



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung

Gregory Föll

Stereo Vision: Vergleich verschiedener
Algorithmen zur Lösung des
Korrespondenzproblems

Gregory Föll

Stereo Vision: Vergleich verschiedener Algorithmen
zur Lösung des Korrespondenzproblems

Ausarbeitung eingereicht im Rahmen der Veranstaltung Anwendungen 2
im Studiengang Informatik (Master)
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 9. August 2010

Inhaltsverzeichnis

1 Einführung	4
2 Stereo Vision	6
2.1 Triangulation	7
2.2 Korrespondenz	7
2.2.1 Epipolarebene	8
2.2.2 Standardform / kanonische Stereogeometrie	8
2.2.3 Probleme bei der Korrespondenzfindung	9
2.2.4 Disparität und Tiefe	10
3 Vergleich einiger Algorithmen und Ergebnisse	11
3.1 Block Matching	14
3.2 Semi-Global Matching	15
3.3 Semi-Global Block Matching	17
4 Zusammenfassung und Ausblick	18
Literaturverzeichnis	19

1 Einführung

Im Rahmen des Masterstudiums soll ein robustes und echtzeitfähiges Verfahren für die Verfolgung von mehreren bewegten Objekten in Videosequenzen, als Unterstützung für die Steuerung eines autonomen Modellfahrzeugs, entwickelt werden.

Das zu entwickelnde Verfahren soll mit Methoden der digitalen Bildverarbeitung realisiert werden. Der Einsatz aktive Sensoren, wie zum Beispiel Ultraschall- oder Infrarotsensoren, sollte nach Möglichkeit vermieden werden.

Nach ausführlichen Recherchen wurde festgestellt, dass sich ein modellbasierter Ansatz bei der Objektverfolgung, am besten für das zu lösende Problem eignet. Bei den modellbasierten Methoden wird im Bildmaterial nach bekannten, im Vorfeld modellierten, vereinfachten Modellen des Objektes gesucht. Die Algorithmen können durch das Vorwissen über die Modelle stark vereinfacht werden. Diese Methoden sind sowohl für die Verfolgung von einfachen, starren Objekten, als auch für die von sehr komplexen, dynamischen Objekten gut geeignet.

In vielen Projekten, die sich mit der Verfolgung von Objekten in Videosequenzen beschäftigen, wird eine monokulare Kamera zur Bilderfassung eingesetzt. Recherchen ergaben, dass durch den Einsatz einer Stereo-Kamera, anstelle einer monokularen Kamera, sich einige Vorteile ergeben.

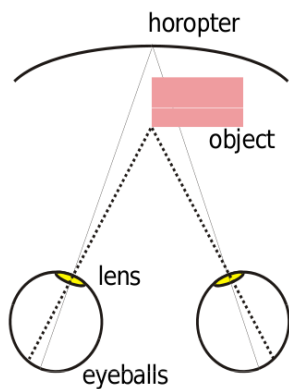
Bei dem sogenannten Stereo Vision wird ein Objekt gleichzeitig von mindestens zwei Positionen aus betrachtet. Die Entfernungsmessung wird hier mit Hilfe der Triangulation realisiert. Das zu erkennende Objekt kann gedreht oder teilweise verdeckt sein und gleiche Objekte können verschiedene Größen aufweisen. Dadurch, dass außer der Stereo-Kamera keine weiteren Sensoren benötigt werden, ergeben sich geringe Hardwarekosten und gegenüber monokularem Ansatz, sind die Algorithmen von annehmbarer Komplexität, was die Verarbeitung in Echtzeit begünstigt.

Das große Problem bei Stereo Vision ist die Korrespondenz zwischen den Pixeln der gleichzeitig aufgenommenen Bilder. Verfahren, die versuchen das Korrespondenzproblem zu lösen, unterscheiden sich in Geschwindigkeit und Genauigkeit.

In dieser Ausarbeitung werden verschiedene Verfahren zur Lösung des Korrespondenzproblems untersucht. Es soll ein Verfahren gefunden werden, welches am besten für eine modellbasierte und echtzeitfähige Objekterkennung und Objektverfolgung geeignet ist.

Um die Stereo-Korrespondenz-Problematik besser zu verstehen, werden zunächst die Grundlagen zu Stereo Vision kurz erläutert, bevor im nächsten Kapitel einige wichtige Verfahren zur Korrespondenzfindung vorgestellt und miteinander verglichen werden.

2 Stereo Vision



Bei der Stereoskopie (Stereo Vision) wird ein Objekt gleichzeitig von zwei Positionen aus betrachtet, die durch eine Basislinie voneinander getrennt sind (zwei Augen), so erscheint es unter verschiedenen Blickwinkeln. Durch die Unterschiede in den beiden entstandenen Bildern, kann mittels Triangulation, die räumliche Ausdehnung des Objektes berechnet werden. Menschen können die beiden Bilder meistens sehr gut automatisch integrieren. Der Stereoeindruck entsteht dabei durch die Disparitätsunterschiede. Fusion der Bilder erfolgt nur, wenn sich das Objekt nicht zu weit von der Horopterfläche entfernt befindet.

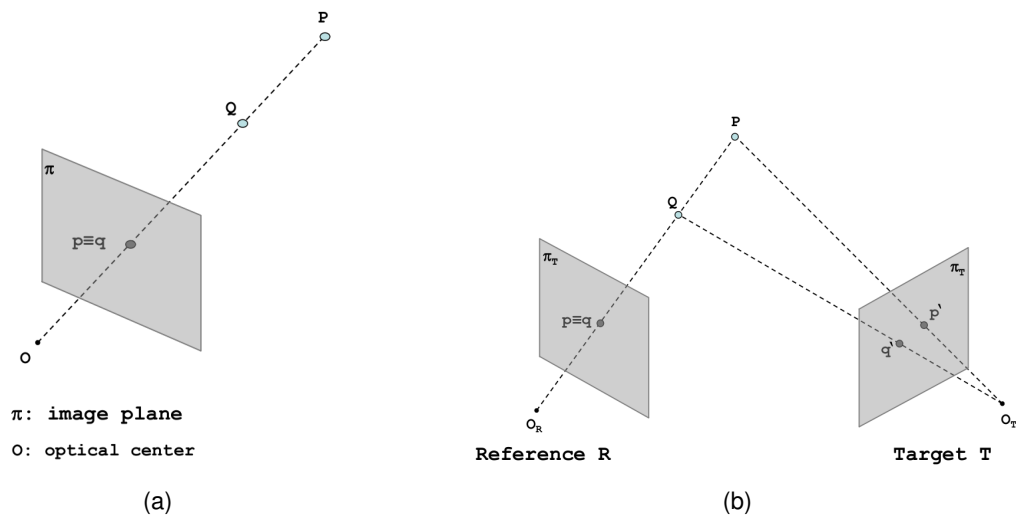
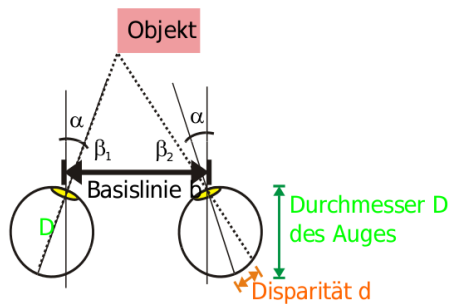


Abbildung 2.1: (a) eine Kamera (b) zwei Kameras

Wie in Abbildung 2.1(a) zu sehen ist, werden bei einer Kamera zwei auf der selben Sichtlinie liegende, reale Punkte Q und P , auf den selben Bildpunkt abgebildet.

Abbildung 2.1(b) zeigt zwei Kameras. Falls es hier gelingt, die korrespondierenden Punkte in den beiden Bildern zu finden, kann mittels Triangulation, die räumliche Ausdehnung des Objektes berechnet werden.

2.1 Triangulation



Das Dreieck wird bei der Triangulation durch zwei Winkel und eine Seite bestimmt. Die Winkel sind die Kombination zwischen Fokussierungsrichtung und Disparität. Die Seite ist die Basislinie, also der Abstand zwischen den beiden Linsenzentren. Als Menschen erkennen wir absolute Abstände ziemlich schlecht und können den Abstand zum Objekt nur schätzen (bis zu 450m).

2.2 Korrespondenz

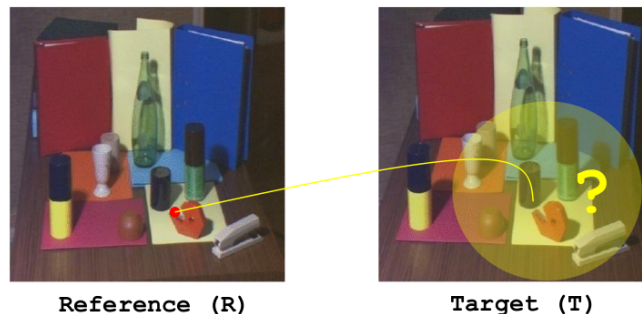


Abbildung 2.2: 2D-Suchraum?

Bei der Frage danach, welche Pixel im Bild R mit welchen Pixeln im Bild T übereinstimmen, könnte man sich vorstellen, dass das gesamte Bild T nach den korrespondierenden Pixeln durchgesucht werden muss.

Glücklicherweise ist das nicht der Fall, denn jeder Ort P im Bild R kann nur die Orte im Bild T zeigen, die auf einer Linie liegen. Diese Linie ist die Projektion von P nach T und heißt Epipolarlinie (siehe dazu Abbildung 2.3). Sie beschränkt die Suche nach Korrespondenzpaaren auf einen 1D-Suchraum.

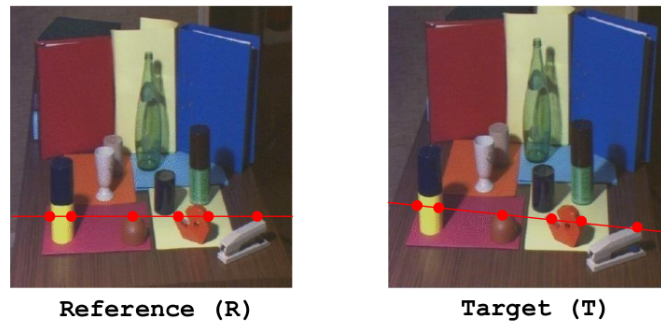


Abbildung 2.3: Epipolarlinie

2.2.1 Epipolarebene

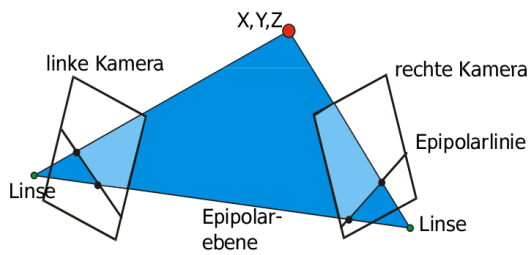


Abbildung 2.4: Epipolarebene

Die Ursprungsorte der beiden Kamerakoordinatensysteme und ein beliebiger Punkt im 3D-Raum, bestimmen eine Epipolarebene. Jeder Punkt in dieser Ebene wird auf dieselben beiden Epipolarlinien in den beiden Bildern projiziert.

2.2.2 Standardform / kanonische Stereogeometrie

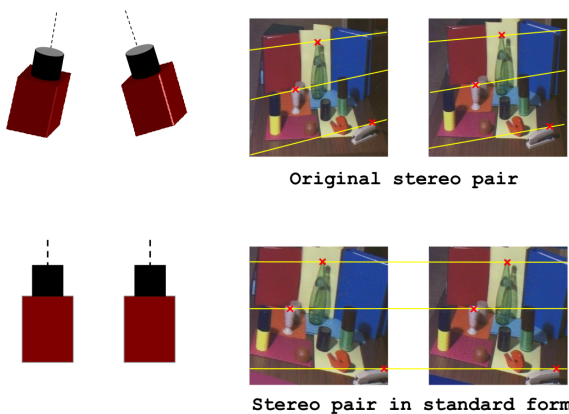


Abbildung 2.5: Standardform

Zur Vereinfachung wird das Stereobild so transformiert, als ob die optischen Achsen der beiden Kameras parallel zueinander sind und in derselben Ebene liegen. Danach liegen Punkte beider Bilder mit gleichem y -Wert auf der gleichen Epipolarlinie.

2.2.3 Probleme bei der Korrespondenzfindung

Das Lösen des Korrespondenzproblems ist leider nicht so trivial wie es vielleicht scheint. Folgende Abbildungen verdeutlichen einige Schwierigkeiten bei der Findung korrespondierender Pixel.



(a) Verzerrungen und Rauschen



(b) Spiegelnde Oberflächen



(c) Verschiedene Ansichten



(d) Homogene Oberflächen

Abbildung 2.6: Probleme bei der Korrespondenzfindung

2.2.4 Disparität und Tiefe

Die Disparität ist der Unterschied der x -Koordinaten zwei korrespondierender Punkte ($x_r - x_t$). Sie wird als Graustufenbild kodiert, wobei hellere Bereiche näher dran sind als dunklere, daraus kann dann eine Tiefenkarte berechnet werden.

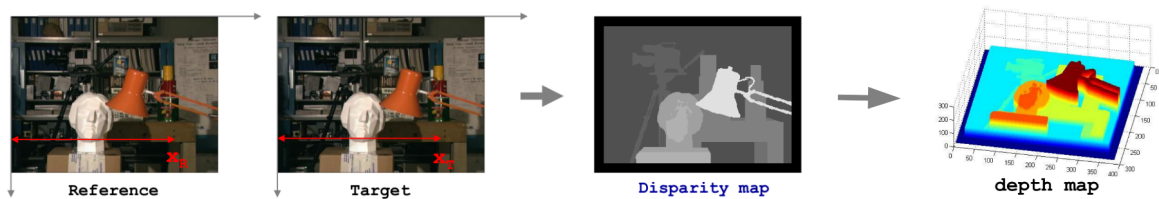


Abbildung 2.7: Disparität und Tiefe

Je näher dabei ein Punkt an der Kamera liegt, desto größer ist die Disparität.

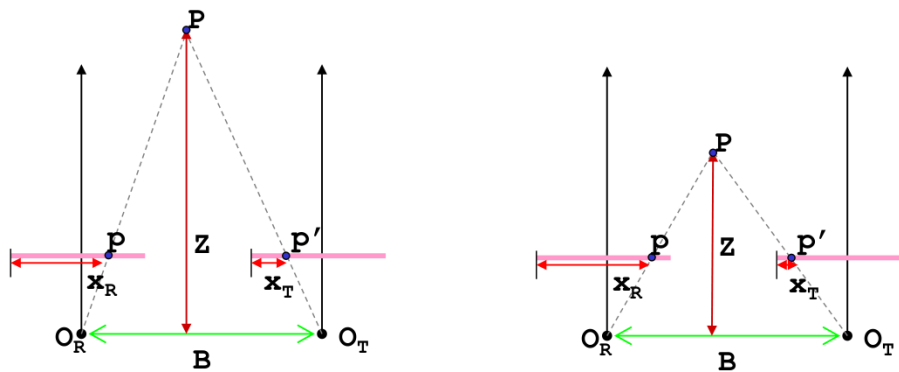


Abbildung 2.8: Disparität

3 Vergleich einiger Algorithmen und Ergebnisse

Das Ziel dieser Arbeit ist es, einen Algorithmus zur Lösung des Korrespondenzproblems zu finden, welcher am besten für eine modellbasierte und echtzeitfähige Objekterkennung und Objektverfolgung geeignet ist. Der gewählte Algorithmus sollte also echtzeitfähig und, für eine modellbasierte Objekterkennung, genau genug sein.

Das Gebiet Stereo Vision ist heute gut erforscht und es gibt eine Vielzahl verschiedener Verfahren und Variationen dieser Verfahren zur Bestimmung von Pixelkorrespondenzen.

Ein einfacher Ansatz ist zum Beispiel das *Intensitäts-basierte Stereo Matching*. Dabei wird für jede Zeile jedes Bildes ein Farbintensitäts-Histogramm erstellt. Diese Histogramme werden dann miteinander verglichen und die ähnlichsten Werte werden als korrespondierende Punkte angenommen.

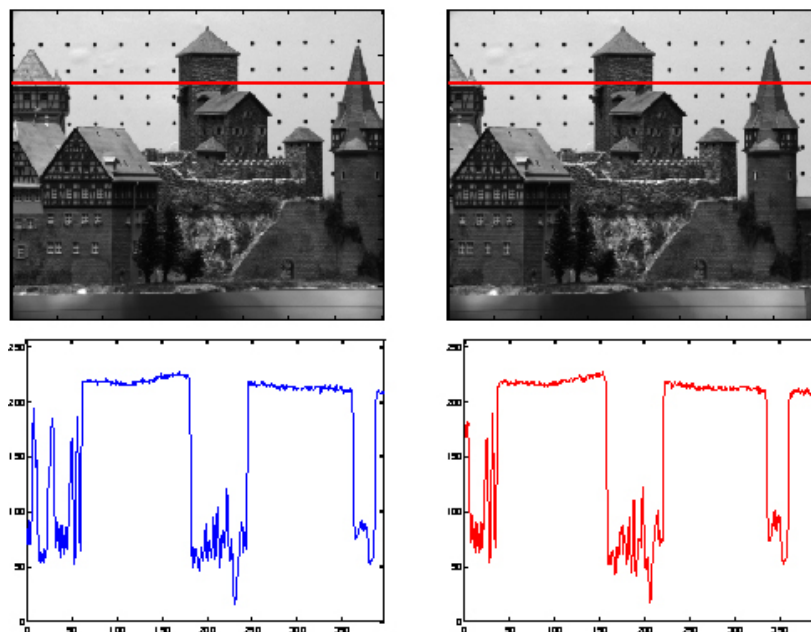


Abbildung 3.1: Histogramme beim Intensity-based Stereo Matching

Das Matching wird durch Minimierung einer Energiefunktion durchgeführt:

$$E_{ij} := |I_1(i, j) - I_2(i, j + D(i, j))| + \lambda |\nabla D(i, j)|$$

$I_k(i, j)$ bezeichnet hierbei die Intensität des Bildes k an der Stelle (i, j) und $D(i, j)$ die Disparität des Bildpunktes (i, j) . Die Energiefunktion wird minimal, wenn die Farbintensitäten im ersten Summanden gleich sind. Um nicht beliebig weit entfernte Bildpunkte mit der selben Intensität miteinander zu vergleichen, wird im zweiten Summanden die Änderung der Disparität in (i, j) betrachtet und mit einem konstanten Faktor λ gewichtet.

Das Problem bei diesem Verfahren ist, dass in den beiden Bildern keine zu großen Änderungen stattfinden dürfen, vor allem in den Bildern, in denen gleich mehrere Kandidaten, in einer Bildzeile, in Frage kommen.

Die verschiedenen Algorithmen werden auch nach Matching-Metriken (lokal und global) unterschieden.

Globale Matching-Verfahren berechnen für jedes Pixel einen 3D-Punkt, indem sie versuchen, jedem Pixel (i, j) im ersten Bild, ein Pixel (i, j) im zweiten Bild zuzuordnen. Bekannte globale Verfahren sind zum Beispiel Dynamic Programming [[Ohta und Kanade \(1985\)](#)] und Graph Cuts [[Kolmogorov und Zabih \(2001\)](#)].

Lokale Verfahren beziehen sich auf einzelne Bildausschnitte (Fenster, Blöcke). Verfahren aus dieser Kategorie sind in der Regel schneller, liefern aber auch oft schlechtere Ergebnisse, bezüglich Genauigkeit, als globale Verfahren.

Es wurden für diese Ausarbeitung viele Verfahren untersucht, da die im Projekt verwendete Bildverarbeitungsbibliothek *OpenCV* immer gute Implementierungen der neusten Forschungsergebnisse zur Verfügung stellt, wurden dort verfügbare Verfahren näher betrachtet. In der aktuellen Version 2.1 implementiert *OpenCV* gleich vier Stereo-Korrespondenz-Algorithmen.

Der *Block-Matching*-Algorithmus nach Kurt Konolige [[Konolige und Agrawal](#)] wurde in dieser Version nochmal deutlich verbessert. Das ist zur Zeit wohl das schnellste verfügbare Verfahren. Es ist aber nicht so genau wie die anderen genannten Algorithmen.

Das *Graph-Cut base Matching* nach Vladimir Kolmogorov [[Kolmogorov und Zabih \(2001\)](#)] liefert zwar sehr akkurate Ergebnisse, ist aber im Vergleich zu *Block-Matching* sehr langsam und dadurch nicht für Echtzeitanwendungen geeignet.

Als "State of the Art"-Verfahren bei Stereo Vision gilt zur Zeit das *Semi-Global Matching* nach Heiko Hirschmüller [[Hirschmüller \(2005\)](#)]. *OpenCV* implementiert außer des Originals auch noch eine modifizierte Version dieses Verfahrens, genannt *Semi-Global Block Matching*. Es

ist schneller und stabiler als das Original. Es ist viel schneller als der *Graph-Cut*-Algorithmus bei annähernd gleich guten, oft sogar besseren, Ergebnissen.

Auf der Middlebury Stereo Vision Webseite [Scharstein und Szeliski] findet man eine Ranking-Datenbank aller aktuellen Stereo-Algorithmen. Es gibt dort mehrere Datensätze von Stereoaufnahmen [Scharstein und Szeliski (2002, 2003); Scharstein und Pal (2007); Hirschmüller und Scharstein (2007)], die bei der Entwicklung und beim Testen eigener Algorithmen hilfreich sein können. Es gibt auch die Möglichkeit, online, den eigenen Algorithmus mit allen anderen Algorithmen in der Ranking-Datenbank zu vergleichen.

Algorithm	Avg. Rank	Tsukuba (100x100)			Venus (100x100)			Teddy (100x100)			Cones (100x100)			Average percent of bad pixels (explanation)
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
ArdantngBP [17]	4.7	1.11	1.37	5.79	0.10	0.71	1.44	4.22	7.06	11.8	2.45	7.97	7.32	4.23
CoopRegion [41]	4.8	0.87	1.16	4.61	0.11	0.21	1.54	2.16	0.31	13.0	2.74	7.18	0.01	4.41
DoubleBP [35]	6.6	0.88	1.29	4.76	0.13	0.45	1.87	3.53	8.30	9.63	2.90	8.78	7.79	4.10
OutlierConf [42]	7.5	0.88	1.43	4.74	0.18	0.26	2.40	5.01	9.12	12.8	2.78	8.57	6.99	4.60
SubPixDoubleBP [30]	10.1	1.24	1.76	5.98	0.12	0.46	1.74	3.42	8.38	10.0	2.93	8.73	7.91	4.39
surfStereo [79]	11.0	1.28	1.65	6.78	0.19	0.20	2.61	3.12	5.10	8.65	2.82	7.95	0.26	4.06
WarpMat [55]	12.1	1.16	1.35	6.04	0.18	0.24	2.44	5.02	9.30	13.0	3.49	8.47	0.01	4.28
Undi+Ov:Seg [48]	16.0	1.89	2.22	7.22	0.11	0.22	1.34	6.51	9.98	16.4	2.92	8.00	7.90	5.39
GC+SegmBorder [57]	17.2	1.47	1.82	7.86	0.19	0.31	2.44	4.25	5.55	10.9	4.99	5.78	8.66	4.52
AdaptOvrSegM [33]	17.9	1.69	2.04	8.64	0.14	0.20	1.47	7.04	11.1	16.4	3.60	8.96	0.04	5.59
GC+Sup [64]	19.4	1.45	1.83	7.71	0.14	0.26	1.90	6.88	13.2	16.1	2.94	8.89	8.32	5.80
PlaneFitBP [32]	19.8	0.97	1.83	5.26	0.17	0.51	1.71	6.65	12.1	14.7	4.17	10.7	10.6	5.78
SvmBP+occ [7]	20.5	0.97	1.75	5.09	0.16	0.33	1.19	6.47	10.7	17.0	4.79	10.7	10.9	5.92
AdaptDispCalib [36]	22.2	1.19	1.42	6.15	0.23	0.34	2.50	7.80	13.6	17.3	3.62	9.33	9.72	6.10
Segm+VidC [4]	22.8	1.30	1.57	6.92	0.19	0.26	3.76	5.00	6.54	12.5	3.72	8.62	10.2	5.40
C-SemIGlob [19]	23.0	2.61	3.29	9.89	0.25	0.57	3.24	5.14	11.8	13.0	2.77	8.35	8.20	5.76
MultiResGC [49]	23.3	0.90	1.32	4.82	0.45	0.84	3.32	6.46	11.8	17.0	4.34	10.5	9.77	6.04
SO+border [29]	23.6	1.29	1.71	6.83	0.25	0.53	2.26	7.02	12.2	16.3	3.90	9.85	10.2	6.03
RandomVox [89]	25.2	4.85	5.54	17.7	0.13	0.45	1.86	3.40	9.54	14.8	2.62	7.93	7.54	6.33
DistinctM [27]	25.6	1.21	1.75	6.39	0.35	0.69	2.63	7.45	13.0	18.1	3.91	9.91	8.32	6.14
OverSegmBP [26]	26.4	1.69	1.97	8.47	0.51	0.68	2.69	6.74	11.9	15.8	3.19	8.81	8.89	6.11
MVSegBP [65]	26.6	1.06	1.78	6.57	0.20	0.61	2.07	6.51	11.7	14.8	3.29	11.7	14.5	6.34
Curvelet+supWgt [74]	27.0	1.40	1.84	7.42	1.00	1.11	4.42	7.85	10.8	16.8	3.82	8.28	8.24	5.75
LocallyConsist [69]	27.5	1.70	2.21	8.67	0.16	0.32	1.63	8.68	13.0	17.0	4.19	10.8	9.72	6.33
SegmentSupport [28]	27.7	1.25	1.62	6.68	0.25	0.61	2.59	8.43	14.2	18.2	3.77	9.87	9.77	6.14
CostAggr+occ [39]	27.8	1.38	1.95	7.14	0.44	1.13	4.87	6.80	11.9	17.3	3.60	8.57	9.36	6.20
RegionTreeDP [11]	29.6	1.39	1.64	6.85	0.22	0.57	1.94	7.42	11.9	16.8	6.31	11.9	11.8	6.56
EnhancedBP [24]	31.2	0.94	1.74	5.05	0.35	0.27	4.34	8.11	13.3	18.5	5.09	11.1	11.4	6.69
PJTV3 [63]	31.5	1.77	3.86	9.42	0.42	0.95	5.72	7.02	14.2	18.3	2.49	9.11	6.56	6.64
GradAdaptWgt [60]	32.2	2.26	2.63	8.99	0.99	1.39	4.92	8.00	13.1	18.6	2.61	7.07	7.43	6.55
AdaptWeight [12]	33.2	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	9.76	6.57
SegTreeDP [22]	33.8	2.21	2.76	10.3	0.46	0.60	2.44	2.58	15.2	18.4	3.23	7.86	8.83	6.82
MultiOvr [51]	34.2	1.20	1.81	6.31	0.43	0.69	3.36	7.09	14.0	17.2	5.42	12.6	12.5	6.89
InteriorPILP [34]	34.7	1.27	1.62	6.82	1.15	1.67	12.7	8.07	11.9	18.7	3.92	9.68	9.62	7.26
ImproveSubPix [25]	35.8	3.00	3.61	10.9	0.88	1.47	7.10	7.12	12.4	16.6	2.96	8.72	11.5	6.90
SCMIGlob [6]	37.6	3.26	3.96	12.8	1.00	1.57	11.3	5.02	14.2	17.3	3.06	9.75	8.90	7.50
BP+DirectedDiff [61]	38.0	2.90	4.47	15.1	0.65	1.20	4.52	5.07	14.7	15.7	2.94	12.6	7.50	7.29
FastBilateral [68]	39.4	2.38	2.80	10.4	0.34	0.92	4.55	9.83	15.3	20.3	3.10	9.31	8.59	7.31
RealTimeARW [81]	40.1	1.26	1.67	6.83	0.33	0.65	3.56	10.7	18.3	21.3	4.81	12.6	10.7	7.90
CostRelaxW [59]	41.8	2.91	3.49	11.4	0.60	1.11	3.9	2.92	13.7	20.9	3.59	9.43	10.3	7.88
PlaneFITSGM [83]	42.2	3.13	4.20	14.9	1.08	1.87	5.16	5.68	11.6	17.1	3.79	9.26	11.3	8.21
BP+ompressed [56]	42.4	2.68	3.63	9.59	1.33	1.89	9.09	8.36	13.9	16.4	3.71	9.85	9.92	7.53
RealTimeBP [21]	43.6	1.49	3.40	8.87	0.77	1.90	9.00	8.72	13.2	17.2	4.61	11.6	17.4	7.69
RealTimeV [65]	44.2	1.71	2.22	8.74	1.55	0.87	2.08	3.90	15.0	19.5	5.66	12.3	15.4	7.88
VariableCross [44]	44.5	1.99	2.65	8.77	0.62	0.96	3.20	9.75	15.1	18.2	6.28	12.7	12.9	7.60
ZOP+occ [37]	45.0	2.91	3.56	9.73	0.24	0.49	2.76	10.9	15.4	20.6	5.42	10.8	12.5	7.79
GC+SegAggr [47]	45.9	1.74	2.11	9.23	0.41	0.28	3.97	8.08	14.3	19.8	7.07	12.9	12.9	8.07
VarNFD [54]	46.6	3.97	5.23	14.9	1.28	1.76	3.78	9.34	14.3	21.0	4.14	9.91	11.4	8.17
FastAggr [45]	48.4	1.16	2.11	8.06	1.03	1.75	6.43	9.04	15.2	20.2	5.37	12.6	11.9	8.24
GC+occ [2]	48.6	1.19	2.01	6.24	1.64	2.19	6.75	11.2	17.1	19.8	5.36	12.4	13.0	8.26
MultiScale [51]	49.8	1.27	1.85	6.48	1.76	1.77	6.66	12.6	17.6	22.6	4.88	11.8	13.1	8.71

Abbildung 3.2: Middlebury Stereo Evaluation

Der in Abbildung 3.2 rot umrandete Algorithmus ist das *Semi-Global Matching* nach Heiko Hirschmüller und das blau umrandete ist das *Graph-Cut base Matching* nach Vladimir Kolmogorov. Middlebury Stereo Evaluation bewertet leider nur die Genauigkeit der Verfahren. Die Geschwindigkeit wird nicht berücksichtigt, deswegen wird der *Block-Matching*-Algorithmus hier erst gar nicht aufgeführt. Verfahren, die in der Tabelle vor dem *Semi-Global Matching* stehen, die also genauere Ergebnisse liefern, sind in der Regel nicht für Echtzeitanwendungen geeignet.

Im Folgenden werden die Vorgehensweisen der beiden in Frage kommenden Ansätze *Block-Matching* und *Semi-Global Matching* detaillierter beschrieben. Die Ergebnisse werden anhand der bekannten *Tsukuba*-Stereoaufnahme (siehe Abbildung 2.7) verdeutlicht.

3.1 Block Matching

Bei Block Matching Verfahren werden zwei gleich große Blöcke aus den beiden Bildern miteinander verglichen. Es wird eine $n \times m$ -Matrix im ersten Bild mit allen möglichen Blöcken im zweiten Bild verglichen. Der mittlere quadratische Fehler (*MSE*) dient dabei als Ähnlichkeitsmaß für die Pixelwerte der beiden Blöcke. Das linke Bild I_L und das rechte Bild I_R sind mit Grauwerten G definiert als $I_L(i, j) = G_L(i, j)$ und $I_R(i, j) = G_R(i, j)$. Der *MSE* mit $n = m = 2k + 1$ ist dann definiert als

$$MSE(x, y, \Delta) = \frac{1}{n \cdot m} \sum_{i=-k}^k \sum_{j=-k}^k (|G_L(x + i, y + j) - G_R(x + i + \Delta, y + j)|^2)$$

Der Offset Δ ist die Differenz ($x_L - x_R$) zwischen den Spalten beider Bilder, um den der $n \times m$ große Block innerhalb des Suchraums pixelweise verschoben wird.

Die Disparität D ist als horizontale Distanz zwischen den Blöcken beider Bilder definiert, und zwar dort, wo der *MSE* minimal ist. Der horizontale Suchraum im rechten Bild wird durch einen maximalen Disparitätswert d_{max} begrenzt. D ergibt sich somit aus

$$D = \min_{|\Delta| \leq d_{max}} \{MSE(x, y, \Delta)\}$$

Um Außreiser zu eliminieren, werden die Blockdisparitäten nach der Berechnung mit einem Medianfilter bearbeitet. Abschließend wird eine Pixelselektion durchgeführt, um für jedes Pixel des Bildes eine Disparität zu berechnen.

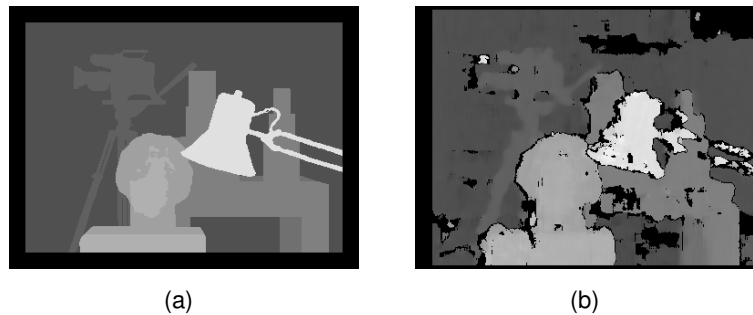


Abbildung 3.3: Tsukuba: (a) Referenzdisparitätskarte (b) berechnete Disparitätskarte nach [Konolige und Agrawal]

3.2 Semi-Global Matching

Das Semi-Global Matching (SGM) Verfahren approximiert die Minimierung einer globalen Kostenfunktion. Dabei werden Matchingkosten pixelweise horizontal, vertikal und diagonal akkumuliert und mit einem kantenerhaltendem Glattheitskriterium verknüpft. Das Glattheitskriterium wird eingesetzt, da die Vergleichskosten oft nicht eindeutig sein können und dadurch falsche Korrespondenzen niedrigere Vergleichskosten haben können als korrekte.

Durch das Glattheitskriterium werden alle Pixel des Bildes miteinander verknüpft. Die Minimierung erfolgt sehr effizient mit der gleichen Komplexität die typisch ist für schnelle korrelationsbasierte Ansätze (linear zur Anzahl der Pixel und Disparitätsstufen). Zur Berechnung der pixelweisen Matchingkosten wird Transinformation eingesetzt, wodurch SGM sehr robust gegenüber großer radiometrischer Differenzen und auch kleinen Störungen wie z.B. Spiegelungen ist.

Transinformation oder mutual information ist ein Maß, das die Stärke des statistischen Zusammenhangs zweier Größen, in diesem Fall zweier Bilder, angibt. Dabei wird für jede der Größen die Entropie, meist die Shannonentropie und die Vereinigungsentropie ermittelt. Die Transinformation ergibt sich aus der Summe der Entropien der Größen abzüglich ihrer Vereinigungsentropie.

Das Glattheitskriterium wird zunächst entlang unabhängiger eindimensionaler Pfade der Richtung r durch Definition der Pfadkosten $L_r(p, d)$ rekursiv formuliert.

$$L_r(p, d) = C(p, d) + \min[L_r(p - r, d), L_r(p - r, d + 1) + P_1, L_r(p - r, d - 1) + P_1, \min_i L_r(p - r, i) + P_2] - \min_l L_r(p - r, l)$$

Der erste Term beschreibt die ursprünglichen Vergleichskosten. Der zweite Term addiert die minimalen Pfadkosten des vorherigen Pixel $p - r$ gewichtet entsprechend des Disparitätsüberganges mit den Kostentermen P_1 und P_2 . Dabei wird zwischen kleinen Disparitätsunstetigkeiten ($|\Delta d| = 1px$) und Disparitätssprüngen ($|\Delta d| > 1px$) unterschieden, um auch schräge und gebogene Oberflächen korrekt interpretieren zu können.

Disparitätssprünge gehen oft mit Intensitätsunterschieden einher. Um das auszunutzen, wird P_2 an den Intensitätsgradienten entlang des Pfades angepasst.

$$P_2 \sim |I(p) - I(p - r)|^{-1}$$

Dabei muss immer $P_1 < P_2$ gelten.

Der letzte Term in der ersten Gleichung verhindert, dass L_r entlang des Pfades kontinuierlich ansteigt. Damit ist die obere Grenze $L_r \leq C_{max} + P_{2,max}$. Da der Term konstant über alle Disparitäten eines Pixel ist, wird das Minimum über d (relevante Information) dadurch nicht beeinflusst.

Um eine quasi-globale Abhängigkeit über das gesamte Bild zu erreichen, werden die Pfadkosten aus mehreren Richtungen zu jedem Pixel berechnet. Anschließend werden die Summenkosten S berechnet.

$$S(p, d) = \sum_r L_r(p, d)$$

Anzahl und Lage der Pfade beeinflussen die Qualität der Disparitätskarte und müssen je nach Anwendung angepasst werden.

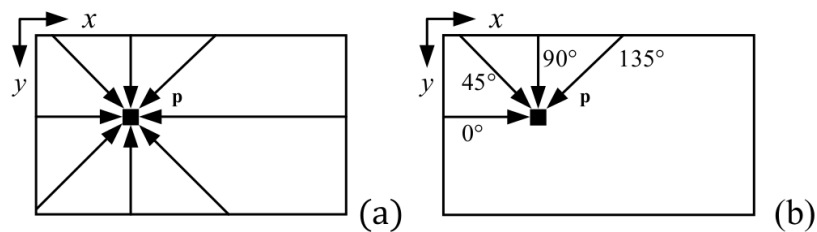


Abbildung 3.4: Mögliche Lage von Pfadrichtungen für (a) acht und (b) vier Pfade.

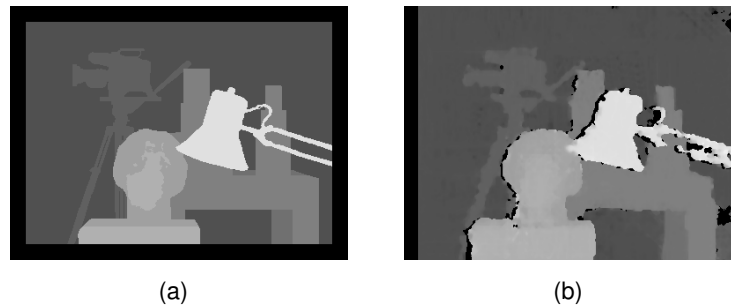


Abbildung 3.5: Tsukuba: (a) Referenzdisparitätsskarte (b) berechnete Disparitätsskarte nach [Hirschmüller (2005)]

3.3 Semi-Global Block Matching

Dieser Algorithmus ist eine modifizierte Version von [Hirschmüller (2005)] mit folgenden Unterschieden:

- Um den Speicherverbrauch zu reduzieren, werden Pfade nur in 5 Richtungen, anstelle von 8 Richtungen berücksichtigt.
- Das Matching erfolgt nicht pixelweise, wie bei der originalen Implementierung, sondern blockweise.
- Die Transinformation-Kosten-Funktion ist nicht implementiert. Stattdessen wird eine Sub-Pixel-Metrik nach [Birchfield und Tomasi (1998)] eingesetzt.
- Der Algorithmus beinhaltet auch einige Pre- und Post-Processing-Schritte aus [König und Agrawal].

Wie oben bereits erwähnt, ist diese Implementierung des Algorithmus schneller und stabiler als die originale.

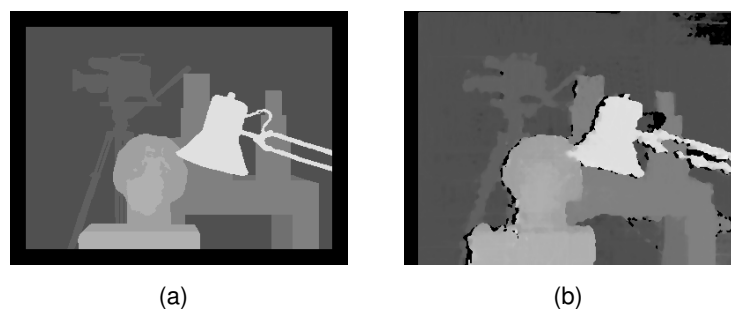


Abbildung 3.6: Tsukuba: (a) Referenzdisparitätsskarte (b) berechnete Disparitätsskarte mit Semi-Global Block Matching

4 Zusammenfassung und Ausblick

Das Ziel dieser Ausarbeitung war es, verschiedene Verfahren zur Lösung des Korrespondenzproblems zu untersuchen und miteinander zu vergleichen, um ein geeignetes Verfahren für eine modellbasierte und echtzeitfähige Objekterkennung und Objektverfolgung in Videosequenzen zu finden.

Um die Stereo-Korrespondenz-Problematik besser verstehen zu können, wurden zunächst die Grundlagen des "3D-Sehens" vorgestellt und dann einige Stereo-Korrespondenz-Algorithmen näher erläutert und verglichen.

Für das zu lösende Problem wurden drei Algorithmen ausgewählt. Wie in den Abbildungen [3.3](#), [3.5](#) und [3.6](#) zu sehen ist, unterscheiden sich die Ergebnisse der beiden Algorithmen *Semi-Global Matching* und *Semi-Global Block Matching* kaum voneinander, während der *Block-Matching*-Algorithmus eine deutlich schlechtere Disparitätskarte liefert.

Zum jetzigen Zeitpunkt scheint *Semi-Global Block Matching* der erfolgversprechendste Kandidat zu sein. Für die Verarbeitung einer Stereoaufnahme mit einer Auflösung von 640×480 Pixel, benötigt dieser Algorithmus $\sim 300ms$, während *Semi-Global Matching* rund $1s$ dafür benötigt. Dabei liefert er recht akkurate Ergebnisse.

Semi-Global Matching wird zur Zeit in vielen Projekten, die sich mit Stereo Vision in Echtzeit beschäftigen, eingesetzt. Die originale Implementierung des Verfahrens ist leider nicht für eine parallele Verarbeitung geeignet, es gibt aber Variationen des Verfahrens, die sich gut parallelisieren lassen. Um die CPU zu entlasten und eine Verarbeitung in Echtzeit zu garantieren, wird im nächsten Projekt untersucht, ob sich das Verfahren auf eine GPU portieren lässt.

Der *Block-Matching*-Algorithmus ist immer noch um einiges schneller als *Semi-Global Block Matching*. Ob er auch für eine modellbasierte Objekterkennung akkurat genug ist, muss im *Projekt 2* genauer untersucht werden.

Literaturverzeichnis

- [Bade 2003] BADE, Richard: *Modifikation einer geeigneten Stereobildverarbeitungs- methode für die Anwendung in einer Echtzeitumgebung*, Otto-von-Guericke-Universität Magdeburg, Diplomarbeit, Juni 2003. – URL http://www-ivs.cs.uni-magdeburg.de/EuK/mitarbeiter/ribade/diplomarbeit_rbj.pdf
- [Banz u. a.] BANZ, Christian ; FLATT, Holger ; BLUME, Holger ; PIRSCH, Peter: *Hardware- Architektur zur echtzeitfähigen Berechnung dichter Disparitätskarten* / Leibniz Universität Hannover, Institut für Mikroelektronische Systeme. – Forschungsbericht
- [Birchfield und Tomasi 1998] BIRCHFIELD, Stan ; TOMASI, Carlo: *Discontinuities by Pixel- to-Pixel Stereo*. In: *IEEE International Conference on Computer Vision* (1998)
- [Hirschmüller und Scharstein 2007] HIRSCHMÜLLER, H. ; SCHARSTEIN, D.: *Evaluati- on of cost functions for stereo matching*. In: *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)* (2007), June. – URL http://bj.middlebury.edu/~schar/papers/evalCosts_cvpr07.pdf
- [Hirschmüller 2005] HIRSCHMÜLLER, Heiko: *Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2005), June
- [Jähne 2002] JÄHNE, Bernd: *Digitale Bildverarbeitung*. 5. Springer, 2002
- [Kolmogorov und Zabih 2001] KOLMOGOROV, Vladimir ; ZABIH, Ramin: *Computing visual correspondence with occlusions using graph cuts*. In: *In International Confe- rence on Computer Vision*, URL [http://www.cs.cornell.edu/rdz/Papers/ KZ-ICCV01-tr.pdf](http://www.cs.cornell.edu/rdz/Papers/KZ-ICCV01-tr.pdf), 2001, S. 508–515
- [Konolige und Agrawal] KONOLIGE, Kurt ; AGRAWAL, Motilal: *Frame-Frame Matching for Realtime Consistent Visual Mapping*. – URL [http://citeseerx.ist.psu.edu/ viewdoc/download?doi=10.1.1.68.4325&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.4325&rep=rep1&type=pdf)
- [Linkenheil 2005] LINKENHEIL, Matthias: *Graph-Cut-Segmentierung für die medizinische Bildverarbeitung* / Deutsches Krebsforschungszentrum Heidelberg Abteilung Medizinische und Biologische Informatik. URL [http://www.dkfz.de/de/mbi/pub/pdf/ tr170.pdf](http://www.dkfz.de/de/mbi/pub/pdf/tr170.pdf), August 2005. – Forschungsbericht

- [Ohta und Kanade 1985] OHTA, Y. ; KANADE, T.: Stereo by intra- and inter-scanline search using dynamic programming. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1985), S. 139–154
- [Scharstein und Pal 2007] SCHARSTEIN, D. ; PAL, C.: Learning conditional random fields for stereo. In: *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)* (2007), June. – URL http://bj.middlebury.edu/~schar/papers/LearnCRFstereo_cvpr07.pdf
- [Scharstein und Szeliski] SCHARSTEIN, D. ; SZELISKI, R.: <http://vision.middlebury.edu/stereo/eval/>
- [Scharstein und Szeliski 2002] SCHARSTEIN, D. ; SZELISKI, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: *International Journal of Computer Vision* (2002), April-June, Nr. 47(1/2/3), S. 7–42. – URL <http://vision.middlebury.edu/stereo/data/taxonomy-IJCV.pdf>
- [Scharstein und Szeliski 2003] SCHARSTEIN, D. ; SZELISKI, R.: High-accuracy stereo depth maps using structured light. In: *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003)* (2003), June, Nr. volume 1, S. 195–202. – URL <http://www.middlebury.edu/~schar/papers/structlight/>
- [Tönnies 2005] TÖNNIES, K.D.: *Grundlagen der Bildverarbeitung*. Pearson Studium, 2005
- [Veksler 2006] VEKSLER, Olga: Reducing Search Space for Stereo Correspondence with Graph Cuts. In: *British Machine Vision conference September 2006 (BMVC'06)* (2006), September, Nr. vol. II, S. 709–718. – URL <http://www.csd.uwo.ca/faculty/olga/Papers/bmvc06-257-final.pdf>
- [Zureiki u. a. 2008] ZUREIKI, Ayman ; DEVY, Michel ; CHATILA, Raja: Stereo Matching and Graph Cuts. (2008). – URL http://sciyo.com/articles/show/title/stereo_matching_and_graph_cuts