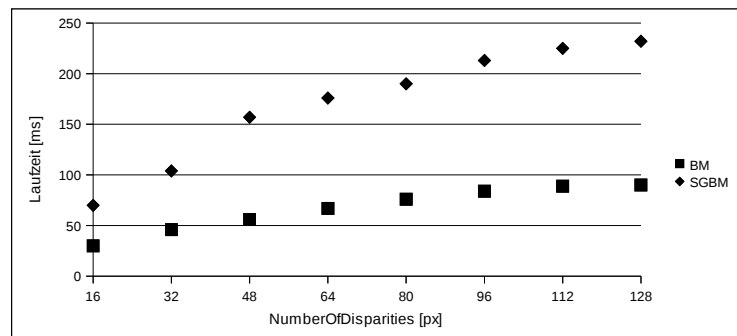


Abbildung A.7.: Laufzeit der Stereo-Algorithmen in Abhängigkeit von *NumberOfDisparities*

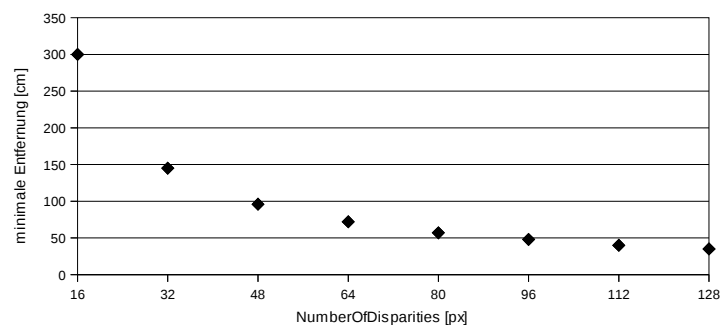


Abbildung A.8.: minimal messbare Entfernung in Abhängigkeit von *NumberOfDisparities*

A.5.2. *UniquenessRatio*



Abbildung A.9.: Linkes Kamerabild zu den folgenden Tiefenbildern

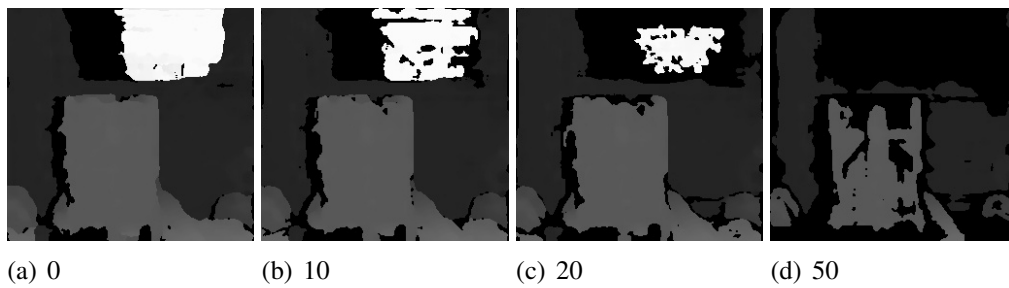


Abbildung A.10.: Auswirkung der *UniquenessRatio* auf die Tiefenkarte beim SGBM

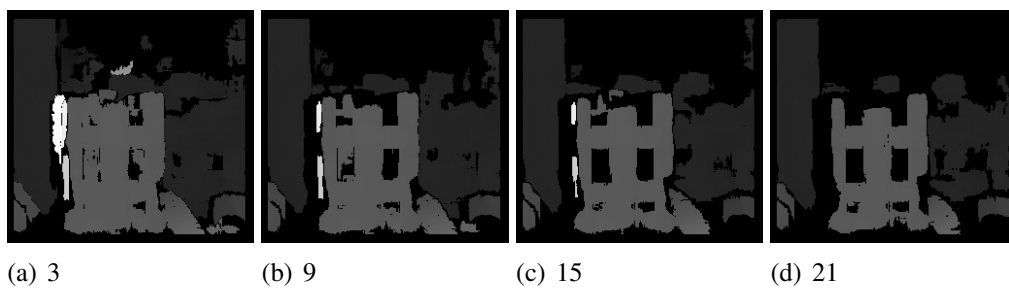


Abbildung A.11.: Auswirkung der *UniquenessRatio* auf die Tiefenkarte beim BM

A.5.3. *SADWindow*

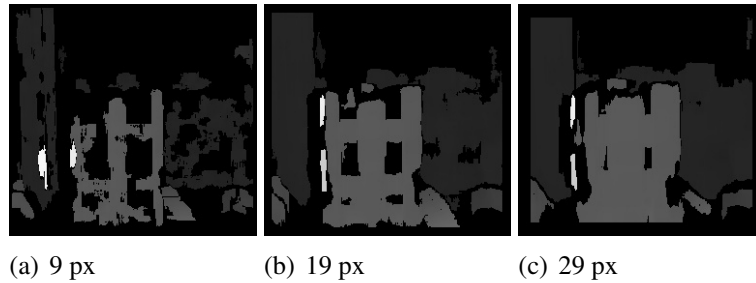


Abbildung A.12.: Auswirkung des *SADWindow* auf die Tiefenkarte beim BM

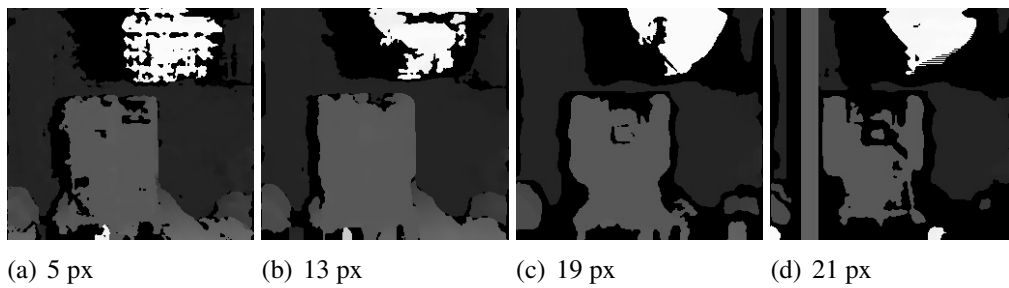


Abbildung A.13.: Auswirkung des *SADWindow* auf die Tiefenkarte beim SGBM

B. Technische Daten

B.1. Kameras

Die Wahl fiel auf zwei Logitech C270 Kameras, da diese bereits auf dem Quadrocopter für andere Zwecke verwendet werden. Zudem wurden auch zwei Logitech C310 Kameras verwendet, die technischen Unterschiede zur C270 sind gering. Einige technische Daten für beide Kameras sind in Tabelle B.1 aufgeführt.

Beide Baureihen sind mit der LightRight-Technologie von Logitech ausgestattet, die ausreichend helle Aufnahmen bei schlechten Lichtverhältnissen ermöglichen soll.

	C270	C310
Brennweite	4 mm	4.4 mm
Fokus	Fix	Fix
Linse	Plastik	Plastik
FOV	60°	60°
Sensor	k.A.	CMOS
optische Auflösung	1280x960	1280x960
Framerate	30fps @ 640x480	30fps @ 640x480
USB	2.0	2.0

Tabelle B.1.: technische Daten der verwendeten Kameras [Logitech, 2010a] [Logitech, 2010b]

Der Öffnungswinkel ist dem Datenblatt entnommen und kann in der Realität stark davon abweichen.

B.2. Notebook

Als Entwicklungsplattform für das Stereo-Vision System diente ein Notebook. Es läuft mit einem AMD Athlon M320 und 2x2.1 Ghz Rechenleistung sowie 4 Gb Arbeitsspeicher.