UNIVERSITÄT KOBLENZ · LANDAU
Fachbereich 4: Informatik

AGAS
Arbeitsgruppe Aktives Sehen

Erstellung einer Tiefenkarte aus Stereobildern für den RoboCup Rescue Wettbewerb

Studienarbeit
im Studiengang Computervisualistik

vorgelegt von

Peter Decker

Betreuer:	Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik
Erstgutachter:	Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik
Zweitgutachter:	Dipl.-Inf. Johannes Pellenz, Institut für Computervisualistik, Fachbereich Informatik

Koblenz, im Januar 2006

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien der Arbeitsgruppe für Studien- und Diplomarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden. ja ☐ nein ☐

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ja ☐ nein ☐

Koblenz, den ..... Unterschrift

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Im Juni 2006 erfolgte im Rahmen der Projektpraktikumsserie Robbie der Universität Koblenz eine Teilnahme am RoboCup 2006 (`http://www.robocup2006.org`) in der Liga Rescue Robots. In dieser Liga wird ein durch Erdbeben oder ähnliches zerstörtes Gebäude nachgestellt. Ziel ist es, mit der Hilfe von Robotern die Räumlichkeiten nach möglichen Überlebenden zu durchsuchen und eine Karte der Umgebung zu erstellen. Diese Karte und die Information über die Lage der Opfer sollen es einem menschlichen Rettungstrupp ermöglichen, im Anschluss in möglichst kurzer Zeit Überlebende zu bergen. Die von den Robotern gelieferten Informationen tragen somit zur Sicherheit der Retter bei, da sie die Einsatzzeit in dem einsturzgefährdeten Gebäude stark reduzieren können.
Zur Erstellung der Karte ist es notwendig, mittels Sensoren Informationen über die Umgebung des Roboters zu sammeln. Ein gängiges Mittel in der Liga sind Laserscanner, welche allerdings noch vergleichsweise teuer sind. Ob ein Stereokamerasystem eine alternative Quelle für die notwendigen 3d-Informationen zur Kartenerstellung in Form einer Tiefenkarte liefern kann, soll in dieser Arbeit untersucht werden.
Die parallel entstandene Arbeit *Schätzung planarer Flächen in verrauschten Tiefenbildern für den RoboCup Rescue Wettbewerb* von Sarah Steinmetz bildet den nächsten Schritt in der Kette und baut auf den Ergebnissen dieser Arbeit auf.

1.2 Umfang der Arbeit

Im Rahmen dieser Arbeit wurden alle zur Tiefenschätzung aus Stereobildpaaren notwendigen Vorverarbeitungsschritte (Stereokamerakalibrierung, Rektifikation) für reale Bildpaare durchgeführt. Es wurden 2 Algorithmen zur Erzeugung von dichten Tiefenkarten aus einem Stereobildpaar, sowohl auf realen wie auch auf synthetischen Bildpaaren angewendet und auf ihre Tauglichkeit in Hinsicht auf die Verwendung zur Kartenerstellung im Rahmen der RoboCup Rescue League untersucht.

1.3 Aufbau der Arbeit

Die Arbeit ist aufgebaut wie folgt: Der Stand der Wissenschaft auf den für diese Arbeit relevanten Teilgebieten wird in Kapitel 2 dargelegt. In Kapitel 3 wird der für die Stereokamerakalibrierung verwendete Algorithmus kurz aufgezeigt. Im Anschluss wird in Kapitel 4 die Notwendigkeit einer Rektifikation der Stereobildpaare dargelegt und die Durchführung kurz erläutert.
In Kapitel 5 wird eine Einführung in die Grundlagen der Tiefen- und Disparitätsschätzung gegeben, und der verwendete Algorithmus genauer beschrieben. Kapitel 6 enthält die Ergebnisse der Experimente, sowie Beschreibungen zum Versuchsaufbau. Ein endgültiges Fazit wird in Kapitel 7 gegeben.

Kapitel 2

Stand der Wissenschaft

Im Folgenden wird eine Übersicht über den Stand der Wissenschaft auf den drei für diese Arbeit relevanten Teilgebieten gegeben werden: Der Kamerakalibrierung, der Rektifikation von Stereobildpaaren und der Disparitätsschätzung aus rektifizierten Stereobildpaaren.

2.1 Kamerakalibrierung

2.1.1 Motivation

Ziel der Kamerakalibrierung ist es, die Eigenschaften der Kamera zu ermitteln. Diese werden intrinische Kameraparameter genannt[Pau03]. Sie hängen sowohl von der Kamera, als auch von dem Objektiv und Einstellungen der Linse ab.

2.1.2 Kalibrierung einer Kamera

Die Kalibrierung von Kameras ist ein gut erforschtes Gebiet. Da wir bei dem gegebenen Szenario der RoboCup Rescue League davon ausgehen können, dass das Kamerasystem vor dem Einsatz zur Kalibrierung zur Verfügung steht, können hier Ansätze, die auf spezielle Kalibriermuster zurückgreifen, in Betracht gezogen werden.

Die Kalibriermethode nach Zhang [Zha00] modelliert sowohl die intrinischen Kameraparameter, als auch die radiale Verzerrung und eine Rotation und Translation relativ zum am Kalibriermuster ausgerichteten Weltkoordinatensystem (extrinsische Kameraparameter).

2.2 Rektifikation eines Stereobildpaares

2.2.1 Motivation

Ziel der Rektifikation (Kapitel 4) eines Stereobildpaares ist die Simulation eines perfekten Stereokamerasystems. Dieses perfekte Stereokamerasystem zeichnet sich dadurch aus, dass die optischen Achsen parallel verlaufen und die Bildebenen koplanar sind. Dies hat zur Folge, dass die Epipole beider Bilder im Unendlichen liegen. Die korrespondieren epipolaren Linien verlaufen dann auf gleicher Höhe, parallel zur x-Achse im Bild. So liegen korrespondierende Punkte stets in der gleichen Zeile im linken und rechten Bild, was den Suchraum für Korrespondenzen erheblich einschränkt.

2.2.2 Methoden

In der Literatur finden sich verschiedene Ansätze um dies zu erreichen. Sind die Projektionsmatrizen der beiden Kameras bekannt, so kann durch einen in [FTV97] beschriebenen Algorithmus ein neues Paar Projektionsmatrizen für das rektifizierte Bildpaar berechnet werden. Die Rektifikation geschieht dann mittels einer linearen Transformation des Bildes. In [PKG99] wird das Bildpaar anhand von Polarkoordinaten parametrisiert. So können die korrespondierenden Epipolaren Linien für das rektifizierte Bildpaar neu abgetastet werden. Die einzige nötige Information für dieses Verfahren ist die F-Matrix.

2.3 Disparitätsschätzung

2.3.1 Motivation

Die Disparität beschreibt die Differenz der x-Koordinaten von Korrespondierenden Pixeln in einem rektifizierten Stereobildpaar. Sie ist aus der Computergrafik auch als horizontale Parallaxe bekannt. Da zwischen dem Tiefenwert eines Punktes und der Disparität ein direkter Zusammenhang besteht (siehe 5.1.1), lässt sich das Problem der Tiefenkarte auf das einer Disparitätskarte reduzieren.

2.3.2 Methoden

Durch die Rektifikation ist es möglich, die Bilder Zeilenweise zu betrachten und einzelne Zeilen zu matchen. Dies kann als ein Optimierungsproblem gesehen werden, bei dem durch das Zuordnen von Pixeln aus dem linken Bild zu korrespondierenden Pixeln aus dem rechten Bild (im Folgenden: Matchen von Pixeln) Kosten entstehen, welche es zu minimieren gilt. Ein gängiges Verfahren zum Lösen derartiger Probleme ist die Dynamische Programmierung.
Die veröffentlichten Algorithmen zur Lösung dieses Problems unterscheiden sich in mehreren Punkten. Zum einen werden unterschiedliche Kostenfunktionen für das Matchen von Pixeln verwendet. Da aufgrund des Stereoaufbaus nicht zwangsweise alle Punkte in beiden Bildern zu sehen sein müssen (siehe Kapitel 5.2.2), müssen zusätzlich zu den Kosten die beim Matchen von Pixeln entstehen noch Kosten für die Verdeckung von Pixeln im linken oder rechten Bild modelliert werden. Der triviale Ansatz ist, konstante Verdeckungskosten anzunehmen.
Birchfield und Tomasi beschreiben einen zweistufigen Algorithmus [BT98], der im Anschluss an das zeilenweise Matching Informationen zwischen den Zeilen propagiert, um die Ergebnisse zu verbessern. Ihre Kostenfunktion belohnt konstante Disparitäten.
Gonzalez legt seinen Fokus auf die Kostenfunktion, die die Verdeckungskosten abhängig von Gradienten modelliert [GCA+99]. Andere Veröffentlichungen benutzen fensterbasierte Kostenfunktionen. Falkenhagen versucht, durch pyramiedenbasierte hierarchische Ver-

fahren diese bestehenden Blockmatchingansätze zu verbessern [Fal97].

Kapitel 3

Kalibrierung

3.1 Motivation

Das generelle Ziel der Kamerakalibrierung ist die Bestimmung der intrinischen Kameraparameter. Da die Kalibrierung in diesem Fall als erster Schritt zur Rektifikation (siehe Kapitel 4) darstellt, muss das Ergebnis dem verwendeten Rektifikationsalgorithmus angemessen sein. Wie in Kapitel 2 bereits erwähnt, verwendet der Algorithmus von Fusiello, Trucco und Verri [FTV97] als Eingabe ein Paar Projektionsmatrizen, welche es also hier zu bestimmen gilt.
Der Kalibrierungsalgorithmus nach Zhang [Zha00] ermöglicht die Bestimmung der radialen Verzerrung und intrinischen Kameraparameter. Auch die extrinsischen Kameraparameter werden, relativ zu einem an dem Kalibriermuster ausgerichteten Weltkoordinatensystem, ermittelt. Die daraus resultierenden Projektionsmatrizen genügen dem verwendeten Rektifikationsalgorithmus als Eingabe.

3.2 Kalibrieralgorithmus nach Zhang

Der folgende Ablauf stammt direkt aus dem Paper von Zhengyou Zhang [Zha00] und soll nur die grobe Vorgehensweise verdeutlichen:

1. Drucken eines Kalibriermusters mit bekannter Geometrie und gut zu detektierenden Merkmalen auf eine ebene Fläche
2. Aufnahme mehrerer Photos des Kalibriermusters unter verschiedenen Winkeln durch Bewegung der Kamera oder des Musters
3. Detektion der Merkmale in den Photos
4. Schätzen der 5 intrinsischen und aller extrinsischen Parameter
5. Schätzen der radialen Verzerrung
6. Verfeinern aller Parameter durch Minimierung des Rückprojektionsfehlers

3.3 Kalibrierung eines Stereokamerasystems

Durch die Informationen über Rotation und Translation aus dem durch das Kalibriermuster vorgegebenen Weltkoordinatensystems in die zwei Kamerakoordinatensysteme, welche in den Projektionsmatrizen enthalten ist, lässt sich leicht die Rotation und Translation zwischen den Kameras berechnen. Der Abstand der Kameras entspricht dem Betrag des Translationsvektors $\boldsymbol{t}$ und wird auch als Stereobasis b bezeichnet.

$$b = \|\boldsymbol{t}\| \tag{3.1}$$

Kapitel 4

Rektifikation

4.1 Motivation

Ziel der Rektifikation ist eine Neuabtastung des Stereobildpaares aus der Position eines perfekt ausgerichteten Stereokamerasystem. Dadurch wird erreicht, dass die epipolaren Linien parallel und horizontal verlaufen. Durch diesen Schritt in der Vorverarbeitung wird das 2-D Problem des Findens von Stereokorrespondenzen auf 1-D reduziert, da korrespondierende Punktepaare nur noch auf den entsprechenden horizontalen Linien im rektifizierten Bildpaar gesucht werden müssen.[FTV97]

4.2 Rektifikation einer Stereogeometrie nach Fusiello, Trucco und Verri

Die Idee des Algorithmus ist es, die Projektionsflächen der beiden Kameras so um die Kamerazentren zu drehen, dass ein perfektes Stereosystem entsteht. Die optischen Zentren der Kameras bleiben dabei dieselben.
Als Eingabe genügen die beiden Projektionsmatrizen der linken und rechten Kamera. Der Translationsanteil sowie die intrinischen Kameraparameter bleiben erhalten, lediglich die Rotation der beiden Kameras wird verändert. So ergibt sich ein neues Paar Projektionsma-

trizen. Zur Durchführung der Rektifikation muss nun lediglich die Transformationsmatrix berechnet werden, die die Bildebene der ursprünglichen Projektionsmatrix auf die neue projiziert.

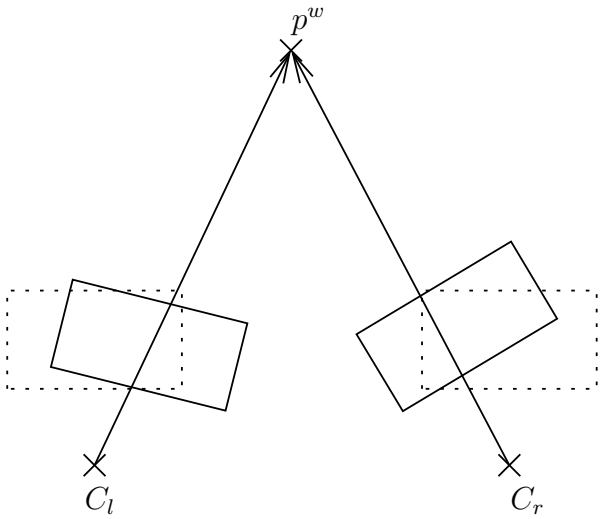

Bild 4.1: Rektifikation nach [FTV97]

In Bild 4.1 sieht man die Bildebenen der linken (C_l) und rechten Kamera (C_r) als durchgezogene Rechtecke. Die gestrichelten Rechtecke zeigen die simulierte Position der Bildebenen, die aus der Rektifikation resultieren.

Kapitel 5

Tiefenschätzung

5.1 Motivation

Um 3-D Informationen für die Erstellung einer Karte aus Stereobildpaaren zu extrahieren, sollen im Folgenden dichte Tiefenkarten erstellt werden. Wie schon in Kapitel 2 erwähnt, ist dieses Problem äquivalent zu dem einer dichten Disparitätskarte. Dichte Disparitätskarten (im Englischen: dense disparity maps) enthalten, im Gegensatz zu sparse disparity maps, Disparitätsinformationen zu jedem Pixel, und nicht nur vereinzelte für gut detektierbare Merkmale.

5.1.1 Zusammenhang zwischen Disparität und Tiefe

Gegeben seien zwei korrespondierende Punkte p im linken und q im rechten Bild eines rektifizierten Stereobildpaares. Die Disparität d ist definiert als

$$d = |p_x - q_x| \tag{5.1}$$

Durch die Kalibrierung (vgl. Kapitel 3) ist die Stereobasis b, sowie die Brennweite F bekannt. Bei gegebener Disparität d zweier korrespondierender Pixel ergibt sich für den

entsprechenden Weltpunkt p^w die Tiefe p_z^w:

$$p_z^w = \frac{F}{d} \cdot b \tag{5.2}$$

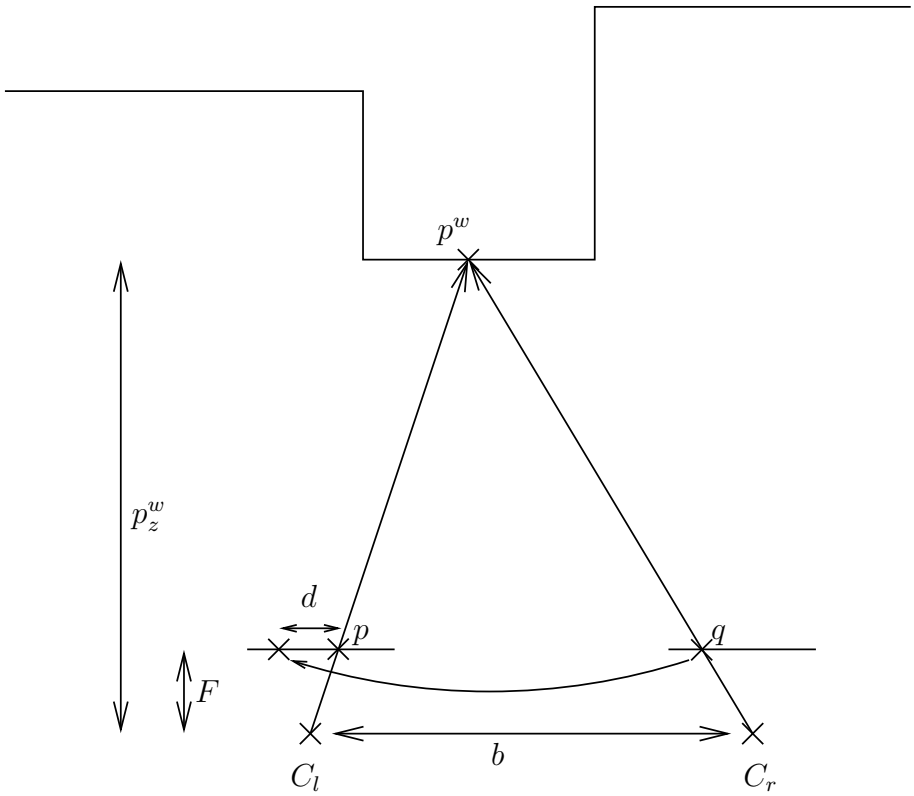

Bild 5.1: Zusammenhang zwischen Disparität und Tiefe

Das Hauptziel der Tiefenschätzung lässt sich also auf die Korrespondenzsuche zwecks Disparitätsermittlung zurückführen. Wie schon in Kapitel 4 ausgeführt, lässt sich der Suchraum für korrespondierende Punkte dank der Rektifikation auf eine Zeile reduzieren.

5.2 Scanlinematching mit dynamischer Programmierung

Gegeben sind im Folgenden 2 Scanlines (Bildzeilen) der Länge n, die es zu matchen gilt, d.h. es soll für jedes Pixel in der linken Scanline das korrespondierende in der rechten gefunden werden. Dieses Problem wird mittels Dynamischer Programmierung gelöst. Die Idee bei der Dynamischen Programmierung ist es, ein komplexes Problem (das Matchen zweier Scanlines) in viele kleine Probleme zu zerlegen (hier: das Matchen von einzelnen Pixeln). Anschließend löst man alle möglichen Teilprobleme (hier: mögliche Pixelmatches), und speichert die Ergebnisse. Diese Methode hat gegenüber rekursiven Ansätzen den Vorteil, dass Teilergebnisse nicht unabhängig voneinander mehrmals berechnet werden.[Sed91]
Angewandt auf das gegebene Problem heißt das:
Beim Matchen von Pixeln entstehen Kosten. Seien $c_{\text{match}}(i,j)$ die Kosten, die für das Matchen von p_i auf q_j entstehen. Diese berechnen sich beispielsweise durch den quadratische Abstand der Grauwerte der Pixel. Die Kosten, die beim Matchen der kompletten Scanline entstehen seien $C(n,n)$. Gesucht ist sowohl die Zuordnungsvariante, die die minimalen Gesamtkosten verursacht, als auch die dafür nötigen Teillösungen, da sie die Information über die Disparitäten halten. Alle möglichen Teillösungen des Problems lassen sich in einer Kostenmatrix festhalten: Man notiere alle möglichen Kosten in der Matrix $\boldsymbol{M} \in \mathbb{R}^{(n,n)}$, wobei die Kosten an der Stelle $\boldsymbol{M}_{i,j}$ die minimalen Kosten für das Matchen der ersten i Pixel auf die ersten j Pixel darstellen.

Ein einzelnes Element der Matrix berechnet sich dann als Summe aus den Matchkosten für das Pixelpaar und den geringsten Vorgängerkosten:

$$\boldsymbol{M}_{i,j} = c_{\text{match}}(i,j) + \min(\boldsymbol{M}_{i-1,j}, \boldsymbol{M}_{i,j-1}, \boldsymbol{M}_{i-1,j-1})) \tag{5.3}$$

Zur Bestimmung der Disparität folgt man dem optimalen Weg der geringsten Kosten rückwärts durch die Matrix, indem man bei $M(n,n)$ startet und entweder einen Schritt nach links, nach oben oder nach links oben geht, je nachdem wo die geringsten Kosten stehen. So erhält man Disparitäten für jedes Pixel, die nach jedem Schritt nach oben oder links oben (d.h. wenn man den aktuell betrachteten Pixel der linken Scanline ändert) abzulesen

ist:

$$d = |i - j| \tag{5.4}$$

5.2.1 Weitere Einschränkung des Suchraums

Aufgrund der Stereogeometrie kann die Anzahl der zu berechnenden Elemente der Kostenmatrix eingeschränkt werden. Folgende Bedingung kann dafür angenommen werden:

$$i \geq j \tag{5.5}$$

Ein Weltpunkt kann auf die linke Scanline nicht weiter links abgebildet werden als auf die rechte. Gleichheit wird für Punkte im Unendlichen zugelassen.

$$i - j \leq d_{\text{max}} \tag{5.6}$$

Eine maximale Disparität d_{max} schränkt den Suchraum weiter ein. Anders als 5.5 stellt 5.6 eine Optimierung dar, die nicht mit physikalischen Tatsachen begründbar ist. Sie modelliert die Annahme, dass alle Objekte in der Szene einen minimalen Abstand zur Kamera nicht unterschreiten.

Bild 5.2 zeigt die reduzierte Anzahl zu berechnender Elemente der Kostenmatrix.

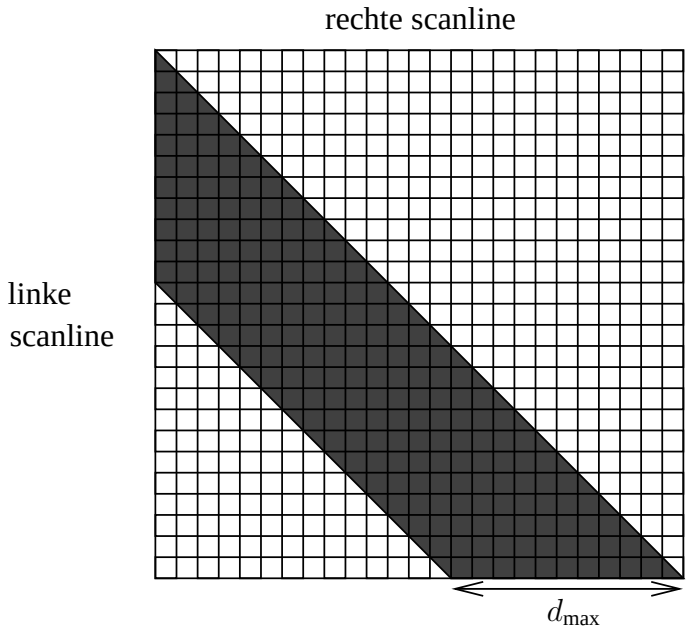

Bild 5.2: Reduktion der zu Berechnenden Elemente der Kostenmatrix

5.2.2 Verdeckung

Gleichung 5.3 impliziert, dass jedes Pixel in der linken Scanline einen Korrespondieren Pixel in der rechten Scanline hat. Dies ist jedoch aufgrund von zwei Faktoren nicht der Fall:

1. Durch das Vorhandensein einer Stereobasis zeigen die beiden Bilder in den meisten Fällen nicht exakt den gleichen Ausschnitt der Welt, sondern sind entlang der x-Achse verschoben.

2. Es können auch in der Mitte der Bilder verdeckte Pixel existieren, die nur in einem der beiden Kamerabilder zu sehen sind (vgl. Bild 5.3).

Um Punkt 1 genüge zu tun, sollte beim Aufbau der Matrix eine beliebige initiale Disparität zugelassen werden:

$$\boldsymbol{M}_{i,1} = c_{\text{match}}(i, 1) \tag{5.7}$$

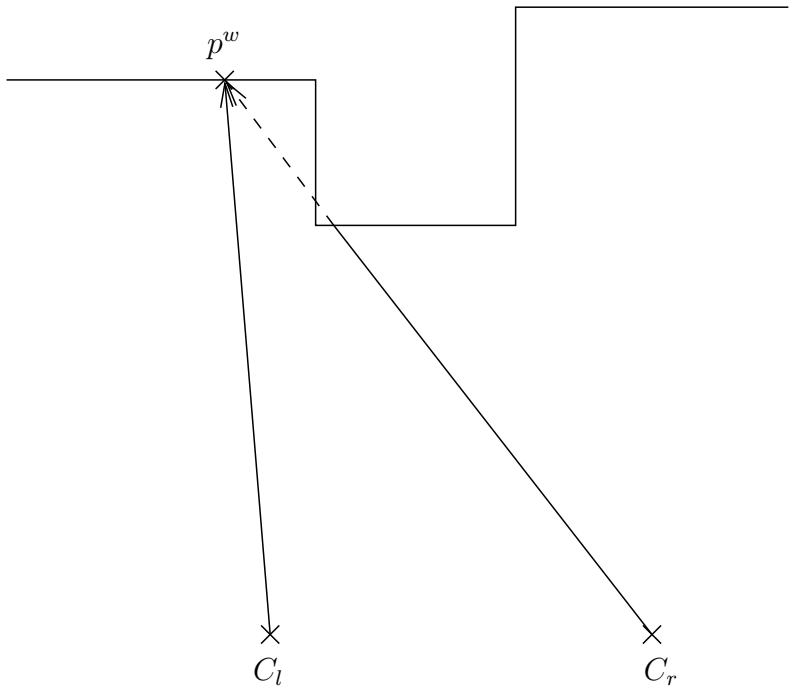

Bild 5.3: Verdeckung in einem Stereobild: der Punkt p^w ist nur im Bild der linken Kamera C_l sichtbar

Weiterhin sollte als Ausgangspunkt des Zurückverfolgens der geringsten Werte das Minimum, bei dem die gesamte linke Scanline auf einen Teil der rechten gematcht wird, gewählt werden. Man beginnt also bei $\min(\boldsymbol{M}(n,1), \boldsymbol{M}(n,2), ...\boldsymbol{M}(n,n))$.
Auch die Verdeckung muss modelliert werden, um Fehler, die sich durch das zwangsweise Matchen von Pixeln, die in der anderen Scanline nicht vorkommen, zu vermeiden. Verdeckungen sind äquivalent zu Änderungen in der Disparität. Bild 5.4 zeigt den Aufbau der Kostenmatrix unter Miteinbeziehung von Verdeckungen im Bild.

Es existieren wieder drei mögliche Werte, mit denen $M(i,j)$ gefüllt werden kann. Allerdings sind diese nicht mehr alle von c_{match} und den niedrigsten Kosten der möglichen Vorgänger abhängig wie in 5.3. Dies kann zur Folge haben, dass der korrekte Weg nicht durch den günstigsten Vorgängerknoten der Matrix führt. Um die Rückverfolgung weiterhin zu ermöglichen, muss also zusätzlich zu den Kosten noch der gewählt Vorgänger in

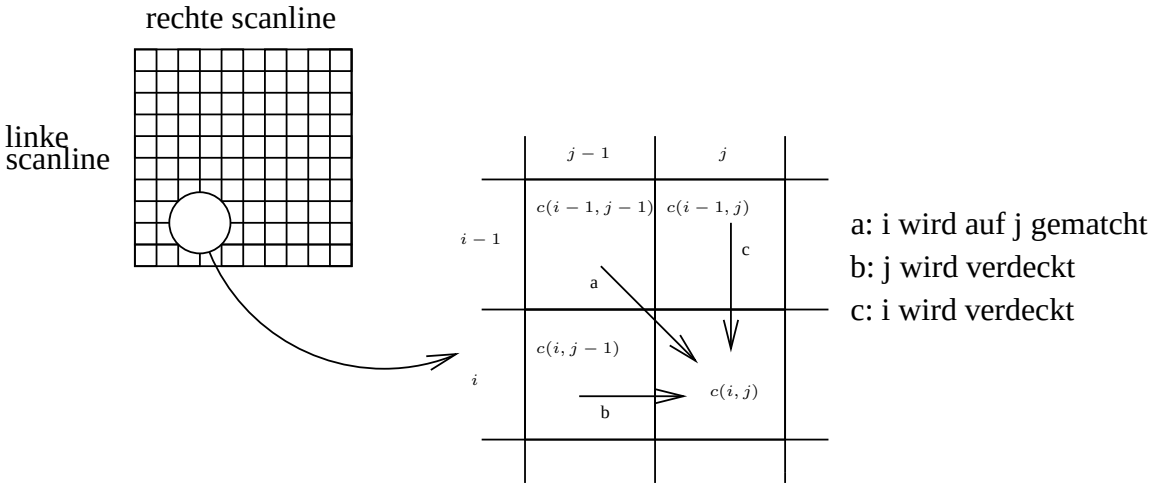

Bild 5.4: Aufbau der Kostenmatrix (vlg.[GCA+99])

der Matrix abgelegt werden.

Der Triviale Ansatz, um das mathematische Modell anzupassen ist, konstante Verdeckungskosten c_{occ} für das ganze Bild anzunehmen. Gleichung 5.3 wird dadurch erweitert zu:

$$\boldsymbol{M}_{i,j} = \min(c_{\text{match}}(i,j) + \boldsymbol{M}_{i-1,j-1}, c_{\text{occ}} + \boldsymbol{M}_{i-1,j}, c_{\text{occ}} + \boldsymbol{M}_{i,j-1}) \tag{5.8}$$

Der komplette Algorithmus sieht dann folgendermaßen aus:

<table>
<tr><td colspan="3">Aufbau der Kostenmatrix nach 5.8, unter Beachtung von 5.7, 5.5 und 5.6</td></tr>
<tr><td colspan="3">Suche das Minimum zum Start des Backtrackings:
$S(i,j) \in \boldsymbol{M} = \min(\boldsymbol{M}(n,n), \boldsymbol{M}(n,n-1), ...\boldsymbol{M}(n,n-d_{\text{max}})$</td></tr>
<tr><td colspan="3">Sei $S(i,j)$ im Folgenden das aktuelle Element der Matrix zum Zeitpunkt t.</td></tr>
<tr><td colspan="3">WHILE (i > 0)</td></tr>
<tr><td></td><td colspan="2">Gehe zum in der Matrix gespeicherten Vorgänger von $S(i,j)$</td></tr>
<tr><td></td><td>IF</td><td>$i_t < i_{t-1}$</td></tr>
<tr><td></td><td>THEN</td><td>Trage die Disparität $d = |i-j|$ an die Stelle i in der Disparitätskarte in der aktuellen Zeile ein.</td></tr>
</table>

5.3 Alternative Kostenfunktion

Wie schon in Kapitel 2 erwähnt, bietet der in Kapitel 5.2 vorgestellte Grundalgorithmus viele Möglichkeiten der Verbesserung. Im Folgenden soll eine alternative Kostenfunktion nach [GCA+99] vorgestellt werden, mit deren Hilfe die Verdeckungskosten abhängig vom Bildinhalt geschickter modelliert werden können.

5.3.1 Motivation

Aufgrund von Fehlern die durch Quantisierung entstehen, kann es zu fehlerhaften Matches kommen, wenn nur die Pixelintensität betrachtet wird.

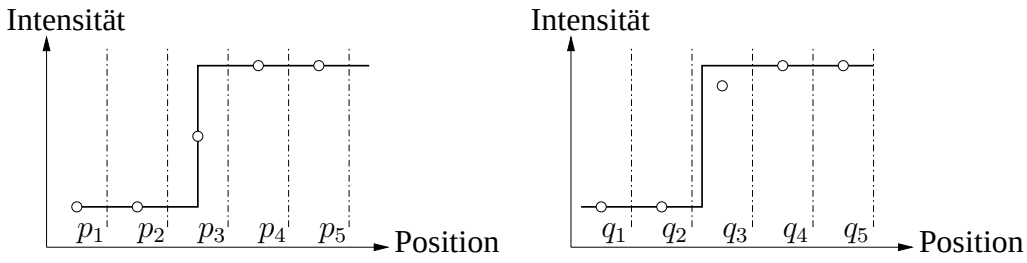

Bild 5.5: Fehlzuordnung durch Quantisierung

Bild 5.5 zeigt eine durch Quantisierung entstandene Fehlzuordnung: Das korrekte Match für p_3 wäre q_3. Durch die Quantisierung ist jedoch $c_{\text{match}}(p_3, q_3) > c_{\text{match}}(p_4, q_3)$. Durch Einbeziehung der Gradienten kann dieser Effekt umgangen werden.

5.3.2 Evidence

Ausschlaggebend für die Kosten einer Verdeckung sollen similarity und confidence sein. Similarity beschreibt die Ähnlichkeit zweier Pixel. Eine hohe Ähnlichkeit deutet auf einen wahrscheinlichen Match hin, die Verdeckungskosten müssen also entsprechend hoch sein. Confidence beschreibt die Zuverlässigkeit dieser Information.

Da Verdeckungen hauptsächlich an Objektgrenzen vorkommen, wird die Funktion auf Gradienten basieren. Evidence dient als Maß für die Wahrscheinlichkeit einer Verdeckung.

$$E(\boldsymbol{g_i}, \boldsymbol{g_j}) = \frac{|\boldsymbol{g_i}| + |\boldsymbol{g_j}|}{2} - |\boldsymbol{g_i} - \boldsymbol{g_j}| \tag{5.9}$$

wobei $\boldsymbol{g_i}$ und $\boldsymbol{g_j}$ die Gradienten an den entsprechenden Stellen im linken bzw. rechten Bild sind.
Der erste Summand in 5.9 modelliert die confidence; je stärker die Gradienten sind, desto sicherer ist die Information die sie tragen. Der zweite Summand modelliert die similarity bzw. den Unterschied der Gradienten; je ähnlicher sich die Gradienten sind, desto geringer ist der Unterschied. Ein stark positiver Wert der Evidence deutet also auf eine hohe Ähnlichkeit mit hoher Sicherheit hin, ein stark negativer Wert auf einen hohen Unterschied mit hoher Sicherheit. Werte nahe Null bedeuten, dass die Information aus den Gradienten nicht sehr zuverlässig ist.

5.3.3 Modified Evidence

Modified Evidence bildet den Wertebereich der Evidence invers auf $[0..1]$ ab.

$$ME(\boldsymbol{g_i}, \boldsymbol{g_j}) = \frac{255 - E(\boldsymbol{g_i}, \boldsymbol{g_j})}{510} = \frac{255 - \frac{|\boldsymbol{g_i}| + |\boldsymbol{g_j}|}{2} + |\boldsymbol{g_i} - \boldsymbol{g_j}|}{510} \tag{5.10}$$

So entsteht ein leicht verwendbares, auf Gradienten basierendes Maß für die Wahrscheinlichkeit einer Verdeckung:

- $ME = 0$: Gleiche Gradienten mit maximaler Länge - Wahrscheinlichkeit einer Verdeckung minimal
- $ME = 0.5$: Untexturierter Bereich, schwache Gradienten - Verdeckung abhängig von Pixelwerten machen
- $ME = 1$: Entgegengesetzte Gradienten mit maximaler Länge - Höchst wahrscheinlich kein Match sondern eine Verdeckung

5.3.4 Kostenfunktion für Verdeckungen nach Gonzalez

Mit dem in Kapitel 5.3.3 eingeführten Maß der Modified Evidence lässt sich eine Kostenfunktion modellieren, die abhängig vom Informationsgehalt der Gradienten die Kosten einer Verdeckung modelliert. Die Verdeckungskosten sollen hierbei maximal werden, wenn die Wahrscheinlichkeit einer Verdeckung minimal ist; eine Verdeckung mit hoher Wahrscheinlichkeit soll zu geringen Verdeckungskosten führen. Gleichung 5.11 genügt diesen Bedingungen:

$$c_{\text{occ}}(\boldsymbol{g_i}, \boldsymbol{g_j}) = K_1 \cdot \left(1 + K_2 \cdot e^{-\frac{ME(\boldsymbol{g_i}, \boldsymbol{g_j})}{K_3}} \right) \tag{5.11}$$

Durch die 3 Parameter K_1, K_2 und K_3 lässt sich die Funktion anpassen:

- K_1: Die minimalen Verdeckungskosten. Da in [GCA+99] die quadrierte Grauwertdifferenz als c_{match} angenommen wird, sollte dieser Wert zwischen 37 und 401 gewählt werden.
- K_2: Ein genereller Faktor, um die Verdeckungskosten zu erhöhen. Als Wertebereich wird in dem Paper $[1..30]$ empfohlen
- K_3: Bestimmt, wie schnell sich die Verdeckungskosten ändern. Empfohlener Wertebereich: $[0.01..0.5]$

Grafiken 5.6 - 5.8 verdeutlichen den Einfluss, den die Parameter auf die Verdeckungskostenfunktion haben. Die Referenzfunktion wurde mit $K_1 = 145$, $K_2 = 10$ und $K_3 = 0.1$ parametrisiert.

Sehr hohe Verdeckungskosten führen im Allgemeinen zu starken Weichzeichnungseffekten innerhalb der Bildzeilen, weil weniger Verdeckungen zu einer geringeren Anzahl an Sprüngen in der Disparität führen. Bei sehr geringen Verdeckungskosten werden hingegen auch ähnliche Pixel zu oft fälschlicherweise Verdeckt, was zu einer hohen Anzahl an Sprüngen in der Disparität und Rauscheffekten führt.

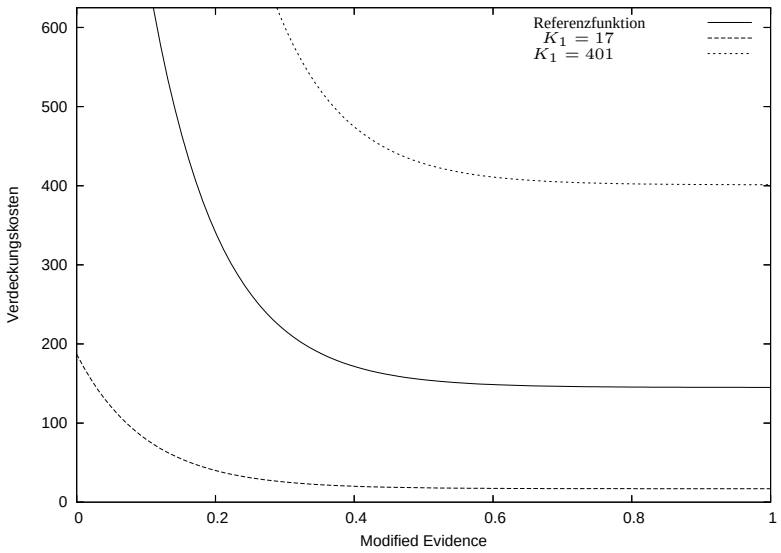

Bild 5.6: Auswirkung des 1. Parameters K_1: Anpassung der minimalen Verdeckungskosten

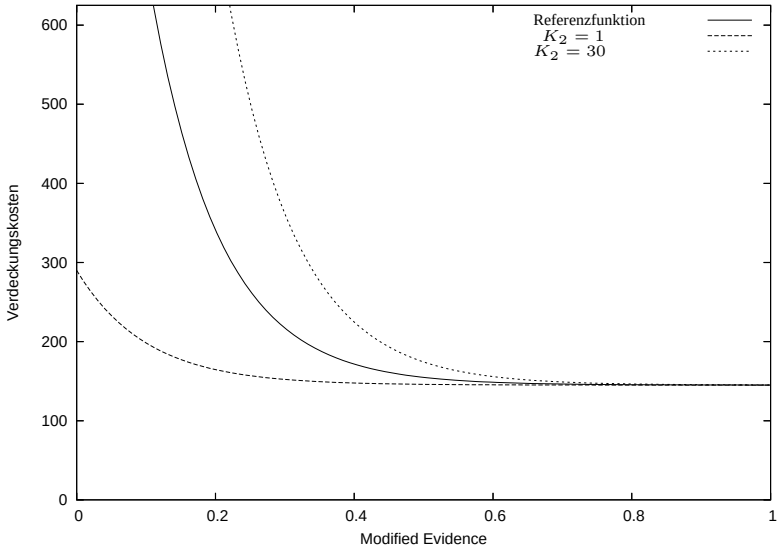

Bild 5.7: Auswirkung des 2. Parameters K_2: Genereller Faktor zur Erhöhung der Verdeckungskosten

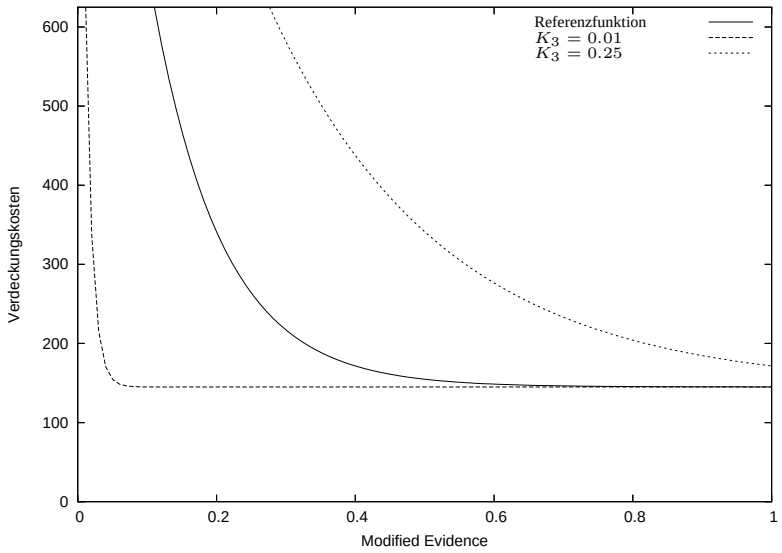

Bild 5.8: Auswirkung des 3. Parameters K_3: Beeinflusst, wie schnell sich die Verdeckungskosten bei steigender Modified Evidence an die Minimalen Verdeckungskosten annähern

5.3.5 Postprocessing

Im Gegensatz zu dem Algorithmus nach Birchfield, ist in dem Paper von Gonzalez nicht vorgesehen, Informationen scanlineübergreifend zu verwenden. Um die Auswirkungen von durch Rauschen oder anderen Fehlerquellen bedingten Fehlzuordnungen kompletter Scanlines zu minimieren und geschlossene Flächen für die Weiterverarbeitung zu garantieren empfiehlt es sich, einen Postprocessing Schritt anzuschließen. Ein Glättungsfilter der auf das Disparitätsbild angewendet wird, soll entstandene Fehler beseitigen. Der Medianfilter scheint hier aus mehreren Gründen geeignet. Er besitzt die erforderlichen Glättungseigenschaften, selbst komplett falsche Zeilen werden, wenn sie von innerhalb eines Objekts liegen, korrigiert. Außerdem entstehen, im Gegensatz zu anderen Glättungsfiltern, wie zum Beispiel dem Gauss-Filter, keine neuen Werte.

Bilder 5.9 und 5.10 zeigen die rauschunterdrückende Wirkung des Medianfilters bei den Standartstereobildpaaren Sawtooth und Tsubaka.
Die Stereobildpaare stammen von `www.middlebury.edu/stereo`.

Bild 5.9: Von links nach rechts: Das ursprüngliche Disparitätsbild; mit Medianfilter der Größe 3x3 geglättet; mit Medianfilter der Größe 5x5 geglättet

Bild 5.10: Von links nach rechts: Das ursprüngliche Disparitätsbild; mit Medianfilter der Größe 3x3 geglättet; mit Medianfilter der Größe 5x5 geglättet

Kapitel 6

Experimente und Ergebnisse

6.1 Versuchsaufbau

Für eine erste Versuchsreihe wurden 2 Kameras vom Typ Imaging Source DFx 31BF03 annähernd parallel ausgerichtet und fixiert. An den entstandenen Aufnahmen wird im Folgenden beispielhaft der Prozess des Erstellens einer Disparitätskarte demonstriert.

6.2 Kamerakalibrierung

Als Kalibriermuster wurde ein Schachbrett mit 12 x 12 Feldern der Größe 5cm x 5cm verwendet. Das Kalibriermuster ist auf einem Stück Holz angebracht um planarität zu gewährleisten (Es entstand bereits im Rahmen des Projektpraktikums “Robbie 5: Roboter im Outdoorbereich“).

Bild 6.1 und 6.2 zeigen zwei von drei Stereobildpaaren, die zur Bestimmung der intrinsischen Kameraparameter verwendet wurden. Aus 6.1 wurden die extrinsischen Kameraparameter extrahiert und damit auch die Projektionsmatrizen, welche für die Rektifikation verwendet werden.

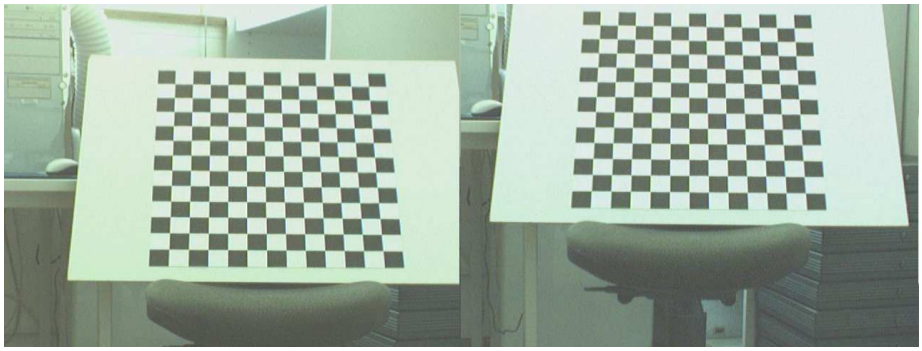

Bild 6.1: Stereobildpaar, aus welchem eine Disparitätskarte erstellt werden soll

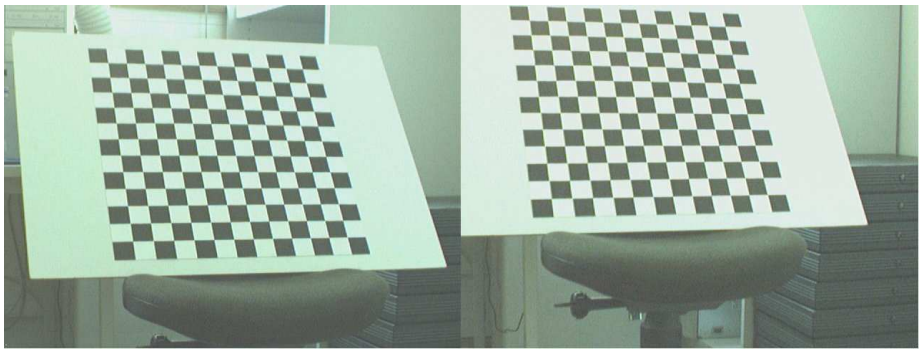

Bild 6.2: Ein weiteres Stereobildpaar, welches das Kalibriermuster aus einer anderen Perspektive zeigt

6.3 Rektifikation

Mit Hilfe der durch die Kalibrierung gewonnener Information konnte das Stereobildpaar 6.1 rektifiziert werden:

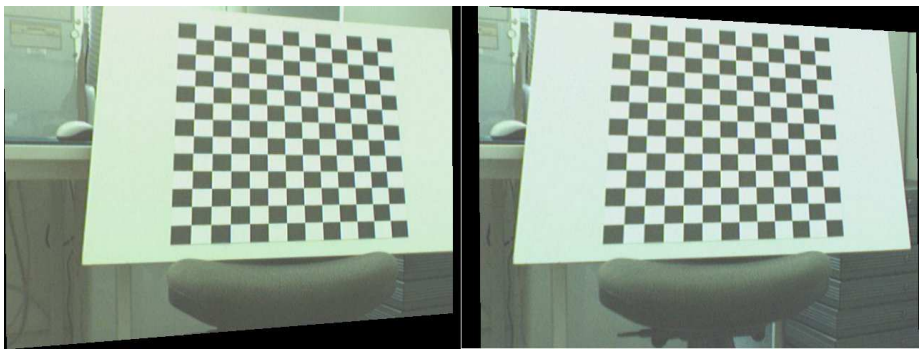

Bild 6.3: Ergebnis der Rektifikation von Bild 6.1

6.4 Disparitätskarten

Auf die entstandenen Disparitätskarten wurde ein Histogramm-Stretching angewendet, um die Visualisierung zu verbessern. Hohe Grauwerte deuten auf eine hohe Disparität, also eine geringere Tiefe hin.

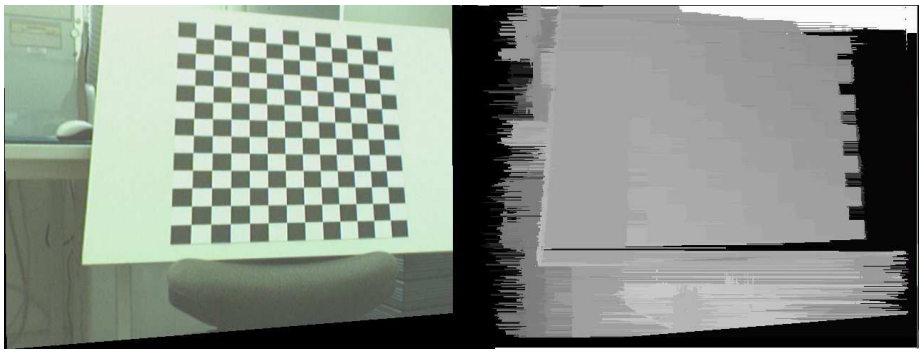

Bild 6.4: Ergebnis des Algorithmus nach Birchfield

Die Transformation der Rektifikation kann rückgängig gemacht werden, um die Disparitätskarte für das ursprüngliche linke Kamerabild zu erhalten.

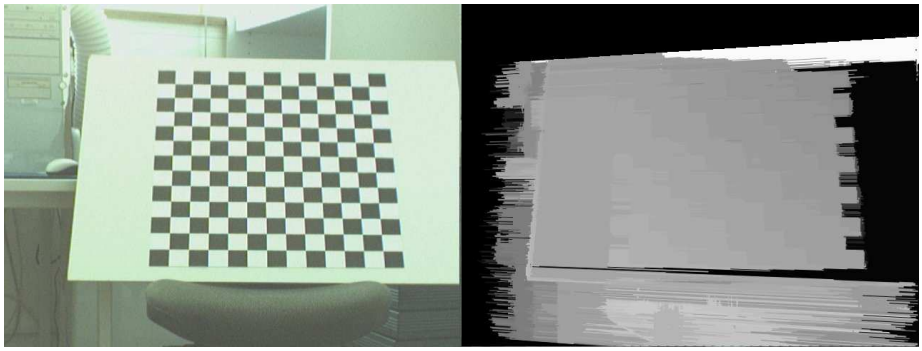

Bild 6.5: Rückrechnung der Rektifikation von 6.4

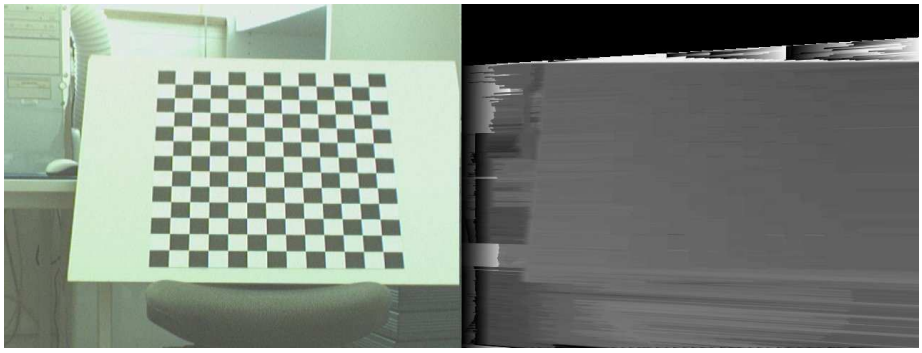

Bild 6.6: Ergebnis des Algorithmus nach Gonzalez

6.5 Ergebnisse auf synthetischen Bildern

6.5.1 USARSim

Im Rahmen der RoboCup Rescue League entstand das Projekt USARSim (`http://sourceforge.net/projects/usarsim`), ein auf dem kommerziellen Computerspiel Unreal Tournament aufsetzender Mod der es erlaubt, ähnliche Umgebungen wie in der Rescue League zu simulieren und mit virtuellen Robotern zu durchfahren. Diese Umgebung eignet sich vor allem gut zum Testen von Autonomiealgorithmen, ohne dass aufwendige Arbeiten zum Schaffen der Testumgebung notwendig sind. Simuliert werde kön-

nen verschiedene Sensoren, wie Laserscanner und Sonar. Das Auslesen von simulierten Kamerabildern ist im momentanen Stadium noch nicht ohne weiteres möglich. Im Hinblick auf die weitere Entwicklung dieser Software scheint es aber zumindest möglich, dass dieses Feature Einzug in zukünftige Versionen hält.
Da die Bilder aber angezeigt werden können, konnten sie per Screenshot zum Testen der Algorithmen gewonnen werden (Vielen Dank an Stefan Burghardt und Dennis Holzhaeuser).

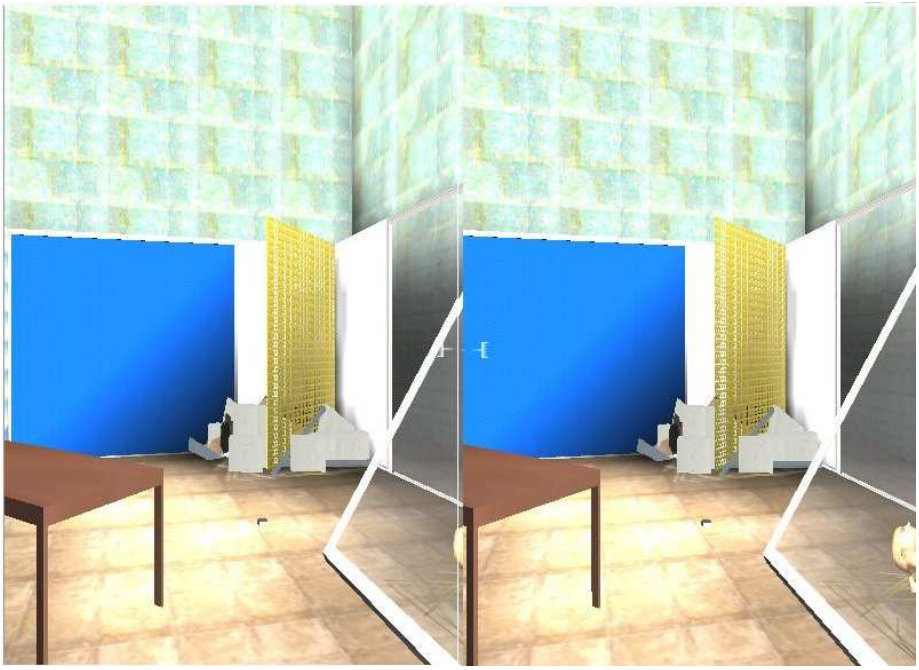

Bild 6.7: Stereobildpaar aus der USARSim

Deutlich zu sehen ist hier ein Problem, dass durch bildbasierte Tiefenschätzung nicht zu lösen ist: Das Glas im rechten Teil von Bild 6.7 wird ignoriert (Bild 6.8). Ein anderes Problem sind Highlights auf nicht diffusen Flächen. Da sich die Highlights, die in den verschiedenen Kamerabildern zu sehen sind nicht an der gleichen Stelle in der Welt befinden, muss die Disparitätsschätzung auch in diesem Fall versagen.

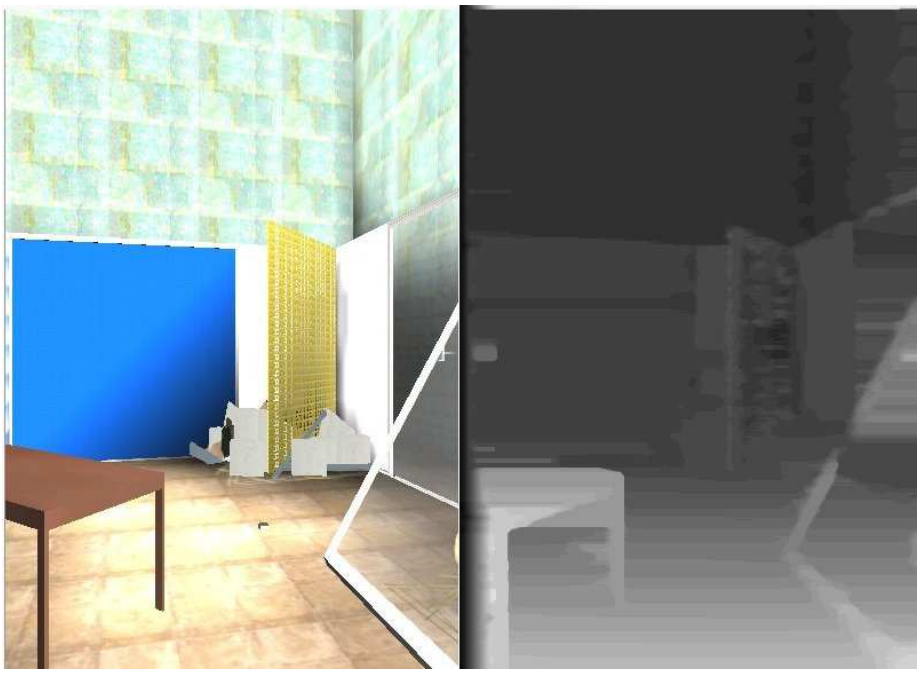

Bild 6.8: Tiefenkarten aus dem Stereobildpaar 6.7

Bild 6.9: Ein weiteres Stereobildpaar aus der USARSim

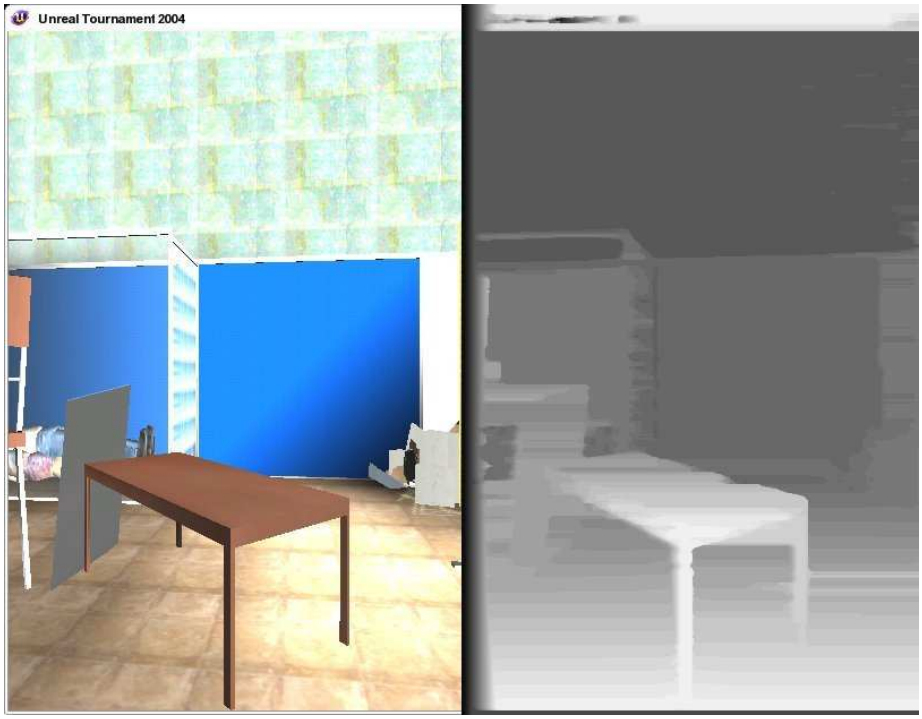

Bild 6.10: Tiefenkarten aus dem Stereobildpaar 6.9

Kapitel 7

Fazit und Ausblick

Die Stärken eines Stereokamerasystems als Sensoreinheit zur 3D-Kartenerstellung liegen eindeutig in den geringen Anschaffungskosten. In vielen Fällen ist schon mindestens eine Kamera vorhanden, deren Bilder vom Operator direkt zur Navigation genutzt werden. Für ein Stereosystem ist es allerdings unerlässlich, dass zwei Kameras vom gleichen Typ verwendet werden um möglichst gleiche Bilder zu erhalten. Das Stereosystem von Robbie besteht zum jetzigen Zeitpunkt aus zwei unterschiedlichen Kameras (Sony DFW-X710 und Sony DFW-X700). Die Signale dieser Kameras unterscheiden sich so stak, dass eine Disparitätskartenerstellung sehr schwierig ist. Eine Mögliche Lösung für das Problem besteht in einer Farbnormierung der Kamerabilder.
Ein Nachteil des Stereokamerasystems gegenüber Laserscannern ist die stark begrenzte Tiefenauflösung. Da keiner der vorgestellten Ansätze ein subpixelgenaues Matchen der Scanlines in Betracht zieht, ist die Anzahl der sich ergebenden Tiefenwerte begrenzt durch die maximal vorkommende Disparität. Hierdurch können starke, aliasingähnliche Effekt bei Flächen entstehen, die in die Tiefe des Raums laufen und nicht orthogonal zur Kamerahauptachse liegen. Ein weiteres Problem sind Mehrdeutigkeiten die sich durch schwachtexturierte Flächen, vor allem an den Bildrändern ergeben. Ebenso problematisch sind nicht diffuse Objekte, sowie Transparenzen.
Zieht man die erwähnten Schwächen in Betracht, so halte ich ein Stereokamerasystem einem System das auf einem schwenkbaren Laserscanner aufbaut in Hinsicht auf eine

automatische Erstellung für eine Karte im Rahmen der RoboCup Rescue League für klar unterlegen. Eventuell ist die Extraktion komplexerer geometrischer Objekte, wie zum Beispiel Ebenen aus dichten Tiefenkarten eine Möglichkeit, stabilere Daten als die stark verrauschten und fehlerbehafteten einzelnen Tiefenwerte für Pixel zur Kartenerstellung zu gewinnen.

Anhang A

Implementationsdetails

A.1 Verwendete Bibliotheken

Als Bildkontainerklasse dient `IplImage` aus der **OpenCV**-Bibliothek (Version 0.9.6). Außerdem basieren die Rektifikation, Disparitätsschätzung nach Birchfield und die Kalibrierung auf **OpenCV**. Letztere wurde schon Rahmen des “Robbie5 - Roboter im Outdoorbereich“-Projektpraktikums implementiert und hier lediglich erweitert.
Für die Oberfläche (A.2.4) wurden **Qt3** und **OpenGL** mit **GLUT** eingesetzt. Die Kompatibilität zu **PUMA** wird durch eine vorhandene Konverterklasse für die Bildformate gewährleistet.

A.2 Verzeichnisstruktur

```
/prog
  /Calibration
  /Rectification
  /ImageClasses
  /GUI
```

A.2.1 Calibration

Im Unterverzeichnis Calibration befinden sich die Klassen, die der Kalibrierung dienen. Die Zentrale Klasse ist `OpenCVCalibration`. Sie implementiert die Kalibrierung einer Kamera nach Zhang mit Hilfe einer Bildfolge, sowie das Speichern der gewonnen Kameraparameter. Eine `main`-Methode die ebenfalls in dem Verzeichnis liegt, verdeutlicht wie diese Klasse zu verwenden ist.

A.2.2 Rectification

Die Rektifikation besteht nur aus der Klasse `OpenCVRectifier`. Eine Instanz dieser Klasse muß zuerst durch die `readCameraParametersFromFile`-Methode die aus der Kalibrierung gewonnen Daten einlesen. Im Anschluss können durch `computeRectificationMaps` die für die Rektifikation nötigen Interpolationslookuptabellen berechnet werden. Die eigentliche Rektifikation geschieht dann mittels `rectify`. Durch `unrectify` kann die Rektifikation des linken Bildes (oder eines zum linken korrespondierenden Tiefenbildes) wieder rückgängig gemacht werden.

A.2.3 ImageClasses

Als Wrapper für ein Paar `IplImage` dient `StereoImage`. `DisparityImage` hält die Funktionalität zum erzeugen von Disparitätsbildern aus Stereobildpaaren, sowie einige Methoden zur Nachbearbeitung. Die Klasse `StereoImageInfo` wird nur als Hilfsklasse für die GUI verwendet.

A.2.4 GUI

Im GUI Verzeichnis befinden sich neben den Klassen für die GUI die Konverterklassen, welche für das Einbinden von `IplImages` in Qt notwendig sind, sowie `Tracer` und `Clock`, welche zum Debugging genutzt wurden.

A.3 Datenfluss

Nach einmaliger Kalibrierung des Stereokamerasystems stehen die Kalibrierdaten in den Dateien `leftCameraData.txt` und `rightCameraData.txt` zur Rektifikation zur Verfügung. Ein mit diesen Daten initialisiertes `OpenCVRectifier`-Objekt kann im Anschluss sehr effizient Stereobildpaare rektifizieren. Aus einem rektifizierten Stereobildpaar kann dann eine dichte Disparitätskarte erstellt werden.

A.4 GUI

Bild A.2 zeigt das Hauptfenster der GUI. Hier können Stereobildpaare gewählt, Vorverarbeitungsschritte festgelegt, Parameter der Algorithmen verändert und Nachbearbeitungen vorgenommen werden. Das Matching von einzelnen Scanlines kann gezielt im Lines-Tab (Bild A.3) betrachtet werden.

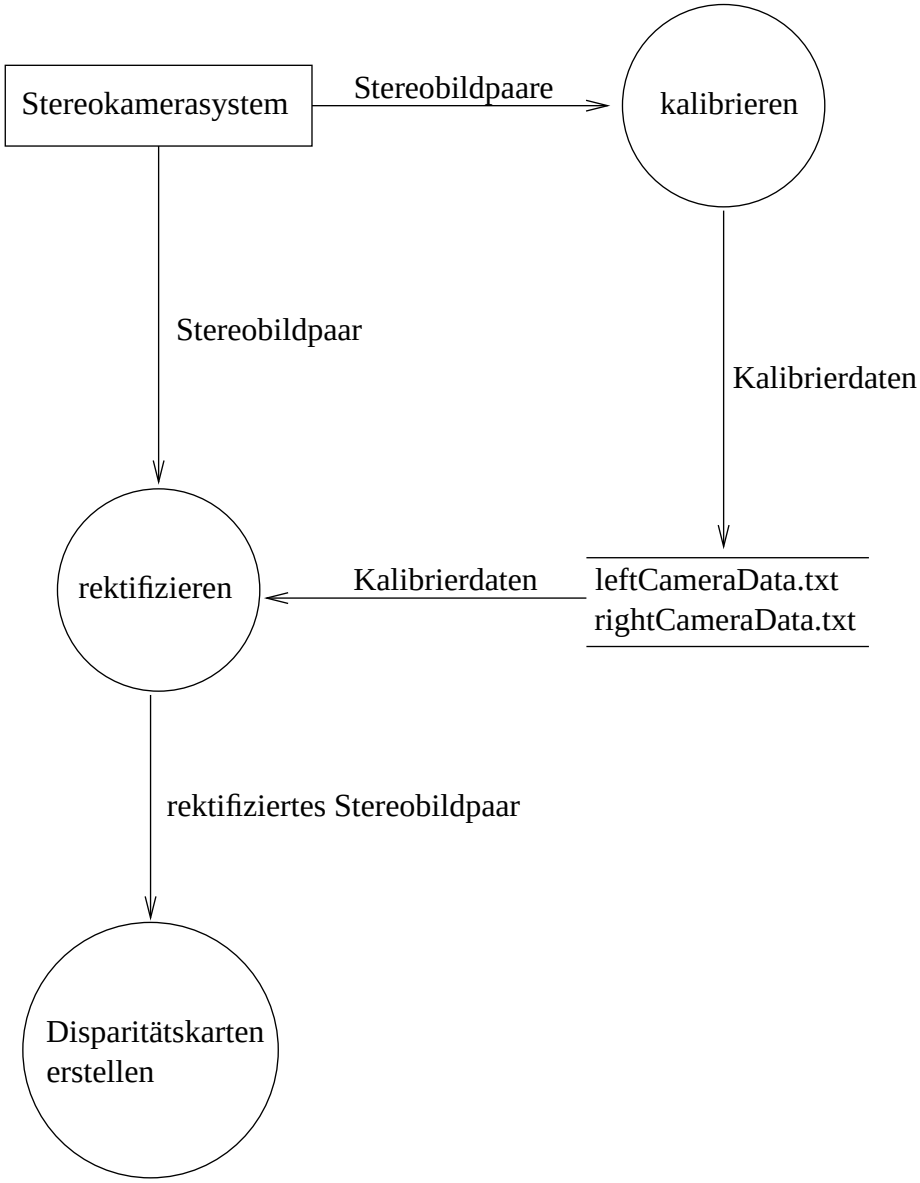

Bild A.1: Datenfluss

Bild A.2: Hauptfenster der GUI

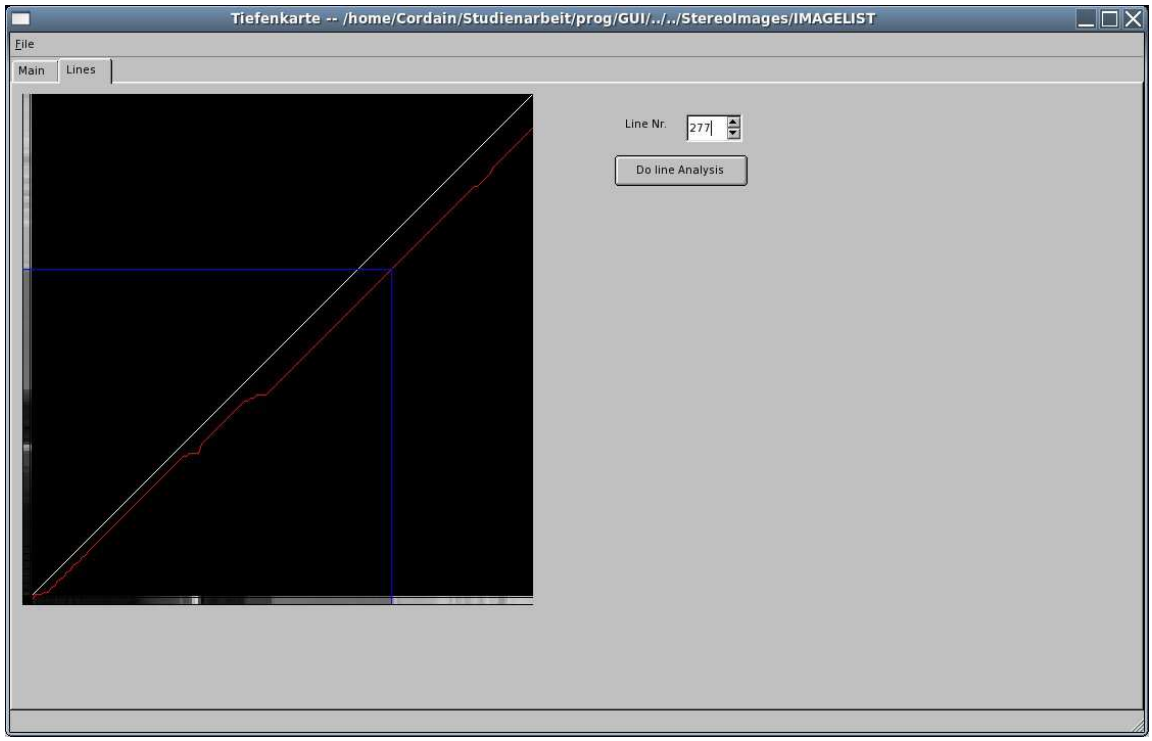

Bild A.3: Visualisierung des Matchings eines einzelnen Scanlinepaars

Literaturverzeichnis

[BT98] Stan Birchfield and Carlo Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *ICCV*, pages 1073–1080, 1998.

[Fal97] Lutz Falkenhagen. Hierarchical block-based disparity estimation considering neighbourhood constraints, 1 1997.

[FTV97] A. Fusiello, E. Trucco, and A. Verri. Rectification with unconstrained stereo geometry. In *British Machine Vision Conference*, pages 400–409, 1997.

[GCA+99] Rafael C. Gonzalez, Jose A. Cancelas, Juan C. Alvarez, Jose A. Fernandez, and Ignacio Alvarez. Dynamic programming stereo vision algorithm for robotic applications. In *Vision Interface '99*, pages 117–124, 5 1999.

[Pau03] Dietrich Paulus. Structure from motion, 5 2003.

[PKG99] Marc Pollefeys, Reinhard Koch, and Luc J. Van Gool. A simple and efficient rectification method for general motion. In *ICCV*, pages 496–501, 1999.

[Sed91] Robert Sedgewick. *Algorithmen*. Addison-Wesley Publishing Company, Reading, MA, 1991.

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.