

Gestaltsanalyse komplexer Objekte bei kontrollierter Bewegung

Diplomarbeit
(CV-Bericht 27)

Dirk Mehren,
(Matr.-Nr.: 113309)

Volker Rodehorst
(Matr.-Nr.: 115076)

Berlin, den 13. März 2003

Computer Vision
Fachbereich Informatik
Prof. Reinhard Klette

Technische Universität Berlin
Franklinstraße 28 / 29
FR 3-11, D-10587 Berlin

Zusammenfassung

In dieser Diplomarbeit soll die dreidimensionale Oberflächenrekonstruktion von Objekten einer Szene durch Auswertung der Grauwertbilder einer CCD-Videokamera beschrieben werden. Die Aufnahmegerometrie wird mit einem Kalibrierungsverfahren bestimmt. Durch die Verwendung einer kontrollierten Bewegung anhand eines Drehtellers kann die Verschiebungsvektorbestimmung und die Berechnung der Tiefeninformation erleichtert werden.

Dazu wurden bekannte Optical-Flow-Techniken erprobt und eigene merkmalsbasierte Methoden entwickelt. Die experimentellen Ergebnisse mit realen und synthetischen Daten zeigen, daß die vorgestellten Verfahren aufgrund ihres erfolgsversprechenden Ansatzes in verschiedene Richtungen weiterentwickelt werden sollten.

Hiermit versichern wir, daß die Diplomarbeit selbstständig verfaßt wurde und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt worden sind.

Inhaltsverzeichnis

1 Einleitung	4
2 Kamera-Kalibrierung	6
2.1 Techniken der Kamera-Kalibrierung	6
2.2 Das Kameramodell	9
2.3 Kamera-Kalibrierung anhand coplanarer Punkte	13
2.3.1 Stufe 1: Berechnung der 3D-Orientierung und XY-Position	13
2.3.2 Stufe 2: Berechnung der internen Kameraparameter	15
2.4 Kamera-Kalibrierung mit nicht-coplanaren Punkten	17
2.4.1 Stufe 1: Berechnung der 3D-Orientierung, XY-Position und Skalierung	17
2.4.2 Stufe 2: Berechnung der internen Kameraparameter	18
2.5 Experimentelle Ergebnisse	19
2.5.1 Generierung synthetischer Daten	19
2.5.2 Durchführung einer Kalibrierung	19
2.5.3 Fehlerbetrachtung	22
2.5.4 Kalibrierung von Realdaten	26
3 Vorverarbeitung	29
3.1 Globale Grauwertanpassung	29
3.2 Dynamischer Schwellwert	29
3.3 Dynamische Histogrammegalorisierung	32
3.4 Medianfilter	33
4 Merkmalsextraktion	34
4.1 Subpixelgenaue Punktbestimmung	34
4.1.1 Grundsätzliche Überlegungen	34
4.1.2 Anmerkung	35
4.2 Der Marr-Hildreth-Operator	37
4.2.1 Anforderungen	37
4.2.2 Laplacian-of-Gaussian (LoG)	37
4.2.3 Modifikationen	39
4.2.4 Eigenschaften	39
4.2.5 Subpixel-Informationen	41
4.3 Der Canny-Operator	42
4.3.1 Filterfunktion	42
4.3.2 Gradientenoperator	42
4.3.3 Kanten-Nachbearbeitung	43
4.3.4 Eigenschaften	46

4.4	Monotonie-Operator	48
4.5	Corner Response Funktion	49
4.6	Anwendung	49
4.7	Beispiele	50
5	Verschiebungsvektor-Bestimmung	54
5.1	Techniken zur Bestimmung des optischen Flußes	54
5.2	Motionkonstraint	57
5.3	Ausgezeichnete Verschiebungsvektoren	61
5.4	Verschiebungsvektorfelder	63
5.4.1	Segmentierung	63
5.4.2	Interpolation	63
6	Gestaltsrekonstruktion	66
6.1	Tiefe aus Bewegung	66
6.1.1	Überbestimmtes Gleichungssystem mit Kenntnis der Rotation	66
6.1.2	Gleichungssystem ohne Kenntnis des Rotationswinkel	68
6.2	Praxis	71
7	Ergebnisse	73
7.1	Gestaltsanalyse mit synthetischen Verschiebungsvektoren	73
7.2	Rekonstruktion der Gestalt mit Optical-Flow-Techniken	78
7.3	Merkmalsbasierte Gestaltsrekonstruktion	82
7.4	Bewertung	86
7.4.1	Ausblick und Fortsetzung	86
8	Programmbeschreibung	87
8.1	Benutzeranleitung	87
8.1.1	Programmsteuerung	87
8.1.2	Kamera-Kalibrierung	87
8.1.3	Vorverarbeitung	92
8.1.4	Merkmalsextraktion	93
8.1.5	Verschiebungsvektorbestimmung	95
8.1.6	Visualisierung	96
8.2	Dateiformate	99
8.2.1	Aufbau der Konfigurationsdatei .cfg	99
8.2.2	Aufbau der Kalibrierungsdateien	103
8.2.3	Format der Gauwertbilder .pgm	104
8.2.4	Format der Bilddaten I8 und F8	104
8.2.5	Format der Verschiebungsvektorkarten .dpm	104
8.2.6	Format der Tiefenkarten .dep	105

Kapitel 1

Einleitung

In diesem Projekt soll die Gestaltsanalyse komplexer Objekte bei kontrollierter Bewegung experimentell durchgeführt werden. Mit dem Begriff *Gestaltsanalyse* ist die Gewinnung von Tiefeninformation zur dreidimensionalen Oberflächenrekonstruktion von Objekten einer Szene gemeint. Die hier beschriebenen Ansätze verfolgen das Ziel, aus mehreren zweidimensionalen Bildern eine dreidimensionale geometrische Beschreibung der beobachteten Szene zu erzeugen. Durch Verwendung von unterschiedlichen Ansichten mit bekannter Aufnahmegerometrie können die Objektmerkmale einander zugeordnet und die räumlichen Parameter berechnet werden. Als bildgebender Sensor soll eine handelsübliche CCD Videokamera dienen.

Die erforderlichen Ansichten sind durch Veränderung der Objektlage, Variation der Kameraposition oder auch durch die Verwendung mehrerer Kameras zu erreichen. Wir haben den ersten Ansatz gewählt und die Bewegung durch einen *Drehteller* realisiert. Der Einsatz einer drehenden Scheibe, auf der die zu rekonstruierenden Objekte plaziert werden, erlaubt eine maximale Kontrolle der Bewegungsparameter. Bei dieser Geometrie existiert nur eine Rotationsachse, die durch den Mittelpunkt des Drehtellers verläuft.

Weil keine Verschiebung des Objekts in der Tellerebene erfolgt, kann die Bewegung durch einen einzigen Winkel beschrieben werden. Diese spezielle Aufnahmesituation soll den nachfolgenden Rekonstruktionsprozeß vereinfachen. Die Abbildungsgeometrie einer *Aufnahmeanordnung* muß durch die Vermessung von in den Kamerabildern sichtbaren Kalibriermarken bestimmt werden. Wie dieser einmalige Vorgang mit herkömmlichen CCD-Kameras durchgeführt werden kann, ist in dem Kapitel 2 beschrieben.

Eine geeignete Vorverarbeitung der Bilder und die Merkmalsextraktion stellen wichtige Hilfsmittel bei der Bewegungsbestimmung im Bild dar. Daher werden im nachfolgenden Kapitel 3 bewährte *Vorverarbeitungsmethoden* aufgeführt, die sich als besonders praktikabel erwiesen haben, um z.B. eine einfache Bildverbesserung vorzunehmen oder die Informationsgewinnung zu erleichtern. In dem Kapitel 4 sollen die verwendeten Operatoren erläutert werden, mit denen prägnante *Objektmerkmale* wie z.B. Kanten und Eckpunkte aus den aufgenommenen Bildern extrahiert werden können.

Die Berechnung der *Verschiebungsvektoren*, welche die zweidimensionale Bewegung der Objektpunkte im Bild zwischen unterschiedlichen Ansichten beschreiben, wirkt sich entscheidend auf die Qualität der Gestaltsrekonstruktion aus. Natürlich kann ein Objekt mit aufwendigen Markierungen versehen werden und das Korrespondenzproblem von Punkten in den verschiedenen Bildern der Fleißarbeit des Benutzer überlassen bleiben. Mit dieser mühsamen Variante gelangt man auch zu einem Ergebnis, jedoch wurde sie aufgrund des Fernziels komplexe Objekte rekonstruieren zu können in diesem Projekt nicht weiter verfolgt.

Daher werden in dem Kapitel 5 die verschiedenen Ansätze zur automatisierten Verschiebungsvektorbestimmung untersucht. Neben der Nutzung bekannter Techniken zur Berechnung des *Optischen Flusses*

wird die Bestimmung der Verschiebungsvektoren für den verwendeten Spezialfall der *kontrollierten Bewegung* hergeleitet. Das anschließende Kapitel 6 beschäftigt sich dann mit der *Tiefenbestimmung* aus den berechneten Verschiebungsvektoren. Die eingeschränkte Bewegung durch den Drehteller erlaubt sogar die Berechnung des Drehwinkels, d.h. der einzige variable Parameter der Bewegung muß nicht unbedingt vom Benutzer ermittelt werden. Um Objekte vollständig rekonstruieren zu können, müssen anschließend die bestehenden Verschiebungsvektorfelder interpoliert werden. Die Leistung der untersuchten Verfahren soll anhand der durchgeföhrten *Experimente* in Kapitel 7 diskutiert werden. Neben der Generierung von synthetischen Eingabedaten werden auch Ergebnisse mit real aufgenommenen Bildern präsentiert.

Ausschlaggebend für die gewählte *Lösungsstrategie* zu der gestellten Aufgabe war der Einsatz von möglichst wenigen Einschränkungen und Nebenbedingungen. Das Ziel beliebige Objekte ohne Kenntnis ihrer Oberflächenbeschaffenheit (z.B. Textur oder Lambert'sche Oberfläche) zu rekonstruieren sowie die Erfahrungen mit bestehenden Techniken zur Bestimmung des Optischen Flußes führten dazu, zunächst nur für markante Objektmerkmale die Verschiebungsvektoren zu bestimmen.

Somit war der Weg über *merkmalsbasierte Verfahren* eingeschlagen. Es wurden zwar auch bestehende Techniken zur Bestimmung des *Optischen Flußes* getestet, jedoch die Analyse nicht weiter vertieft. Eine Gestaltsrekonstruktion aus *Schattierung* oder einer *Texturanalyse* wurde nicht behandelt, sie ließen sich jedoch ohne große Anstrengungen in das vorgestellte Konzept integrieren. In der Abb. (1.1) ist dieser Zusammenhang graphisch dargestellt.

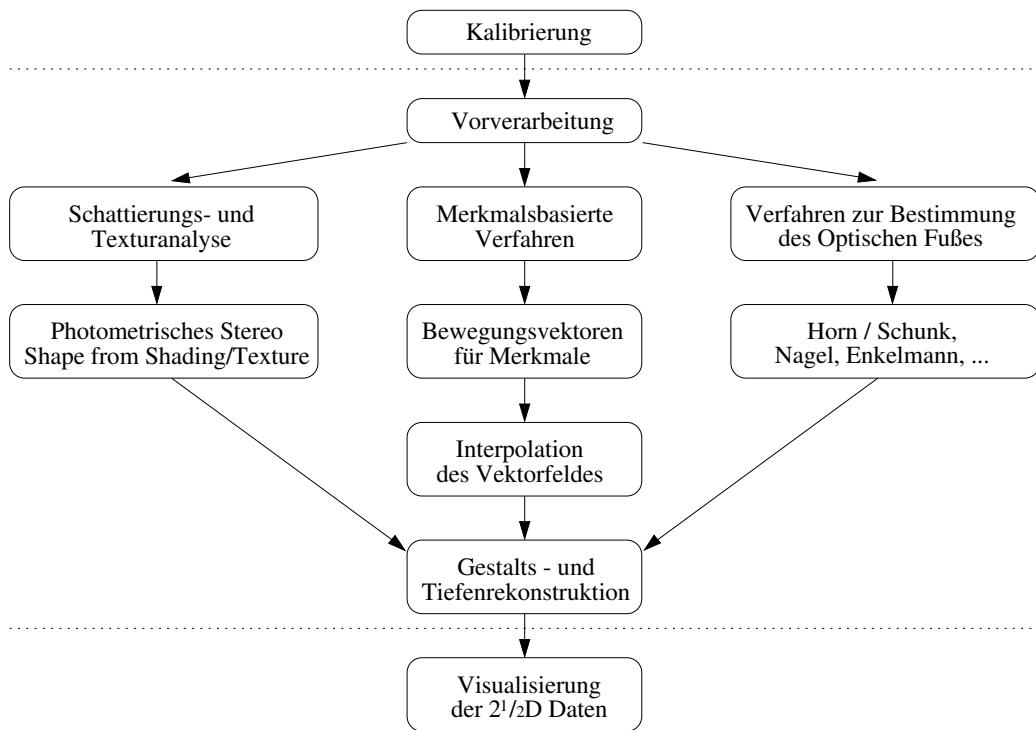


Abbildung 1.1: Die Abbildung zeigt die Einordnung des gewählten Ansatzes und die verschiedenen Prozesse der merkmalsbasierten Gestaltsrekonstruktion im Rahmen bestehender Techniken.

Kapitel 2

Kamera-Kalibrierung

Bei einer Gestaltsanalyse ist es unbedingt notwendig die Bilddurchnahmesituation exakt zu vermessen. Die Position und Ausrichtung der Kamera sind neben der verwendeten Brennweite wichtige Parameter für die angestrebte Rekonstruktion der Gestalt. Eine manuelle Bestimmung dieser Parameter ist aufgrund der benötigten Genauigkeit nur äußerst schwer realisierbar.

Daher soll im diesem Kapitel eine rechnergestützte 3D-Kamerakalibrierung für handelsübliche CCD-Kameras vorgestellt werden. Die untersuchten neueren Verfahren (z.B. Zwei-Ebenen-Methode [GRE88]) sind auf spezielle Probleme zugeschnitten und für unser Projekt weniger geeignet.

Bei der Implementierung wurde daher ein klassisches Verfahren von *Tsai* [TSA86] gewählt. Für diese Technik können wir eine schnelle iterative Lösung zur Entzerrung angeben und eine verbesserte Funktion für die nichtlineare Optimierung vorstellen. Im folgenden Abschnitt soll eine Übersicht der verschiedenen Kalibrierungs-Varianten gegeben werden, um eine Einordnung und Beurteilung der verwendeten Technik zu ermöglichen.

2.1 Techniken der Kamera-Kalibrierung

Die bisher veröffentlichten Kalibrierungs-Techniken können in verschiedene Gruppen eingeteilt werden. Für die Bewertung einer Technik sind die Kriterien der Autonomie, Präzision, Effizienz, Vielseitigkeit und Verwendung handelsüblicher Kameras und Linsen zu untersuchen. Die Stärken und Schwächen einer jeden Klasse beschreibt *Tsai* [TSA86] folgendermaßen:

Klasse 1:

Techniken mit nichtlinearer Optimierung in vollem Umfang

Vorteile :

- Sie erlauben eine einfache Anpassung an ein beliebig genaues oder komplexes Bildaufnahmemodell.

Nachteile :

- Der Beginn der nichtlinearen Suche erfordert eine gute initiale Schätzung (Autonomie-Kriterium).
- Sie benötigen eine rechenintensive nichtlineare Suche in vollem Umfang. (Effizienz-Kriterium).

Referenzen :

- *Faig's* klassischer Ansatz [FAI75] beruht auf einem komplizierten Abbildungsmodell und ist sehr rechenintensiv. Durch die Verwendung der großen Anzahl von 17 Unbekannten ist die Genauigkeit der Kalibrierung mit 1:2000 hervorragend.
- *Abdel-Aziz und Karara* [AZI71, AZI74] entwickelten die Direct Linear Transformation (DLT), damit nur lineare Gleichungen gelöst werden müssen und erreichten eine Präzision von 1:2000 in 80 Prozent des Sichtbereiches. Sobald aber eine Linsenverzerrung angenommen wird, so ist auch hier eine nichtlineare Suche erforderlich.
- *Sobel* [SOB74] stellte eine Methode zur Kamerakalibrierung durch Lösen von großen nichtlinearen Gleichungssystemen vor, wobei 18 Parameter optimiert werden mußten. Eine Linsenverzerrung wurde nicht modelliert und es wurde auch keine Aussage über die Genauigkeit getroffen. Das System ist von der Initialisierung der Parameter durch den Benutzer abhängig.
- *Gennery* [GEN79] beschreibt eine iterative Methode, um die Kameraparameter zu finden. Durch Fehlerminimierung eines Epipolar-Constraints wurden keine 3D-Koordinaten der Kalibrierungspunkte benötigt, was sich jedoch als recht fehlerträchtig erwiesen hat.

Klasse 2 :

Techniken, die eine Perspektiv-Transformationsmatrix
anhand linearer Gleichungen vorab berechnen

Vorteile :

- Eine nichtlineare Optimierung ist nicht erforderlich.

Nachteile :

- Eine Linsenverzerrung kann nicht berücksichtigt werden.
- Die Anzahl der verwendeten Variablen ist viel größer als die Anzahl der Unbekannten, d.h. die zu bestimmenden Variablen sind nicht immer linear unabhängig. Die Nachteile dieser redundanten Parametrisierung liegen bei verrauschten Situationen in der mangelnden Genauigkeit durch die fehlerbehaftete Kombination dieser Parameter.

Referenzen :

- *Sutherland* [SUT74] formulierte für die Computergrafik sehr genau die Berechnung der perspektivischen Transformationsmatrix aus einer Anzahl von Punkten.
- *Yakimovsky und Cunningham* [YAK78] präsentierten eine Kalibrierungstechnik für ein kleines Sichtfeld und große Objektentfernungen, bei der durch Kombination von Parametern zu einzelnen Variablen das Problem als lineares Gleichungssystem formuliert werden kann. Sie erreichten eine Genauigkeit von 1:400, aber entgegen ihrer Annahme sind die Variablen nicht vollständig linear unabhängig.
- *Hall, Tio, McPherson und Sadjadi* [HAL82] benutzten eine durchweg lineare Technik für die Methode der kleinsten Quadrate, um die Elemente der perspektivischen Transformationsmatrix zu bestimmen und gekrümmte Oberflächen zu vermessen.
- *Ganapathy* [GAN84] und *Strat* [STR84] leiteten eine nicht-iterative Technik zur Berechnung der Kamera-Parameter bei gegebener perspektivischer Transformationsmatrix ab.

Klasse 3 :
Die Zwei-Ebenen Methode

Vorteile :

- Es brauchen nur lineare Gleichungen gelöst zu werden.
- Es wird kein explizites Kameramodell verwendet.
- Die Genauigkeit kann (beliebig) präzisiert werden.

Nachteile :

- Die Anzahl der verwendeten Variablen ist mit 24 (12 für jede Ebene) viel größer als die Anzahl der Unbekannten.
- Die affine Transformation zwischen Kamera- und Weltkoordinatensystem wird als bekannt vorausgesetzt.

Referenzen :

- Die Technik von *Martins, Birk und Kelly* [MAR81] besteht in der Vermessung von Kalibrierungsdaten in verschiedenen Ebenen sowie im Bild. Die Daten von anderen Pixeln werden durch Interpolation gewonnen. *Grattoni* [GRA91] berichtete nach einem Vergleich mit den klassischen Techniken, daß dieses Verfahren vier mal besser sei.
- *Isaguirre, Pu und Summers* [ISA85] erweiterten den Ansatz, um die Kalibrierung als Funktion der Position und Orientierung von der Kamera beschreiben zu können. Sie verwendeten dabei einen auf Kalman-Filterung basierenden iterativen Ansatz.
- *Gremban, Thorpe und Kanade* [GRE88] lösten das Projektionsproblem der Zwei-Ebenen Methode, allerdings durch Wiedereinführung der Zentralprojektion.

Klasse 4 :
Geometrische Techniken

Vorteile :

- Es kann auf die nichtlineare Suche verzichtet werden.

Nachteile :

- Eine Linsenverzerrung kann nicht berücksichtigt werden.
- Die Brennweite wird als bekannt vorausgesetzt.
- Eine Unbestimmtheit des Bildskalierungsfaktors ist nicht erlaubt.

Referenzen :

- *Fischler und Bolles* [FIS81] verwendeten eine geometrische Konstruktion für eine direkte Lösung der Kameraposition und -orientierung. Interne Kameraparameter können nicht berechnet werden und es wurden keine Genauigkeitsangaben gegeben.

Dieser Klassifizierung entsprechend muß das verwendete Verfahren von *Tsai* [TSA86] zu der ersten Kategorie (*Techniken mit nichtlinearer Optimierung*) gezählt werden. Es ermöglicht eine effiziente Berechnung der Kameraposition und -orientierung relativ zum Objektkoordinatensystem sowie der Brennweite, der radialen Linsenverzerrung und der Parameter für die Bildabtastung.

2.2 Das Kameramodell

Zunächst soll das verwendete Kameramodell mit Perspektivprojektion und radialer Linsenverzerrung vorgestellt werden. Dadurch wird deutlich, welche Parameter im einzelnen kalibriert werden müssen. Die zugrundeliegende Kamerageometrie wird in Abb. (2.1) dargestellt und zeigt die verwendeten (links-händigen) Koordinatensysteme.

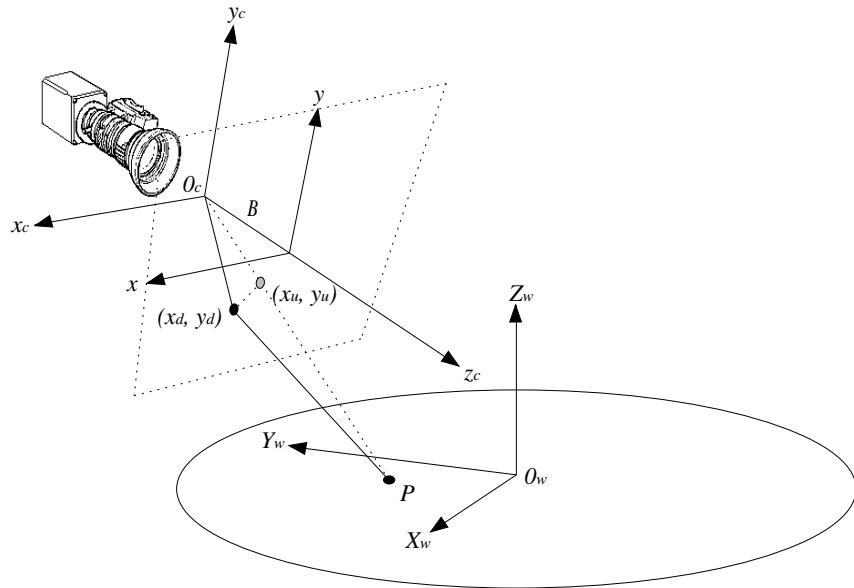


Abbildung 2.1: Kamerageometrie mit Perspektivprojektion und radialem Linsenverzerrung.

Im einzelnen bezeichnen

- (X_w, Y_w, Z_w) die 3D-Koordinaten eines Objektpunktes P im *Weltkoordinatensystem WKS*,
- (x_c, y_c, z_c) die 3D-Koordinaten von P im *Kamerakoordinatensystem KKS*,
- B den Abstand der Bildeckenebene vom Brennpunkt (*Brennweite*),
- (x_u, y_u) die *unverzerrten* Bildkoordinaten von (x_c, y_c, z_c) unter Verwendung eines perfekten Lochkameramodells,
- (x_d, y_d) die *realen* Bildkoordinaten, die sich von (x_u, y_u) durch radiale Linsenverzerrung unterscheiden, sowie (nicht im Bild eingetragen)
- (x_f, y_f) die *geräteabhängige* Position von (x_d, y_d) im digitalisierten Bild.

Die Z-Achse des Kamerakoordinatensystems z_c entspricht der optischen Achse. Bei unserer speziellen Gestaltsrekonstruktion stimmt die Z-Achse des Weltkoordinatensystems Z_w mit der Rotationsachse des Drehtellers überein. Im allgemeinen ist diese Festlegung für die Kalibrierung aber nicht notwendig und die Achse könnte daher auch beliebig gewählt werden.

Von den aufgeführten Komponenten besitzen alle die gleiche Einheit (z.B. mm). Eine Ausnahme bilden die Positionen (x_f, y_f) in dem digitalisierten Bild, welche in (Sub-)Pixeln anzugeben sind. Neben einer

geräteabhängigen Skalierung unterscheiden sie sich von den übrigen Bildkoordinaten hauptsächlich in der Umrechnung von zentrierten Koordinaten in positive Zeilen- bzw. Spaltenpositionen.

Ein Objektpunkt P kann sich zunächst in beliebiger Lage im 3D-Raum befinden. Anhand der Projektion von P sollen die vier Schritte zur Transformation von 3D-Weltkoordinaten (X_w, Y_w, Z_w) in geräteabhängige Bildpositionen (x_f, y_f) erläutert werden.

1. Schritt

Zuerst muß eine affine Transformation vom Weltkoordinatensystem des Objektpunktes (X_w, Y_w, Z_w) in das 3D-Kamerakoordinatensystem (x_c, y_c, z_c) durchgeführt werden. Hierzu sei

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T \quad (2.1)$$

mit

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad \text{und} \quad T = [T_x, T_y, T_z]^T.$$

Die Transformation eines festen Körpers von einem kartesischen Koordinatensystem in ein anderes bleibt eindeutig, sofern die 3D-Rotation um den Ursprung vor der 3D-Translation ausgeführt wird. Die zu kalibrierenden Parameter sind die *Rotationsmatrix* R und der *Translationsvektor* T .

2. Schritt

Es folgt die Transformation des 3D-Kamerakoordinatensystems (x_c, y_c, z_c) in ideale, unverzerrte Bildkoordinaten (x_u, y_u) unter Verwendung der Perspektivprojektion mit dem Lochkameramodell gemäß

$$x_u = \frac{B \cdot x_c}{z_c} \quad \text{und} \quad y_u = \frac{B \cdot y_c}{z_c}. \quad (2.2)$$

Hier ist die *Brennweite* B zu kalibrieren.

3. Schritt

Die Bestimmung der unverzerrten Bildkoordinaten (x_u, y_u) aus den real abgebildeten Bildkoordinaten (x_d, y_d) gibt Tsai [TSA86] folgendermaßen an

$$x_d + D_x = x_u, \quad y_d + D_y = y_u \quad (2.3)$$

mit den Abkürzungen

$$D_x = x_d \cdot (\kappa_1 r^2 + \kappa_2 r^4), \quad D_y = y_d \cdot (\kappa_1 r^2 + \kappa_2 r^4) \quad \text{und} \quad r = \sqrt{x_d^2 + y_d^2}.$$

In Abb. (2.2) wird der Einfluß der betrachteten radialen Verzerrung auf das aufgenommene Bild dargestellt. Die *Koeffizienten der Verzerrung* κ_1 und κ_2 sind hier die zu kalibrierenden Parameter. Ein positiver Wert von κ_1 bzw. κ_2 bedeutet, daß bei der Projektion die Ecken im Bild gestaucht werden und somit anhand einer Streckung entzerrt werden müssen.

Die Modellierung der gewählten Linsenverzerrung ist im *Manual of Photogrammetry* [MAN80] beschrieben und sieht eine unendliche Reihe von Verzerrungskoeffizienten vor. Bei der verwendeten Technik werden

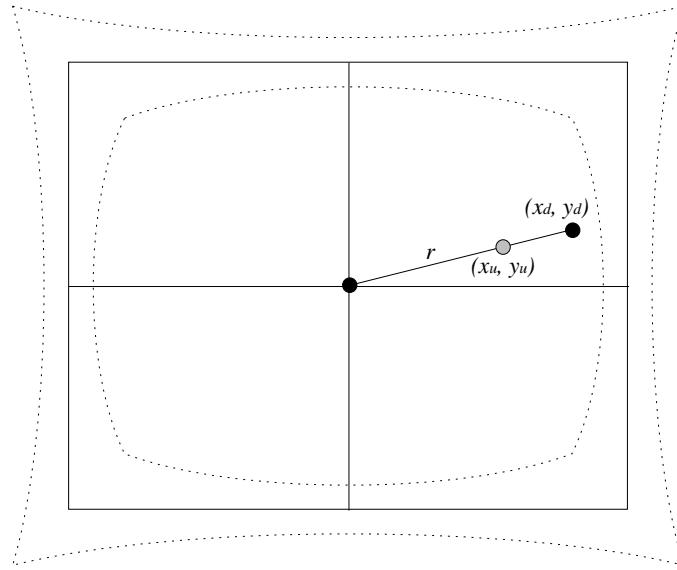


Abbildung 2.2: Deformation des Bildes bei radialer Linsenverzerrung.

aber nur die ersten beiden Koeffizienten betrachtet, da ansonsten numerische Instabilitäten das Ergebnis beeinträchtigen. Die Gleichungen (2.3) bewirken eine *Entzerrung* von Bildpunkten bei bekannten (x_d, y_d) . Für die bisher betrachtete Transformationsrichtung müssen sie nach eben diesen Koordinaten aufgelöst werden, wodurch sich nichtlineare Gleichungen ergeben.

Die Verwendung eines numerischen Verfahrens zur Lösung allgemeiner nichtlinearer Gleichungssysteme ist in der Geschwindigkeit nicht akzeptabel, wenn z.B. ein komplettes Bild transformiert werden muß. Aus diesem Grund wird für die *Verzerrung* von idealen Bildpunkten (x_u, y_u) folgende einfache Approximation eingesetzt:

$$x_{di} = \frac{x_u}{1 + \kappa_1 r_{(i-1)}^2 + \kappa_2 r_{(i-1)}^4}, \quad y_{di} = \frac{y_u}{1 + \kappa_1 r_{(i-1)}^2 + \kappa_2 r_{(i-1)}^4} \quad (2.4)$$

$$\text{mit } r_i = \sqrt{x_{di}^2 + y_{di}^2} \quad \text{für } i \in \{1 \dots n\}.$$

Mit einem Startwert von $r_0 = \sqrt{x_u^2 + y_u^2}$ erhält man durch die Gleichungen (2.4) eine erste Annäherung für die gesuchte Lösung von (x_d, y_d) . Mit den Ergebnissen können fortlaufend verbesserte Radien r_i bestimmt werden. Durch mehrfache Anwendung dieser Berechnungsvorschrift kann die Lösung über n Iterationen approximiert werden. In den meisten Fällen stabilisierte sich das Ergebnis in weniger als 8 Durchläufen.

4. Schritt

Abschließend erfolgt die Transformation der realen Bildkoordinaten (x_d, y_d) in geräteabhängige Bildpositionen (x_f, y_f) . Neben einer Skalierung und der Umrechnung von zentrierten Koordinaten in Zeilen- bzw. Spaltenpositionen wird eine Variable für eventuelle Fehler in der Bildabtastung eingeführt. Es sei

$$x_f = \frac{s_x x_d}{d'_x} + c_x, \quad y_f = \frac{y_d}{d_y} + c_y, \quad (2.5)$$

wobei

- (c_x, c_y) der Ursprung des Bildkoordinatensystems ist,
- $d'_x = d_x \frac{N_{cx}}{N_{fx}}$ der Abstand benachbarter Pixel in einer Bildzeile ist,
- d_x den Abstand benachbarter Sensorelemente in X-Richtung bezeichnet,
- d_y den Abstand benachbarter Sensorelemente in Y-Richtung bezeichnet,
- N_{cx} die Anzahl der Sensorelemente in X-Richtung ist und
- N_{fx} die Anzahl der Pixel in einer Bildzeile ist.

Die Fehlerbetrachtung ist notwendig, da das Hardware-Timing für die Abtastung und Digitalisierung bei CCD-Kameras und Frame-Grabbern nicht immer perfekt ist. Tsai [TSA86] berichtet von 25 pix horizontaler Abweichung durch diese Ungenauigkeiten und schlägt vor, durch eine zusätzliche Variable s_x die entsprechenden Störfaktoren zu akkumulieren. Die zu kalibrierenden Parameter sind somit der *Bildskalierungsfaktor* s_x und der *Bildursprung* (c_x, c_y) . Die Sensordaten sind durch Herstellerangaben zu bestimmen. Der Gesamtprozeß der beschriebenen Koordinatentransformationen ist in Abb. (2.3) noch einmal zusammengefaßt.

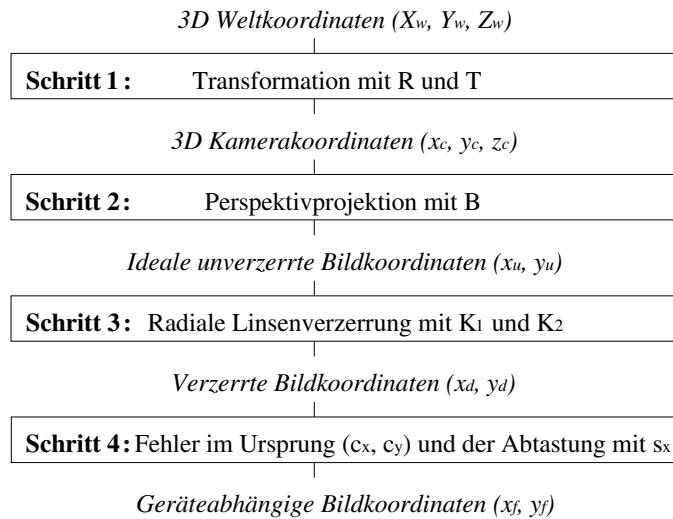


Abbildung 2.3: Abbildungsprozeß des verwendeten Kameramodells.

2.3 Kamera-Kalibrierung anhand coplanarer Punkte

Nach der Definition des Kameramodells und der zu kalibrierenden Parameter soll nun die Bestimmung der Unbekannten erläutert werden. Für die Vermessung der Aufnahmesituation ist ein besonderes *Kalibrierungsobjekt* erforderlich. Die Weltkoordinaten (X_w, Y_w, Z_w) und deren Ursprung können dazu vom Anwender frei gewählt werden.

An dieser Stelle unterscheidet das Verfahren von *Tsai* [TSA86] zwei Situationen, die auch beide implementiert wurden. In einer ersten Variante sei zunächst angenommen, daß alle Kalibrierungspunkte in einer Ebene mit $Z_w = 0$ liegen (*coplanar*). Die zweite Variante (vgl. Abschnitt 2.4) behandelt den allgemeineren Fall mit beliebig positionierbaren Kalibrierungspunkten.

Nach der möglichst exakten Vermessung von mindestens fünf coplanaren Kalibrierungspunkten und deren Projektion im Bild können die internen und externen Kameraparameter berechnet werden. Um dieses umfangreiche Kapitel 2 übersichtlich halten zu können, wird auf diverse Herleitungen und Beweise der Existenz und Eindeutigkeit von Lösungen verzichtet. Stattdessen wird eine Berechnungsvorschrift gegeben, wie die Unbekannten bis zur fertigen Kalibrierung schrittweise bestimmt werden können.

2.3.1 Stufe 1: Berechnung der 3D-Orientierung und XY-Position

Berechnung der realen Bildkoordinaten (x_d, y_d)

Als erstes wird ein Bild aufgenommen und in den Rechner eingelesen. Anschließend wird die Zeilen- und Spaltenposition (x_f, y_f) jedes Kalibrierungspunktes im Bild ermittelt, was durchaus auch subpixelgenau erfolgen kann (siehe Abschnitt 4.1). Aufgrund der positiv definierten Bildpositionen und des linkshändigen Koordinatensystems befindet sich die Position $(0, 0)$ in der linken unteren Ecke.

Die Parameter N_{cx}, N_{fx}, d'_x und d_y können durch Herstellerangaben bestimmt werden. Als Ursprung (c_x, c_y) wird vorerst ein Pixel im Bildmittelpunkt angenommen. Danach werden für alle Kalibrierungspunkte die realen Bildkoordinaten (x_d, y_d) folgendermaßen berechnet:

$$x_d = \frac{(x_f - c_x) \cdot d'_x}{s_x} \quad \text{und} \quad y_d = (y_f - c_y) \cdot d_y. \quad (2.6)$$

Dabei muß für s_x *a-priori Wissen* verwendet werden, da im coplanaren Fall der Bildskalierungsfaktor nicht bestimmt werden kann.

Berechnung von fünf Transformationsparametern

Für jeden Objektpunkt (X_w, Y_w, Z_w) im Weltkoordinatensystem und seinen entsprechenden realen Bildpunkt (x_d, y_d) kann nun folgende lineare Gleichung

$$[y_d X_w, y_d Y_w, y_d, -x_d X_w, -x_d Y_w] \cdot L = x_d$$

mit

$$L = \left[\frac{r_1}{T_y}, \frac{r_2}{T_y}, \frac{T_x}{T_y}, \frac{r_4}{T_y}, \frac{r_5}{T_y} \right]^T \quad \text{für} \quad T_y \neq 0$$

aufgestellt werden, wobei T_x und T_y die Komponenten des Translationsvektors T bezeichnen und r_i die Elemente der Rotationsmatrix R darstellen.

Bei mehr als fünf Kalibrierungspunkten kann ein überbestimmtes Gleichungssystem erzeugt werden und z.B. mit der *Pseudoinversen* nach L aufgelöst werden. Dann ist

$$L = (M^T M)^{-1} M^T \cdot X, \quad (2.7)$$

wobei M und X bei n Kalibrierungspunkten folgendermaßen gebildet werden:

$$M = \begin{bmatrix} y_{d1}X_{w1} & y_{d1}Y_{w1} & y_{d1} & -x_{d1}X_{w1} & -x_{d1}Y_{w1} \\ y_{d2}X_{w2} & y_{d2}Y_{w2} & y_{d2} & -x_{d2}X_{w2} & -x_{d2}Y_{w2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{dn}X_{wn} & y_{dn}Y_{wn} & y_{dn} & -x_{dn}X_{wn} & -x_{dn}Y_{wn} \end{bmatrix} \quad \text{und} \quad X = \begin{bmatrix} x_{d1} \\ x_{d2} \\ \vdots \\ x_{dn} \end{bmatrix}.$$

Berechnung von $|T_y|$

Für die Betragsbestimmung sei

$$C = \begin{bmatrix} r'_1 & r'_2 \\ r'_4 & r'_5 \end{bmatrix} = \begin{bmatrix} \frac{r_1}{T_y} & \frac{r_2}{T_y} \\ \frac{r_4}{T_y} & \frac{r_5}{T_y} \end{bmatrix}$$

eine mit T_y normierte 2×2 -Untermatrix der Rotationsmatrix R . Falls keine ganze Spalte oder Zeile von C zu 0 wird, dann kann

$$T_y^2 = \frac{S_r - \sqrt{S_r^2 - 4 \cdot (r'_1 r'_5 - r'_4 r'_2)^2}}{2 \cdot (r'_1 r'_5 - r'_4 r'_2)^2} \quad \text{mit} \quad S_r = r'^2_1 + r'^2_2 + r'^2_4 + r'^2_5 \quad (2.8)$$

berechnet werden. Ansonsten wird der Betrag von T_y durch

$$T_y^2 = \frac{1}{(r'^2_i + r'^2_j)} \quad (2.9)$$

bestimmt, wobei r'_i und r'_j die Elemente der Zeile oder Spalte sind, die nicht verschwinden.

Bestimmung des Vorzeichens von T_y

Es wird ein beliebiger Objektpunkt P im Weltkoordinatensystem (X_w, Y_w, Z_w) ausgesucht, dessen Bildposition (x_f, y_f) möglichst weit vom Ursprung (c_x, c_y) entfernt liegt. Das Vorzeichen von T_y wird als positiv angenommen. Anschließend werden aus dem Lösungsvektor L die Parameter

$$r_1 = \frac{r_1}{T_y} \cdot T_y, \quad r_2 = \frac{r_2}{T_y} \cdot T_y, \quad r_4 = \frac{r_4}{T_y} \cdot T_y, \quad r_5 = \frac{r_5}{T_y} \cdot T_y, \quad T_x = \frac{T_x}{T_y} \cdot T_y \quad (2.10)$$

extrahiert und mit der Position von P

$$x = r_1 X_w + r_2 Y_w + T_x, \quad y = r_4 X_w + r_5 Y_w + T_y \quad (2.11)$$

verglichen. Falls x und x_d sowie y und y_d des Kalibrierungspunktes die gleichen Vorzeichen haben, so bleibt T_y erhalten, ansonsten ist $\text{sgn}(T_y) = -1$.

Berechnung der 3D-Rotationsmatrix R

Nach der Bestimmung des Vorzeichens von T_y muß die Gleichung (2.10) erneut angewendet werden, um die Parameter r_1, r_2, r_4, r_5 und T_x zu aktualisieren. Die Rotationsmatrix R berechnet sich dann nach der Formel

$$R = \begin{bmatrix} r_1 & r_2 & \sqrt{1 - r_1^2 - r_2^2} \\ r_4 & r_5 & s\sqrt{1 - r_4^2 - r_5^2} \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (2.12)$$

mit $s = -\text{sgn}(r_1r_4 + r_2r_5)$. Die noch fehlenden Komponenten r_7, r_8, r_9 werden aus dem Kreuzprodukt der ersten beiden Zeilen und unter Verwendung der Orthonormalitätseigenschaft $r_7^2 + r_8^2 + r_9^2 = 1$ bestimmt. Ist bei der anschließenden Berechnung der Brennweite B (siehe Abschnitt 2.3.2) das Vorzeichen negativ, dann gilt

$$R = \begin{bmatrix} r_1 & r_2 & -\sqrt{1 - r_1^2 - r_2^2} \\ r_4 & r_5 & -s\sqrt{1 - r_4^2 - r_5^2} \\ -r_7 & -r_8 & r_9 \end{bmatrix}. \quad (2.13)$$

2.3.2 Stufe 2: Berechnung der internen Kameraparameter

Bestimmung einer Approximation von B und T_z

Für jeden Kalibrierungspunkt kann die lineare Gleichung

$$[y \quad -y_d] \begin{bmatrix} B \\ T_z \end{bmatrix} = wy_d$$

mit $y = r_4X_w + r_5Y_w + r_6 \cdot 0 + T_y$ und $w = r_7X_w + r_8Y_w + r_9 \cdot 0$ aufgestellt werden. Bei mehr als 2 Kalibrierungspunkten kann ein überbestimmtes Gleichungssystem erzeugt werden und z.B. mit der *Pseudoinversen* nach den Unbekannten B und T_z aufgelöst werden. Dann ist

$$\begin{bmatrix} B \\ T_z \end{bmatrix} = (M^T M)^{-1} M^T \cdot X, \quad (2.14)$$

wobei M und X bei n Kalibrierungspunkten folgendermaßen gebildet werden:

$$M = \begin{bmatrix} y_1 & -y_{d1} \\ y_2 & -y_{d2} \\ \vdots & \vdots \\ y_n & -y_{dn} \end{bmatrix} \quad \text{und} \quad X = \begin{bmatrix} w_1y_{d1} \\ w_2y_{d2} \\ \vdots \\ w_ny_{dn} \end{bmatrix}.$$

Berechnung der exakten Lösung von B , T_z , κ_1 und κ_2

Die genaue Berechnung der Brennweite B , der Tiefe T_z und der Verzerrungskoeffizienten κ_1 und κ_2 erfolgt unter Verwendung eines Standard-Optimierungsverfahrens mit den bereits ermittelten Approximationen als Startwerte. Für die radialen Verzerrungskoeffizienten wird 0 als initialer Wert angenommen.

Wir haben uns bei der Implementation für die *Gradientenmethode* entschieden. Zur Bestimmung der Parameter können dann z.B. die Gleichungen

$$x_{u1} = B \cdot \frac{r_4 X_w + r_5 Y_w + r_6 Z_w + T_x}{r_7 X_w + r_8 Y_w + r_9 Z_w + T_z}, \quad y_{u1} = B \cdot \frac{r_4 X_w + r_5 Y_w + r_6 Z_w + T_y}{r_7 X_w + r_8 Y_w + r_9 Z_w + T_z} \quad (2.15)$$

betrachtet werden. Anschaulich beschreiben diese Gleichungen die Transformation des Weltpunktes (X_w, Y_w, Z_w) in das Kamerakoordinatensystem mit anschließender Perspektivprojektion. Der entstehende ideale Bildpunkt kann aber auch durch radiale Entzerrung des real abgebildeten Bildpunktes (x_d, y_d) erzeugt werden. Dazu wird

$$x_{u2} = x_d \cdot (1 + \kappa_1 r^2 + \kappa_2 r^4), \quad y_{u2} = y_d \cdot (1 + \kappa_1 r^2 + \kappa_2 r^4) \quad \text{mit} \quad r = \sqrt{x_d^2 + y_d^2} \quad (2.16)$$

berechnet. Die *Fehlerfunktion*

$$E = \sqrt{(x_{u2} - x_{u1})^2 + (y_{u2} - y_{u1})^2} \quad (2.17)$$

für die Suche im 4-dimensionalen Vektorraum ergibt sich dann aus dem euklidischen Abstand des auf unterschiedliche Weise berechneten idealen Bildpunktes. Der *Gradient* von E

$$\nabla E = \left[\frac{\partial E}{\partial \kappa_1}, \frac{\partial E}{\partial \kappa_2}, \frac{\partial E}{\partial B}, \frac{\partial E}{\partial T_z} \right]^T \quad (2.18)$$

berechnet sich aus den partiellen Ableitungen

$$\begin{aligned} \frac{\partial E}{\partial \kappa_1} &= \frac{r^2 \cdot ((x_{u2} - x_{u1}) \cdot x_d + (y_{u2} - y_{u1}) \cdot y_d)}{E}, \\ \frac{\partial E}{\partial \kappa_2} &= \frac{r^4 \cdot ((x_{u2} - x_{u1}) \cdot x_d + (y_{u2} - y_{u1}) \cdot y_d)}{E}, \\ \frac{\partial E}{\partial B} &= -\frac{((x_{u2} - x_{u1}) \cdot (r_1 X_w + r_2 Y_w + r_3 Z_w + T_x) + (y_{u2} - y_{u1}) \cdot (r_7 X_w + r_8 Y_w + r_9 Z_w + T_y))}{E \cdot (r_7 X_w + r_8 Y_w + r_9 Z_w + T_z)}, \\ \frac{\partial E}{\partial T_z} &= \frac{B \cdot ((x_{u2} - x_{u1}) \cdot (r_1 X_w + r_2 Y_w + r_3 Z_w + T_x) + (y_{u2} - y_{u1}) \cdot (r_7 X_w + r_8 Y_w + r_9 Z_w + T_y))}{E \cdot (r_7 X_w + r_8 Y_w + r_9 Z_w + T_z)^2} \end{aligned}$$

und gibt die Richtung des steilsten Anstiegs der Fehlerfunktion E an. Da ein *Fehlerminimum* benötigt wird, muß entgegen der Gradientenrichtung iterativ das Optimum gesucht werden.

Der Bildmittelpunkt (c_x, c_y) kann ebenfalls optimiert werden, wenn in der Fehlerfunktion E der reale Bildpunkt folgendermaßen ersetzt

$$x_d = \frac{d'_x \cdot (x_f - c_x)}{s_x}, \quad y_d = d'_y \cdot (y_f - c_y)$$

und der Gradient ∇E entsprechend angepaßt wird. Tsai [TSA86] beschreibt allerdings, daß dies nicht zwingend erforderlich sei. In seinen Experimenten hätte eine Verschiebung des Ursprungs um 10 pix kaum Einfluß auf die Genauigkeit der Kalibrierung gehabt.

2.4 Kamera-Kalibrierung mit nicht-coplanaren Punkten

Die allgemeinere Variante ermöglicht die Verwendung beliebig positionierter Kalibrierungspunkte und bestimmt zusätzlich den Bildskalierungsfaktor s_x . Allerdings müssen dazu mindestens 7 Kalibrierungspunkte ausgemessen werden.

2.4.1 Stufe 1: Berechnung der 3D-Orientierung, XY-Position und Skalierung

Berechnung der realen Bildkoordinaten (x_d, y_d)

Die Berechnung der Koordinaten (x_d, y_d) erfolgt genau wie im coplanaren Fall, mit der Ausnahme, daß der Bildskalierungsfaktor s_x vorerst auf 1 gesetzt wird.

Berechnung von sieben Transformationsparametern

Das folgende Verfahren entspricht dem coplanaren Fall, nur daß Z_w nicht auf 0 gesetzt wird. Für jeden Objektpunkt im Weltkoordinatensystem (X_w, Y_w, Z_w) und seinen entsprechenden Bildpunkt kann folgende lineare Gleichung

$$[y_d X_w, y_d Y_w, y_d Z_w, y_d, -x_d X_w, -x_d Y_w - x_d Z_w] \cdot L = x_d$$

mit

$$L = \left[\frac{r_1 s_x}{T_y}, \frac{r_2 s_x}{T_y}, \frac{r_3 s_x}{T_y}, \frac{T_x s_x}{T_y}, \frac{r_4}{T_y}, \frac{r_5}{T_y}, \frac{r_6}{T_y} \right]^T \quad \text{für } T_y \neq 0$$

aufgestellt werden. Bei mehr als sieben Punkten kann ein überbestimmtes Gleichungssystem erzeugt werden und z.B mit der *Pseudoinversen* nach L aufgelöst werden. Dann ist

$$L = (M^T M)^{-1} M^T \cdot X, \quad (2.19)$$

wobei M und X bei n Kalibrierungspunkten folgendermaßen zusammengesetzt sind:

$$M = \begin{bmatrix} y_{d1} X_{w1} & y_{d1} Y_{w1} & y_{d1} Z_{w1} & y_{d1} & -x_{d1} X_{w1} & -x_{d1} Y_{w1} & -x_{d1} Z_{w1} \\ y_{d2} X_{w2} & y_{d2} Y_{w2} & y_{d2} Z_{w2} & y_{d2} & -x_{d2} X_{w2} & -x_{d2} Y_{w2} & -x_{d2} Z_{w2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{dn} X_{wn} & y_{dn} Y_{wn} & y_{dn} Z_{wn} & y_{dn} & -x_{dn} X_{wn} & -x_{dn} Y_{wn} & -x_{dn} Z_{wn} \end{bmatrix}, X = \begin{bmatrix} x_{d1} \\ x_{d2} \\ \vdots \\ x_{dn} \end{bmatrix}.$$

Berechnung von $|T_y|$

Zunächst sollen folgende Abkürzungen

$$\begin{aligned} a_1 &= \frac{s_x r_1}{T_y}, & a_2 &= \frac{s_x r_2}{T_y}, & a_3 &= \frac{s_x r_3}{T_y}, & a_4 &= \frac{s_x T_x}{T_y}, \\ a_5 &= \frac{r_4}{T_y}, & a_6 &= \frac{r_5}{T_y}, & a_7 &= \frac{r_6}{T_y} \end{aligned} \quad (2.20)$$

für die Komponenten des Lösungsvektors L eingeführt werden.

Der Betrag von T_y kann dann durch

$$|T_y| = \frac{1}{\sqrt{a_5^2 + a_6^2 + a_7^2}} \quad (2.21)$$

berechnet werden.

Bestimmung von s_x

Der Bildskalierungsfaktor s_x wird anhand der Gleichung

$$s_x = \sqrt{a_1^2 + a_2^2 + a_3^2} \cdot |T_y| \quad (2.22)$$

ermittelt.

Berechnung der 3D-Rotationsmatrix R

Anschließend werden aus dem Lösungsvektor L die gesuchten Parameter

$$\begin{aligned} r_1 &= a_1 \cdot \frac{T_y}{s_x}, & r_2 &= a_2 \cdot \frac{T_y}{s_x}, & r_3 &= a_3 \cdot \frac{T_y}{s_x}, & T_x &= a_4 \cdot \frac{T_y}{s_x}, \\ r_4 &= a_5 \cdot T_y, & r_5 &= a_6 \cdot T_y, & r_6 &= a_7 \cdot T_y. \end{aligned} \quad (2.23)$$

extrahiert. Die fehlenden Komponenten r_7, r_8, r_9 werden wiederum durch das Kreuzprodukt der ersten beiden Zeilen und unter Verwendung der Orthonormalitätseigenschaft $r_7^2 + r_8^2 + r_9^2 = 1$ bestimmt.

Im Originaltext von *Tsai* [TSA86] hat sich anscheinend ein kleiner Fehler eingeschlichen. In der Gleichung (2.23) werden anstelle der ersten vier alle sieben Komponenten durch s_x dividiert. Es lässt sich aber leicht verifizieren, daß für einen Bildskalierungsfaktor $s_x \neq 1$ die Orthonormalitätseigenschaft der zweiten Zeile verletzt, bzw. die Determinante der Rotationsmatrix R ungleich 1 wird. Dieser kleine Fehler soll allerdings in einer neueren Veröffentlichung [TSA87], die uns nicht zur Verfügung stand, erkannt und beseitigt worden sein.

2.4.2 Stufe 2: Berechnung der internen Kameraparameter

Das Verfahren ist identisch mit dem coplanaren Fall, nur daß auch hier Z_w nicht auf 0 gesetzt werden darf.

2.5 Experimentelle Ergebnisse

Ohne größeren technischen Aufwand ist es nicht einfach, die Genauigkeit der Kalibrierung bei einer realen Aufnahmesituation zu bestimmen. Die Weltkoordinaten können meistens noch millimetergenau vermessen werden, doch quantitative Aussagen über die Kameraparameter sind kaum zu verifizieren. So ist die Position des Brennpunktes und dessen Abstand vom Sensor von außen nur zu schätzen. Aus diesem Grund soll zunächst die Analyse der Kalibrierung mit synthetischen Daten durchgeführt werden.

Im folgenden wird die Kalibrierung mit *nicht-coplanaren Punkten* beschrieben, da dieses Verfahren praktikabler ist. Es werden zwar zwei Kalibrierungspunkte mehr benötigt, aber in der Positionierung ist man nicht eingeschränkt und beim Skalierungsfaktor ist man nicht auf ein Vorwissen angewiesen. Der Spezialfall mit coplanaren Punkten zeigt bei einer guten Schätzung von s_x gleichwertige Resultate, so daß man sich auf die Auswertung eines Verfahrens beschränken kann.

2.5.1 Generierung synthetischer Daten

Als *Referenzmodell* wird ein ursprungszentrierter Würfel mit der Kantenlänge von 90cm und insgesamt 48 Punkten in 3 Ebenen generiert. Die Verzerrungskoeffizienten κ_1 und κ_2 betragen jeweils $-1.0 \cdot 10^{-5}$. In Abb. (2.4) sind einige Variationen des Referenzmodells dargestellt.

Durch einen Zufallszahlen-Generator werden die Weltkoordinaten (X_w, Y_w, Z_w) der Kalibrierungspunkte verrauscht und bezüglich einer vorgegebenen Position und Orientierung transformiert. Anschließend werden die gestörten Koordinaten in das Bild projiziert (x_f, y_f) und ebenfalls mit einem Rauschen unterlegt. Für das Referenzmodell wurde ein *normalverteiltes Rauschen* mit der Standardabweichung $\sigma = 0.1$ verwendet, d.h die mittlere quadratische Abweichung beträgt bei den Weltkoordinaten 0.1mm und mindestens $\frac{1}{10}$ Pixel im Bild. Die generierte Störung entspricht somit 10 Prozent des in Tabelle (2.3) dargestellten Fehlers.

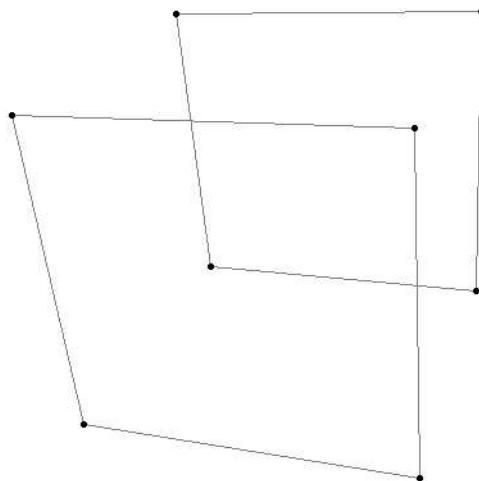
Die synthetisch erzeugten Kalibrierungspunkte im Weltkoordinatensystem und deren subpixelgenaue Bildkoordinaten werden gespeichert und dienen als Eingabe für die Kalibrierung. Bei der Durchführung der Tests wurden die *Herstellerangaben* (vgl. Tabelle 2.1) aus der technischen Beschreibung der SONY CCD Kamera [SON89] übernommen.

Herstellerangaben		
Parameter	horizontal	vertikal
Sensorgröße	8.8 mm	6.6 mm
Sensorpunkte	786 sels	581 sels
Bildpunkte	512 pix	512 pix
Bildmittelpunkt	256 pix	256 pix

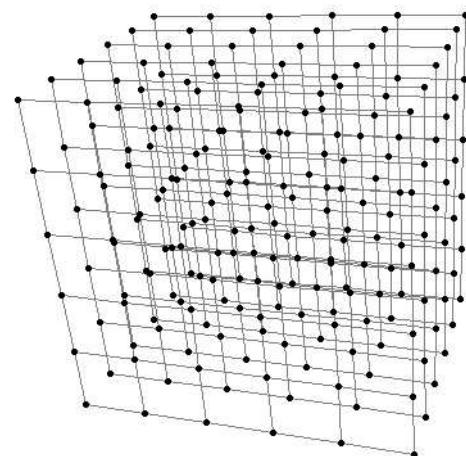
Tabelle 2.1: Herstellerangaben der verwendeten Kamera.

2.5.2 Durchführung einer Kalibrierung

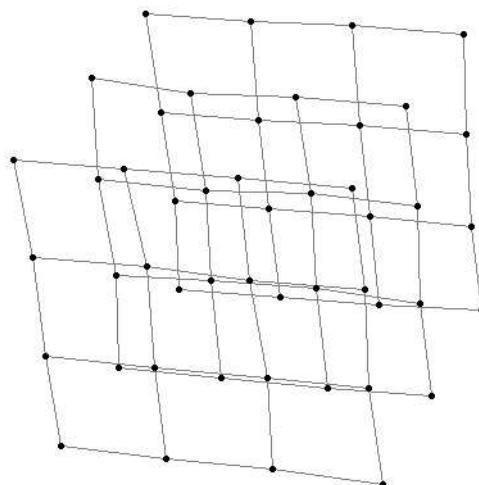
In der Tabelle (2.2) sind die Ergebnisse von zwei Kalibrierungen mit synthetischen Daten aufgeführt. Der *Datensatz I* beschreibt die Kalibrierung des verwendeten Referenzmodells mit einem 8mm Weitwinkelobjektiv aus ca. 2m Entfernung. Diese Werte bilden die Grundlage für die in Abschnitt (2.5.3) durchgeführte Fehlerbetrachtung.



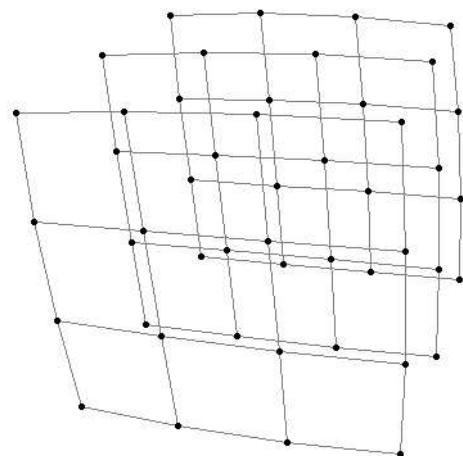
a.) Zwei Ebenen mit jeweils vier Punkten.



b.) Sechs Ebenen mit jeweils 36 Punkten.



c.) Teleobjektiv und starkes Rauschen.



d.) Starke radiale Linsenverzerrung.

Abbildung 2.4: Beispiele der synthetischen Generierung von Kalibrierungspunkten.

Externe Kameraparameter				
Parameter	Originaldaten I	Kalibrierung I	Originaldaten II	Kalibrierung II
Translation	T_x 80.0 mm	80.000 mm	80.0 mm	79.998 mm
	T_y 100.0 mm	100.011 mm	100.0 mm	100.008 mm
	T_z 2000.0 mm	1998.276 mm	8000.0 mm	7999.535 mm
Rotation R	Yaw 60.0 grd	59.994 grd	60.0 grd	59.994 grd
	$Pitch$ 20.0 grd	20.003 grd	20.0 grd	20.002 grd
	$Roll$ -5.0 grd	-5.003 grd	-5.0 grd	-5.003 grd

Interne Kameraparameter				
Parameter	Originaldaten I	Kalibrierung I	Originaldaten II	Kalibrierung II
Brennweite	B 8.0 mm	7.998 mm	32.0 mm	32.016 mm
Skalierungsfaktor	s_x 1.020	1.020	1.020	1.020
Verzerrungskoeffizienten	κ_1 $-1.0 \cdot 10^{-5} \frac{1}{mm^2}$	$-1.3 \cdot 10^{-7} \frac{1}{mm^2}$	$-1.0 \cdot 10^{-5} \frac{1}{mm^2}$	$-1.5 \cdot 10^{-7} \frac{1}{mm^2}$
	κ_2 $-1.0 \cdot 10^{-5} \frac{1}{mm^2}$	$-3.1 \cdot 10^{-6} \frac{1}{mm^2}$	$-1.0 \cdot 10^{-5} \frac{1}{mm^2}$	$-2.7 \cdot 10^{-6} \frac{1}{mm^2}$

Tabelle 2.2: Ergebnis der Kamera-Kalibrierung mit synthetischen Testdaten.

Bei der Änderung der Brennweite mußte die Distanz zum Objekt angepaßt werden, um das Bild gleichmäßig auszufüllen und die Versuchsbedingungen anzugeleichen. Der *Datensatz II* zeigt die Ergebnisse dieser Kalibrierung mit einem Teleobjektiv von 32mm aus ca. 8m Entfernung. Zum besseren Vergleich der affinen Transformation zwischen Welt- und Kamerakoordinatensystem ist es sinnvoll, die *Rotationsmatrix* R als Funktion der *Eulerwinkel* $Yaw \theta$, $Pitch \phi$ und $Roll \psi$ auszudrücken. Die Rotationsmatrix

$$R = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \cos \theta \sin \phi \\ \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi & \cos \theta \cos \phi \end{bmatrix}$$

ist durch die drei Winkel eindeutig bestimmt und kann anhand folgender Berechnung

$$\begin{aligned} \psi &= \arctan \left(\frac{r_4}{r_1} \right), \\ \phi &= \arctan \left(-\frac{r_7}{r_1 \cos \psi + r_4 \sin \psi} \right), \\ \theta &= \arctan \left(\frac{r_3 \sin \psi - r_6 \cos \psi}{r_5 \cos \psi - r_2 \sin \psi} \right) \end{aligned}$$

in eine gleichwertige Repräsentation mit nur drei Parametern umgeformt werden.

2.5.3 Fehlerbetrachtung

Als *Fehler* kann der euklidische Abstand zwischen der berechneten und der erwarteten Position eines Kalibrierungspunktes definiert werden. Bei der Rückprojektion wird die ideale Tiefe Z_w verwendet und die Abweichung in der XY -Ebene betrachtet. Für die Darstellung des Fehlers in Abhängigkeit von unterschiedlichen Parametern werden folgende Abkürzungen eingeführt:

- mp Durchschnittlicher Fehler der Projektion (*Mittelwert*).
- mb Durchschnittlicher Fehler der Rückprojektion.
- sp Mittlere quadratische Abweichung der Projektion (*Standardabweichung*).
- sb Mittlere quadratische Abweichung der Rückprojektion.

Bei der Interpretation der Diagramme ist folgendes zu beachten. Bezuglich der Transformationsrichtung ist der Ordinate die Einheit *pix* bzw. *mm* zuzuordnen. Mit Ausnahme des einen variablen Parameters auf der Abszisse entsprechen alle übrigen Werte dem oben beschriebenen Referenzmodell. Aufgrund der annähernden Symmetrie wurde bei den Verzerrungskoeffizienten nur der positive Wertebereich dargestellt.

Die Abbildung (2.5) zeigt, daß prinzipiell mit der *Anzahl der Kalibrierungspunkte* auch die Genauigkeit steigt. Aber bei mehr als 20 Punkten pro Ebene ist eine Verbesserung kaum noch festzustellen. Ein ähnliches Verhalten kann man in Abb. (2.6) betrachten, wo der Fehler mit zunehmender *Anzahl der Ebenen* sinkt. Auch hier ist eine Stagnation bei mehr als 4 parallelen Kalibrierungsebenen festzustellen. Der *generierte Fehler* einer normalverteilten Störung der Kalibrierungspunkte mit einer Standardabweichung σ von jeweils 1.0 ist in der Tabelle (2.3) aufgeführt.

Störung der Kalibrierungspunkte		
Durchschnittlicher Fehler der	Weltpunkte	1.650 mm
	Bildpunkte	1.354 pix
Maximale Abweichung der	Weltpunkte	4.364 mm
	Bildpunkte	3.365 pix

Tabelle 2.3: Rauschen bei der Generierung von Kalibrierungspunkten mit der Standardabweichung $\sigma = 1$.

Nach der Kalibrierung hat sich der Fehler verringert mit einer resultierenden mittleren quadratischen Abweichung von 0.5 *pix* im Bild bzw. 0.95 *mm* bei der Rückprojektion. Dieser Sachverhalt ist in der Abbildung (2.7) dargestellt. Die Reduzierung der Abweichung ist durch die quadratische Fehlerminimierung der Pseudoinversen zu erklären.

Entgegen anfänglicher Vermutungen zeigt die Abb. (2.8), daß die Kalibrierung von der *Brennweite* nahezu unabhängig ist, d.h. die Verwendung von Teleobjektiv oder Weitwinkel spielt anscheinend keine wesentliche Rolle, wenn der Abstand zum Objekt entsprechend angepaßt wird. Die *Verzerrungskoeffizienten* bewegen sich in einem sehr kleinen Wertebereich. Der resultierende Fehler in Abhängigkeit von der radialen Verzerrung ist in Abb. (2.9) dargestellt. Während der Betrag der Verzerrung bei der verwendeten SONY Kamera kaum wahrnehmbar im Bereich von 10^{-5} liegt, zeigt das Kalibrierungsverfahren erst ab 10^{-3} sichtbare Schwächen. Bei der Annahme einer solch starken Verzerrung wird das Bild allerdings so extrem zu einem Kreis oder *X* deformiert, wie es bei keiner realen Kamera der Fall sein dürfte.

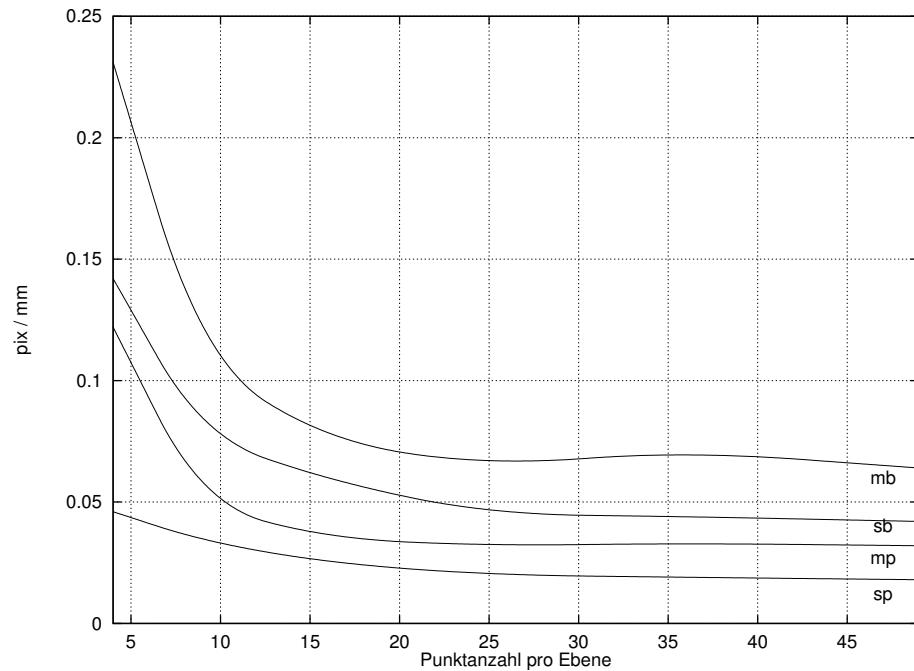


Abbildung 2.5: Fehler der Kalibrierung bei variabler Punktzahl pro Ebene.

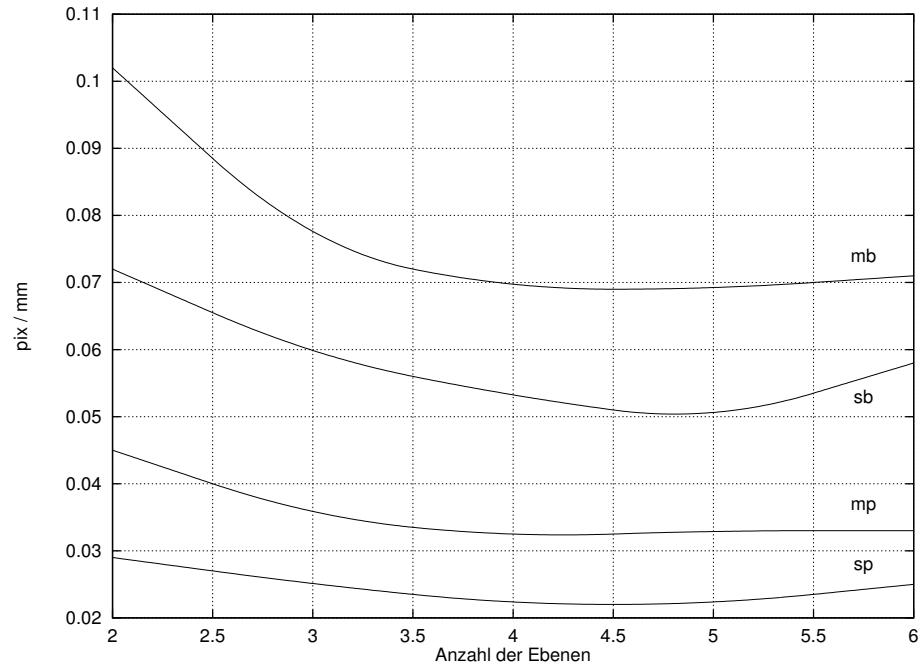


Abbildung 2.6: Kalibrierungsfehler bei unterschiedlich vielen Ebenen.

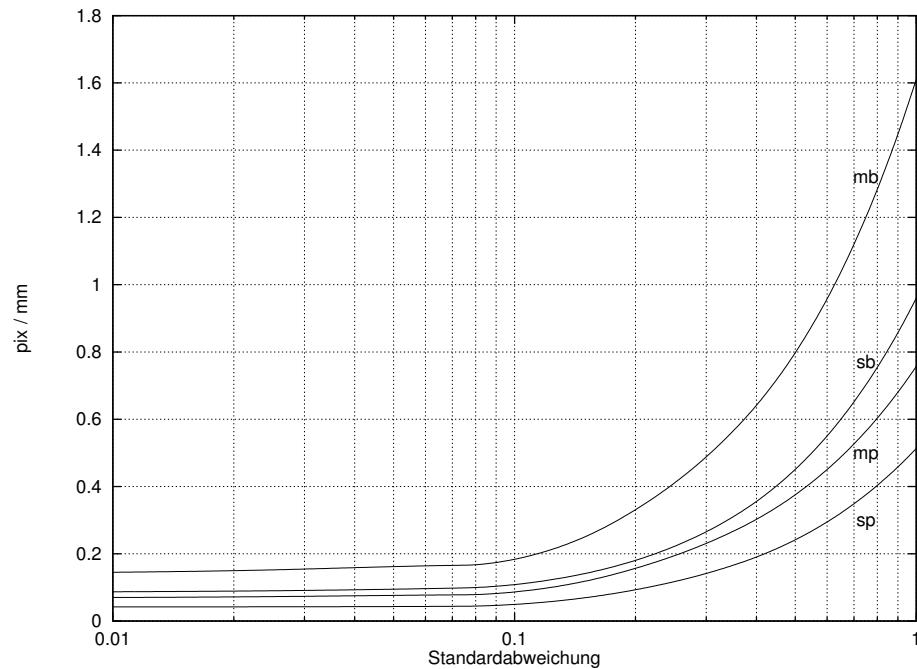


Abbildung 2.7: Fehler durch Verrauschen der Kalibrierungsdaten.

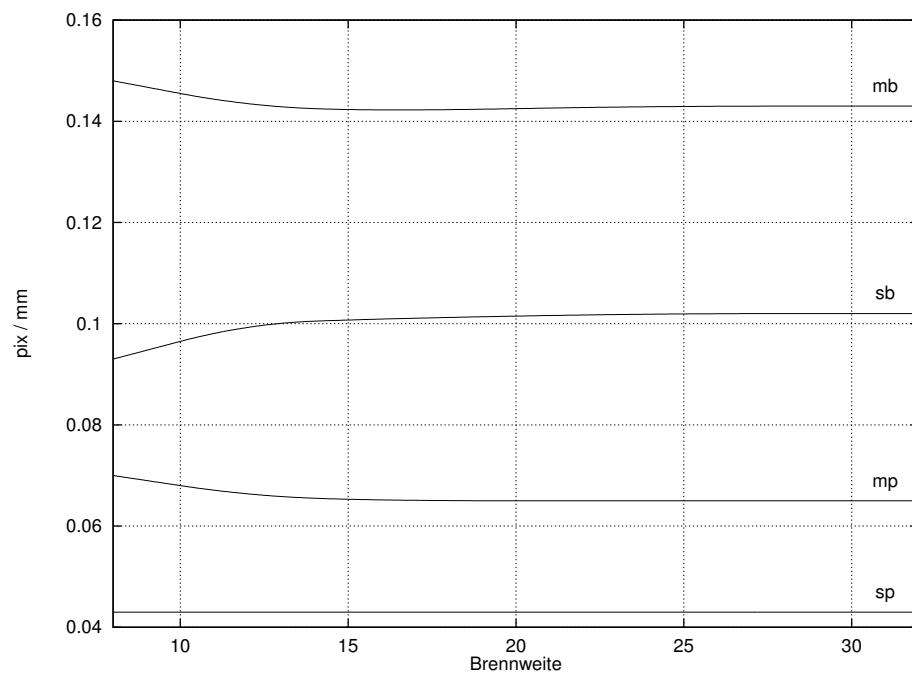


Abbildung 2.8: Kalibrierungsfehler bei unterschiedlichen Brennweiten.

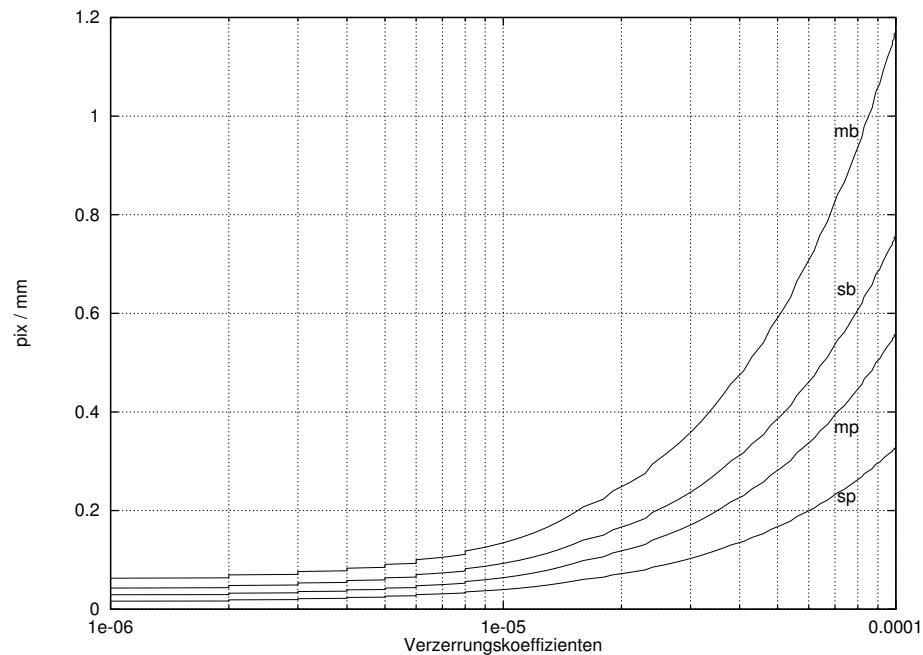
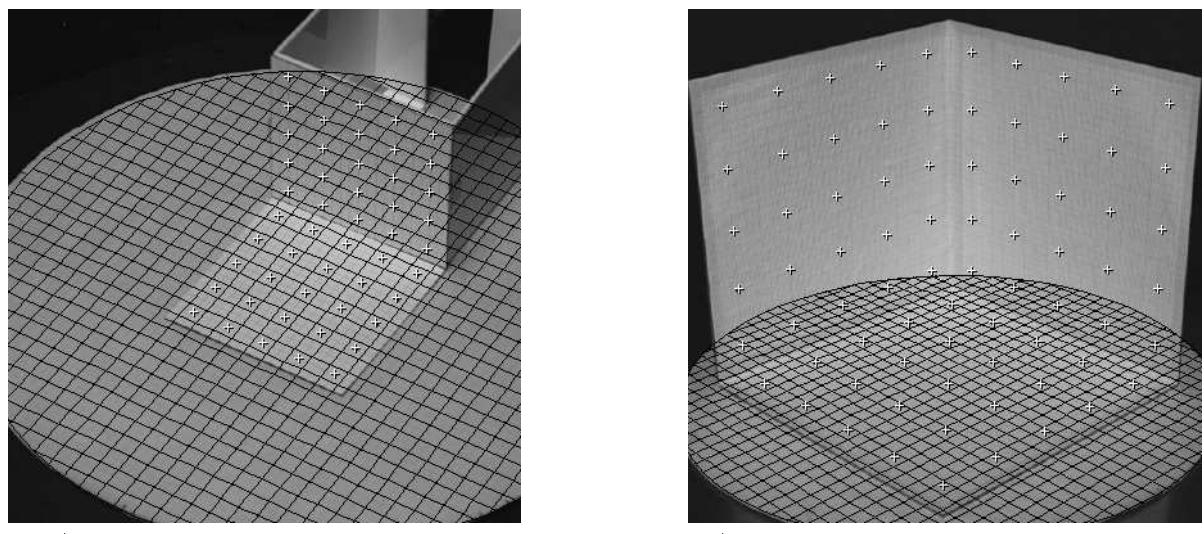


Abbildung 2.9: Fehler in Abhängigkeit der radialen Verzerrung.



a.) Szene III mit zwei kleinen Ebenen.

b.) Szene IV mit besserer Bildausnutzung.

Abbildung 2.10: Beispiele realer Kalibrierungsbilder mit eingeblendetem Drehteller-Raster.

Externe Kameraparameter		
Parameter	Kalibrierung III	Kalibrierung IV
Translation T_x	19.787 mm	1.491 mm
	-14.194 mm	-76.666 mm
	1565.392 mm	830.460 mm
Rotation R	Yaw -140.702 grd	-123.766 grd
	Pitch 21.164 grd	41.392 grd
	Roll -23.704 grd	-23.016 grd

Interne Kameraparameter		
Parameter	Kalibrierung III	Kalibrierung IV
Brennweite B	35.117 mm	14.767 mm
Skalierungsfaktor s_x	1.013	1.018
Verzerrungskoeffizienten	$\kappa_1 = -5.7 \cdot 10^{-6} \frac{1}{mm^2}$	$3.3 \cdot 10^{-7} \frac{1}{mm^2}$
	$\kappa_2 = -1.3 \cdot 10^{-5} \frac{1}{mm^2}$	$-1.2 \cdot 10^{-5} \frac{1}{mm^2}$

Kalibrierungsfehler		
Parameter	Kalibrierung III	Kalibrierung IV
Mittelwert Projektion mp	0.564 pix	0.459 pix
Mittelwert Rückprojektion mb	0.268 mm	0.232 mm
Standardabweichung Projektion sp	0.337 pix	0.260 pix
Standardabweichung Rückprojektion sb	0.141 mm	0.163 mm

Tabelle 2.4: Ergebnis der Kamera-Kalibrierung bei realen Kalibrierungsdaten.

2.5.4 Kalibrierung von Realdaten

Bei der Verwendung von realen Daten muß das Kalibrierungsobjekt neu überdacht werden, da mehrfach gestaffelte Ebenen durch ihre Verdeckung nicht einfach zu handhaben sind. Dieses Problem kann man zwar durch mehrere Aufnahmen von einer Ebene in verschiedenen Abständen lösen, jedoch ist selbst unter Laborbedingungen die exakt parallele Positionierung äußerst schwierig. Aus diesem Grund haben wir die Ebenen mit den Kalibrierungspunkten orthogonal fixiert. Die Tabelle (2.4) zeigt die Ergebnisse bzw. Fehler von Kalibrierungen mit realen Daten.

Die Abbildung (2.10) gibt eine grobe optische Kontrolle der erreichten Qualität. Dabei wurden die Weltkoordinaten der Kalibrierungspunkte mit den berechneten Parametern in das Bild projiziert und mit einem Kreuzchen versehen. Bei einer guten Bestimmung der Parameter müssen sie die Position der real abgebildeten Kalibrierungspunkte exakt überlagern. Zusätzlich wurde das Millimeterraster des bekannten Drehtellers in das digitalisierte Bild eingeblendet, um die berechnete Orientierung und Skalierung einfacher verifizieren zu können. In der linken Szene III mit den zwei kleinen Kalibrierungsebenen ist das Bild ungünstig aufgeteilt. Genauere Ergebnisse lassen sich erzielen, wenn die Kalibrierungspunkte wie in der rechten Szene IV besser über das komplette Bild verteilt werden und den gesamten Raum abdecken.

Literaturverzeichnis

- [AZI71] **Abdel-Aziz, Y.I. and Karara, H.M.**: *Direct Linear Transformation into Object Space Coordinates in Close-Range Photogrammetry*, Symposium on Close-Range Photogrammetry, University of Illinois at Urbana-Champaign, pp. 1-18, Jan. 26-29 (1971).
- [AZI74] **Abdel-Aziz, Y.I. and Karara, H.M.**: *Photogrammetric Potential of Non-Metric Cameras*, Civil Engineering Studies, Photogrammetry Series No. 36, University of Illinois at Urbana-Champaign, Mar. (1974).
- [FAI75] **Faig, W.**: *Calibration on Close-Range Photogrammetry Systems: Mathematical Formulation*, Photogrammetric Engineering and Remote Sensing, Vol. 41, No. 12, pp. 1479-1486, (1975).
- [FIS81] **Fischler, M.A. and Bolles, R.C.**: *Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography*, Computer Vision, Graphics and Image Processing 17, pp. 173-180, (1981).
- [GAN84] **Ganapathy, S.**: *Decomposition of Transformation Matrices for Robot Vision*, Proceedings of Int. Conf. on Robotics and Automation, pp. 130-139, (1984).
- [GEN79] **Gennery, D.B.**: *Stereo-Camera Calibration*, Proceedings Image Understanding Workshop, pp. 101-108, Nov. (1979).
- [GRA91] **Grattoni, P. et al.**: *Geometric camera calibration: a comparison of methods*, Proc. '91 ICAR-5th Internat. Conf. Advanced Robotics, Pisa, pp. 1775-1779, Jun. (1991).
- [GRE88] **Gremban, K.D, Thorpe, C.E. and Kanade, T.**: *Geometric camera calibration using system of linear equations*, Proc. Image Understanding Workshop, Cambridge, MA, pp. 820-825, Apr. (1988).
- [HAL82] **Hall, E.L., Tio, M.B.K., McPherson, C.A. and Sadjadi, F.A.**: *Curved Surface Measurement and Recognition for Robot Vision*, Conference Record, IEEE Workshop on Industrial Applications of Machine Vision, May 3-5 (1982).
- [ISA85] **Isaguirre, A., Pu, P. and Summers, L.**: *A New Development in Camera Calibration: Calibrating A Pair of Mobile Cameras*, Proceedings of Int. Conf. on Robotics and Automation, pp. 74-79, (1985).
- [MAN80] *Manual of Photogrammetry*, American Society of Photogrammetry, 4. Auflage, (1980).
- [MAR81] **Martins, H.A., Birk, J. R. and Kelley, R.B.**: *Camera Models Based on Data from Two Calibration Planes*, Computer Graphics and Image Processing, 17, pp. 173-180, (1981).
- [SOB74] **Sobel, I.**: *On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes*, Artificial Intelligence, 5, pp. 185-198, (1974).

- [SON89] **SONY Corporation**: *Bedienungsanleitung 3 CCD Video Camera DXC-730P*, Anhang Technische Daten, pp. 135, (1989).
- [STR84] **Strat, T.M.**: *Recovering the Camera Parameters from a Transformation Matrix*, Proceedings: DARPA Image Understanding Workshop, pp. 264-271, Oct. (1984).
- [SUT74] **Sutherland, I.**: *Three-Dimensional Data Input by Tablet*, Proceedings of the IEEE, Vol. 62, No. 4, pp. 453-461, Apr. (1974).
- [TSA86] **Tsai, Roger Y.**: *An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision*, Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 364-374, (1986).
- [TSA87] **Tsai, Roger Y.**: *A Versatile Camera Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV Cameras and Lenses*, IEEE Journal of Robotics and Automation, pp. 323-344, August 1987.
- [YAK78] **Yakimovsky, Y. and Cunningham, R.**: *A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras*, Computer Graphics and Image Processing, 7, pp. 195-210, (1986).

Kapitel 3

Vorverarbeitung

In diesem Kapitel werden einige Vorverarbeitungsschritte beschrieben, die eine spätere Merkmalsextraktion in den Grauwertbildern erleichtern soll. Eine einfache Operation zur Bildverbesserung stellt z.B. die *globale Grauwertanpassung* dar, die nach jeder realen Aufnahme mit der CCD-Videokamera durchgeführt wurde. Der *dynamische Schwellwert* wurde speziell für die Detektion von Kalibriermarken im Bild eingesetzt. Weil diese den gesamten Raum abdecken sollen, ist eine optimale Ausleuchtung des Kalibrierungsobjekts nicht ganz unproblematisch. Mit einem festen Schwellwert ist es dann nahezu unmöglich alle Kalibrierungspunkte im Bild wiederzufinden. Durch den maximalen lokalen Kontrast der *dynamischen Histogrammegalisierung* wurde versucht, möglichst viele Merkmale hervorzuheben, die dann über mehrere Bilder verfolgt werden können.

3.1 Globale Grauwertanpassung

Bei ungünstigen Beleuchtungsverhältnissen kann die *Helligkeit* und der *Kontrast* eines Bildes auf einfache Weise verbessert werden. Die folgende lineare, ortsunabhängige und globale Transformation nutzt den ganzen zur Verfügung stehenden Grauwertbereich $[G_{min}, \dots, G_{max}]$ aus

$$f'(x, y) = (f(x, y) - f_{min}) \cdot \frac{G_{max} - G_{min}}{f_{max} - f_{min}}, \quad (3.1)$$

wobei f_{min} und f_{max} die minimal bzw. maximal vorkommenden Grauwerte des Ausgangsbildes f beschreiben. Unter- bzw. Überbelichtungen werden durch die konstante Grauwertverschiebung ausgeglichen und eine Multiplikation mit einem Faktor größer als 1 verursacht eine Kontrastverstärkung (vgl. Abb.3.1).

3.2 Dynamischer Schwellwert

Bei dem adaptiven Schwellwert-Verfahren wird das Ausgangsbild f in $n \times n$ Felder eingeteilt und für jedes einzelne Feld ein lokales Histogramm berechnet. Es wird nun ein lokaler Schwellwert gesucht, der die interessanten Objekte vom Hintergrund trennt, wozu jeweils eine *Bimodalitätsprüfung* [HAB89] mit folgenden Schritten durchgeführt wird:

Im Histogramm wird von G_{min} aufsteigend das erste lokale Maximum ermittelt. In einem bestimmten Fangbereich wird nach weiteren lokalen Maxima gesucht. Von G_{max} absteigend wird ebenfalls das erste Maximum gesucht und ein Fangbereich verwendet. Liegt zwischen den ermittelten Maxima ein weiteres lokales Maximum, so ist das Histogramm nicht bimodal.

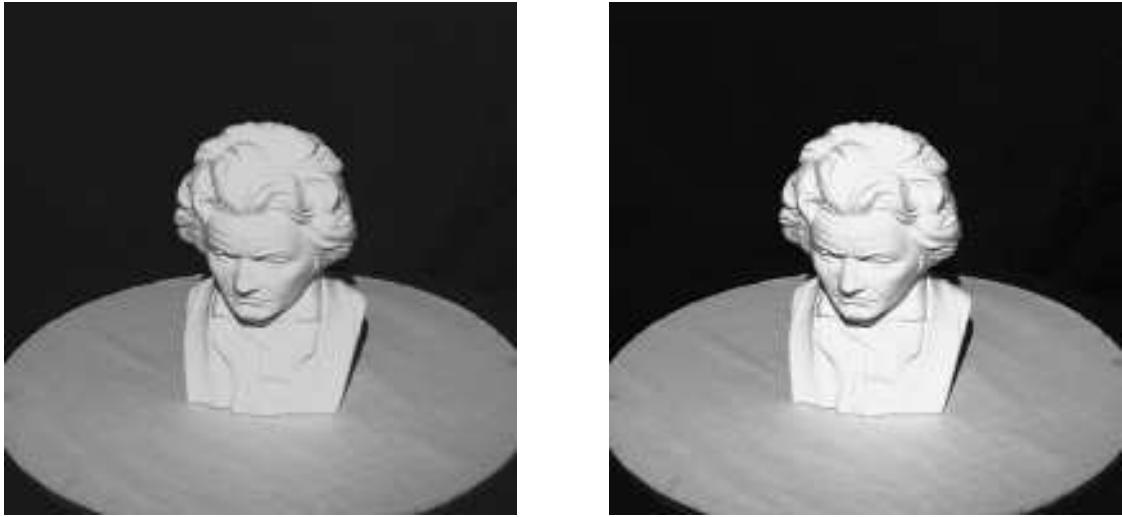


Abbildung 3.1: Kontrastanhebung durch globale Grauwertanpassung.

Anschießend wird ein Minimum zwischen den lokalen Maxima gesucht und das Verhältnis zu dem kleineren der beiden Maxima betrachtet. Das Histogramm wird als bimodal akzeptiert, wenn dieses Verhältnis größer als ein vorgegebener Schwellwert ist. Dadurch erhält man für jedes der $n \times n$ Histogramme automatisch einen lokalen Schwellwert.

Ist ein Histogramm nicht bimodal, d.h. es konnte kein Schwellwert abgeleitet werden, so wird dieser durch arithmetische Mittelung der bereits berechneten Schwellwerte in der 8-Nachbarschaft ergänzt. Es muß allerdings verhindert werden, daß es an den Übergängen zwischen den Feldern durch die unterschiedlichen lokalen Schwellwerte zu Diskontinuitäten kommt. Dazu wird der lokale Schwellwert S jeweils in der Feldmitte angenommen und zwischen den einzelnen Feldern bilinear interpoliert (siehe Abb.3.2). In Abhängigkeit vom Abstand zum Feldmittelpunkt der Nachbarfelder (a, b) ergibt sich für einen Bildpunkt an der Stelle (x, y) folgender Schwellwert:

$$S(x, y) = (1 - a)(1 - b) \cdot S(i, j) + (1 - a)b \cdot S(i, j + 1) + a(1 - b) \cdot S(i + 1, j) + ba \cdot S(i + 1, j + 1). \quad (3.2)$$

Das Ergebnis der Transformation ist abhängig von der Feldgröße. Werden die Felder zu groß gewählt, so entspricht die Transformation einer herkömmlichen globalen Schwellwertbildung. In einem Feld sollten mindestens so viele Punkte liegen, wie Grauwerte zur Verfügung stehen, da sonst die Schätzung der Grauwertverteilung nicht repräsentativ sein kann. Somit ergeben sich sinnvolle Feldgrößen zwischen 16×16 und 64×64 Bildpunkten.

Bei der Bimodalitätsprüfung kann der Fangbereich automatisch bestimmt werden. Beginnend von $0.5 \cdot (G_{max} - G_{min})$ wird er solange verengt, bis er zu Null wird oder das Histogramm als bimodal erkannt wird. Die Schwelle für die Akzeptanz eines bimodalen Histogramms sollte nicht zu niedrig gewählt werden, da ansonsten falsche Schwellwerte bestimmt werden. Das Zurückweisen eines Histogramms wirkt sich durch die Interpolation der korrekten Nachbarn weniger gravierend aus. Ein Beispiel für die Anwendung des dynamischen Schwellwertes ist in Abb. (3.3) gegeben.

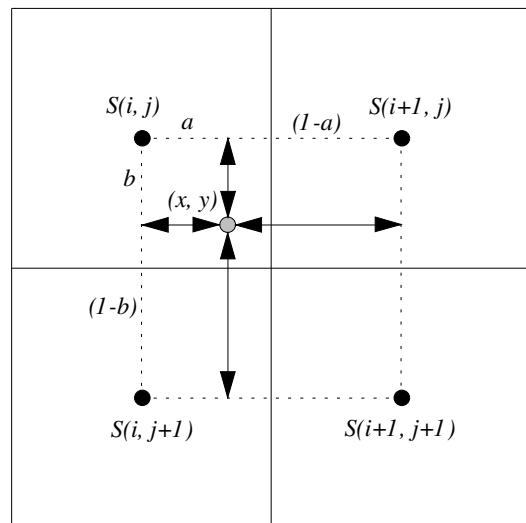
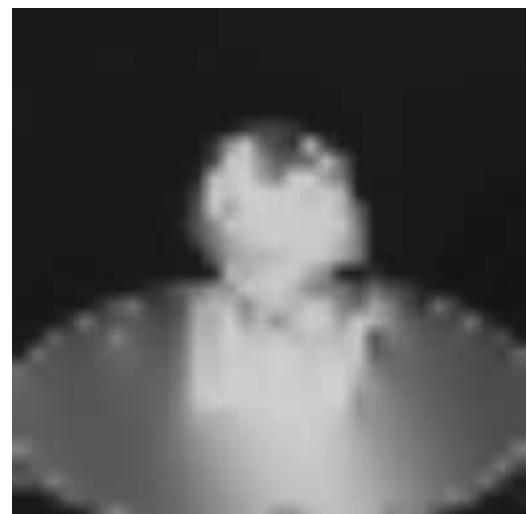


Abbildung 3.2: Interpolation der lokalen Schwellwerte.



a.) Binärbild bei dynamischem Schwellwert.



b.) Der Schwellwert bei Fenstergröße 16.

Abbildung 3.3: Anwendung des dynamischen Schwellwertes.



a.) Fenstergröße 16.

b.) Fenstergröße 64.

Abbildung 3.4: Histogrammegalorisierung mit verschiedenen Fenstergrößen

3.3 Dynamische Histogrammegalorisierung

Damit einzelne Gebiete gut unterscheidbar werden, ist ein optimaler lokaler Kontrast erforderlich. Bei der sogenannten Histogrammegalorisierung [WAH84] werden dazu die Intensitätswerte gleichmäßig auf den gesamten zur Verfügung stehenden Grauwertbereich verteilt. Dazu wird ein Vektor mit den neuen Intensitätswerten (*Look-Up-Tabelle*) erstellt, wobei die Grauwerte f des Ausgangsbildes folgendermaßen als Index dienen:

$$G'(i) = \frac{G_{\max} \cdot K(f_i)}{N} \quad \text{für} \quad i \in \{G_{\min} \dots G_{\max}\} \quad (3.3)$$

mit

$$\begin{aligned} K(f_i) &= \left(\sum_{j=f_{\min}}^{f_i} H(j) \right) - \frac{(H(f_i) + H(f_{\min}))}{2}, \\ N &= \left(\sum_{j=f_{\min}}^{f_{\max}} H(j) \right) - \frac{(H(f_{\max}) + H(f_{\min}))}{2}. \end{aligned}$$

Man bezeichnet K als *kumulative Verteilungsfunktion* und N normiert auf die Pixelanzahl, wobei $H(f_{\min})$ und $H(f_{\max})$ die Häufigkeit des kleinsten bzw. größten Grauwertes angeben. Um den negativen Einfluß dominanter Grauwertbereiche auszuschließen, wird das Bild, wie oben beschrieben, in mehrere Felder eingeteilt und lokale Histogramme erstellt. Der Grauwert f eines Bildpunktes an der Stelle (x, y) dient als Index zur Transformation in den benachbarten Histogrammen. Der eigentliche Grauwert wird anhand der relativen Gewichtung zu den resultierenden Grauwerten entsprechend interpoliert. Die Auswirkung unterschiedlicher Fenstergrößen auf die dynamische Histogrammegalorisierung wird in Abb. (3.4) deutlich.

3.4 Medianfilter

Der *Medianfilter* gehört zur Klasse der *Rangordnungsoperatoren* [BJA93]. Aus den Pixeln einer Umgebung wird nicht ein Mittelwert gebildet, sondern es wird der Grauwert, der sich in der Mitte der sortierten Liste von Grauwerten aus der Filtermaske befindet, ausgewählt. Formal gilt

$$\text{Median} = G_{N/2} \in (G_1, G_2, \dots, G_{N-1}, G_N) \subseteq \quad G_i \in \text{Umgebung}, \quad 1 \leq i \leq N$$

Mit \subseteq als reflexive, asymmetrische, transitive Relation.

Die wichtigsten *deterministischen* Eigenschaften des Medianfilters sind, daß

- Grauwertkanten erhalten bleiben,
- Grauwertimpulse eliminiert werden (Rauschunterdrückung) und
- der Wertebereich der Bildfunktion nicht erweitert wird.

Anwendung

Alle real aufgenommenen Grauwertbilder werden in der Vorverarbeitung Median gefiltert. Es wird aus Gründen der Effizienz eine eindimensionale Filtermaske der Größe 3 in X- und Y-Richtung angewendet. Als Ordnung wird die bekannte Größer-Relation der Natürlichen Zahlen verwendet.

Kapitel 4

Merkalsextraktion

4.1 Subpixelgenaue Punktbestimmung

Im folgenden wird beschrieben, wie die Koordinaten von Bildpunkten zur subpixelgenauen Bestimmung von Marken in Kalibrierungsbildern errechnet werden. Die grundsätzliche Idee besteht in der Berechnung der Zentroiden der Flächen der Punkte. Dabei müssen jedoch Digitalisierungseffekte berücksichtigt werden.

4.1.1 Grundsätzliche Überlegungen

Unter der Annahme, daß ein Weltpunkt schwarz und der Hintergrund weiß ist, wird eine Kante nicht immer genau auf das Gitter des Kamerasensors abgebildet. In den nachfolgenden Betrachtungen wird ein ideales Kameramodell, das heißt ideale Perspektivprojektion, angenommen. Dies bedeutet, daß alle Punkte *scharf* abgebildet werden, und keine *Unschärfe* im Bild durch fehlerhafte Fokussierung entstehen kann. Die Abb. 4.1 veranschaulicht den Sachverhalt der Digitalisierung.

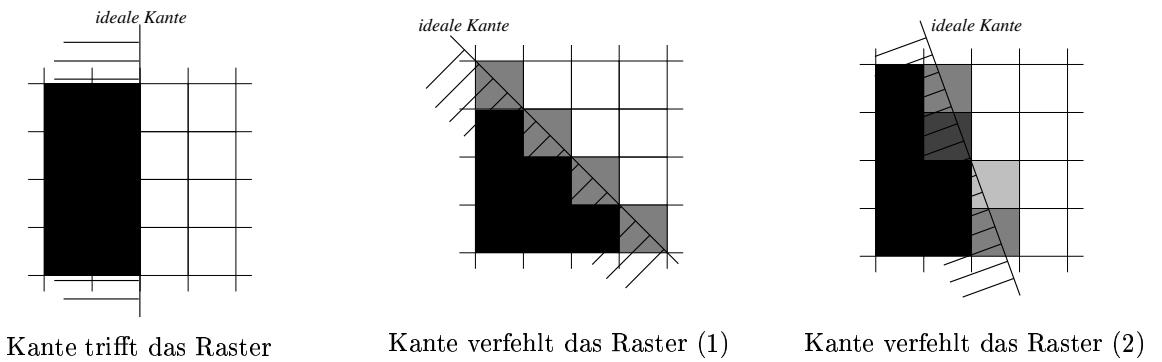


Abbildung 4.1: Digitalisierungseffekt am Beispiel einer Kante

Wie leicht zu erkennen ist, beschränkt sich der Einfluß der Kante auf die unmittelbare Nachbarschaft. In einem Gebiet Ω - der Fläche eines abgebildeten Kalibrierungspunktes - definieren wir die im Idealfall schwarzen Rasterpunkte P als *Kern* und die grauen als *Rand*. Beides zusammen ergibt das Gebiet Ω , wobei jedoch die Randpunkte gewichtet mit ihrem Grauwert G in die Berechnung des Zentroiden eingehen müssen, da sie nur Teile der Kante repräsentieren.

Für ein Grauwertfunktion $g(x, y)$ eines Bildes lautet die allgemeine Definition absoluter Momente $m_{r s}$ höherer Ordnung

$$m_{r s} = \int_x \int_y x^r y^s g(x, y) dx dy .$$

Der *Zentroid* $C = (x_c, y_c)$ ist darauf basierend definiert als

$$(x_c, y_c) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) ,$$

wobei m_{00} als die *Fläche* von Ω bekannt ist.

Im Falle einer diskreten digitalen Bildfunktion $f(x, y)$ und damit einer begrenzten Anzahl von Punkten, verändern sich die Berechnungen von der Integration zur endlichen Summationen über ein Gebiet Ω .

$$\begin{aligned} m_{r s} &\approx \sum_x \sum_y x^r y^s \frac{f(x, y)}{G_{max}} & \forall (x, y) \in \Omega \text{ und } G_{max} \neq 0 \\ &\approx \sum_{f(x, y) \in \Omega} x^r y^s \frac{f(x, y)}{G_{max}} \end{aligned}$$

Der Divisor G_{max} (maximaler Grauwert $\neq 0$) wird hier eingesetzt, um die einzelnen Werte $f(x, y)$ von Bildpunkten auf das Intervall $[0, 1]$ zu normieren. Die Berechnung des Zentroiden sieht dann folgendermaßen aus:

$$(x_c, y_c) \approx \left(\frac{\sum_{f(x, y) \in \Omega} x f(x, y)}{\sum_{f(x, y) \in \Omega} f(x, y)}, \frac{\sum_{f(x, y) \in \Omega} y f(x, y)}{\sum_{f(x, y) \in \Omega} f(x, y)} \right)$$

Um eine korrekte Gewichtung der Grauwerte bzgl. unserer Semantik eines schwarzen Punktes zu erhalten, muß dafür gesorgt werden, daß der *Kern* (die schwarzen Punkte) den maximalen Grauwert G_{max} und der Hintergrund (die weißen Punkte) den Wert Null erhält. Dies kann wenn nötig durch Invertierung des Grauwertbildes erreicht werden.

In der Praxis wird zur Berechnung der Zentroide der Kalibrierungspunkte, wie in Abb.4.2 dargestellt, vorgegangen.

4.1.2 Anmerkung

Als Kalibrierungspunkte haben sich Kreise, deren Abbilder größer als ein Pixel sind, bewährt. Diese kreisförmigen Merkmale werden als verzerrte Ellipsen abgebildet, deren perspektivische Verzerrung auch den Zentroiden C des Kreises verschiebt (siehe Abb. 4.3).

In unserer Praxis hat sich dieser Fehler, der abhängig von der Größe der Ellipsenfläche ist, als unterhalb der möglichen detektierbaren Genauigkeit und somit als vernachlässigbar herausgestellt.

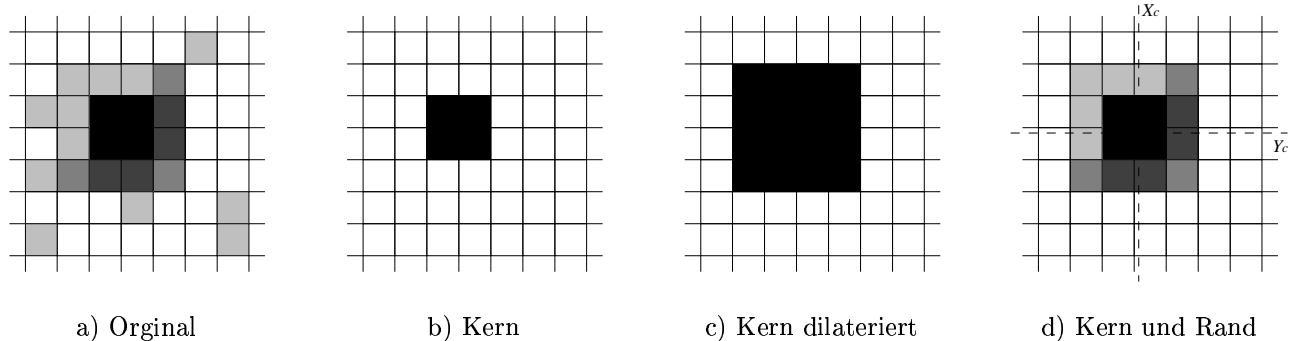


Abbildung 4.2: Subpixelgenaue Punktebestimmung: b) Über einen Schwellwert wird der *Kern* des Originalbildes a) selektiert. c) Durch binäre Dilatation mit einer 3×3 Maske wird der *Rand* miteinbezogen. d) Mit dem berechneten Bild c) als Schablone werden die Grauwerte aus dem Originalbild a) ausgestanzt und der Zentroid von Ω wird jetzt grauwertgewichtet bestimmt.

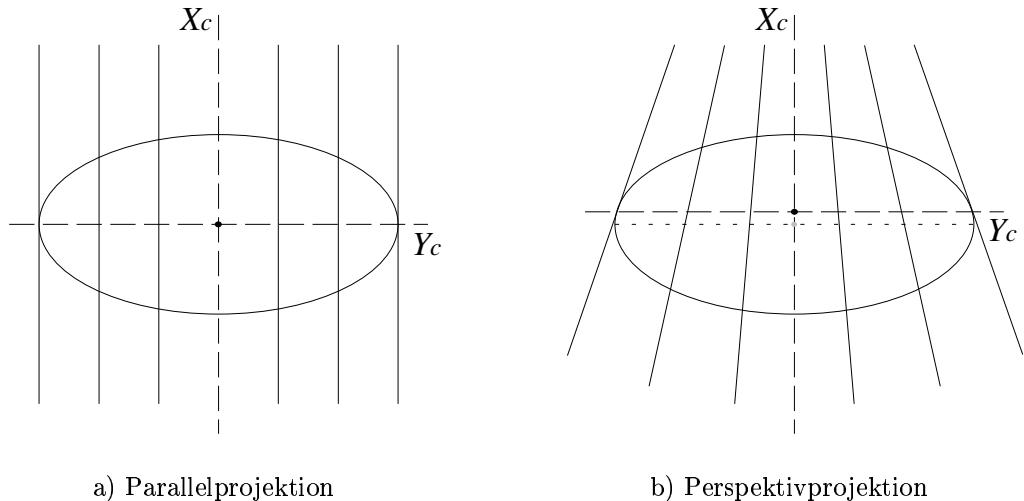


Abbildung 4.3: Perspektivische Verzerrung von Ellipsen a) Beispiel der Parallelprojektion eines Kreises aus einer Ebene antiparallel zur Projektionsebene, b) Perspektivprojektion des zuvor beschriebenen Kreises (hier die Darstellung der vertikalen Abweichung des Zentroiden von den tatsächlichen Koordinaten)

4.2 Der Marr-Hildreth-Operator

Im folgenden sollen *optimale Operatoren* zur Detektion von Kanten in Grauwertbildern vorgestellt werden, die im Projekt zur Merkmalsbestimmung eingesetzt werden. Die Theorie des Marr-Hildreth-Operators stützt sich auf neurophysiologische Untersuchungen des menschlichen Sehens und funktioniert wie ein *Bandpaßfilter*, also einer Kombination aus Tiefpaßfilter mit einem Hochpaßfilter [MAR80, HIL83].

4.2.1 Anforderungen

Der Operator sollte eine begrenzte Bandbreite haben, weil es kaum möglich ist, einen *optimalen Kantenoperator* für alle Ortsfrequenzbereiche zu finden. Die jeweils zu verwendende Bandbreite sollte über Parameter einzustellen sein.

Durch die Feststellung, daß bei der Lokalisierung von Kanten im Ausgangsbild die Intensitätsänderungen örtlich beschränkt sind, ist folgendes klassisches Problem gegeben:

Unschärferelation der Fourier-Transformation: „Eine genaue Lokalisation in einem Raum, sei es auch nur durch steile Kanten, führt zu einer Verschmierung über weite Bereiche im reziproken Raum.“

Der optimale Operator muß also die beiden widersprüchlichen Anforderungen der begrenzten Bandbreite im Ortsfrequenzraum und des begrenzten Einzugsbereichs im Ortsraum minimieren.

4.2.2 Laplacian-of-Gaussian (LoG)

Der Marr-Hildreth-Operator folgt der klassischen Vorstellung eines Kantenoperators, bestehend aus einem Glättungsfilter und einem Ableitungsoperator.

Glättungsfilter

Neben der Dirac-Funktion ist bekanntermaßen die *Gauß-Funktion* (Glockenkurve) die einzige, die sowohl im Orts- als auch im Frequenzraum ihre Form beibehält und somit zur Glättung verwendet werden kann (siehe Abb. 4.4). Der erste Schritt beim Marr-Hildreth-Filter ist die Glättung des Ausgangsbildes anhand der Faltung

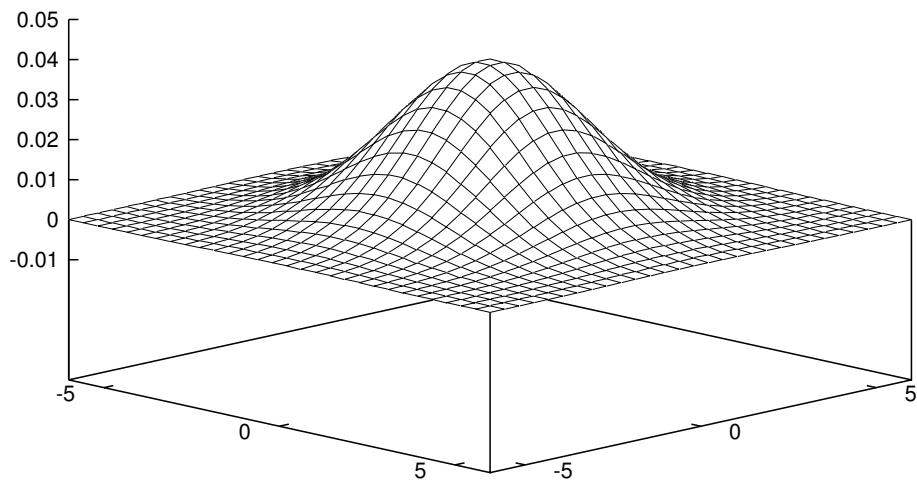


Abbildung 4.4: Gaußfunktion mit Standardabweichung $\sigma = 2$.

$$f_g(x, y) = G_\sigma(x, y) * f(x, y) \quad (4.1)$$

mit dem Gauß-Filter

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right] \quad (4.2)$$

und der Standardabweichung σ . Bei einer Filtergröße von $(2r + 1) \times (2r + 1)$ ergibt sich insgesamt:

$$f_g(x, y) = \frac{1}{2\pi\sigma^2} \sum_{i=-r}^{+r} \sum_{j=-r}^{+r} \exp\left[-\frac{(i^2 + j^2)}{2\sigma^2}\right] f(x + i, y + j). \quad (4.3)$$

Ableitungsoperator

Im zweiten Schritt wird mit dem linearen und rotationssymmetrischen *Laplace-Operator* eine Anhebung der hohen Frequenzen durchgeführt.

$$f_{LoG}(x, y) = \nabla^2 f_g(x, y) \quad (4.4)$$

Beide Filter sind linear und ortsunabhängig, so daß sie zu einem Filter, dem sogenannten *Marr-Hildreth-Operator*

$$f_{LoG}(x, y) = \nabla^2(G_\sigma(x, y) * f(x, y)) = (\nabla^2 G_\sigma(x, y)) * f(x, y) \quad (4.5)$$

mit

$$\nabla^2 G_\sigma(x, y) = -\frac{1}{\pi\sigma^4} \left(\frac{(x^2 + y^2)}{2\sigma^2} - 1 \right) \exp\left[-\frac{(x^2 + y^2)}{2\sigma^2}\right] \quad (4.6)$$

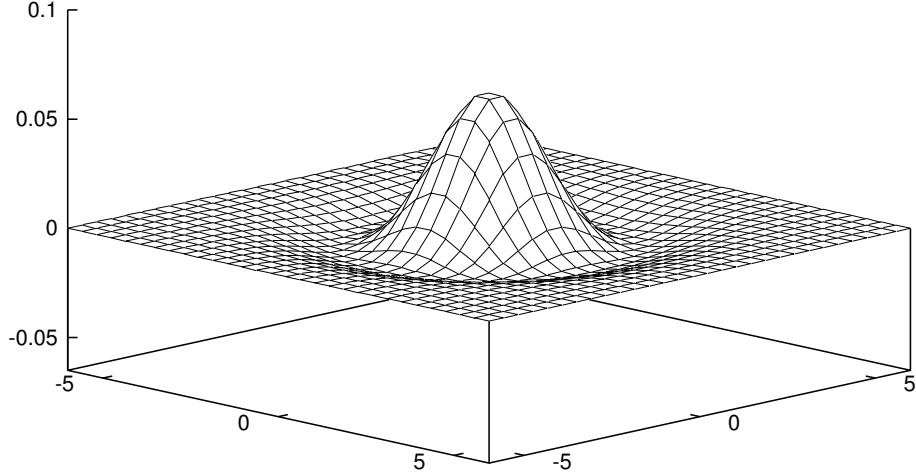


Abbildung 4.5: LoG-Operator mit Standardabweichung $\sigma = 1.5$.

kombinert werden können. Wegen der charakteristischen Form von $\nabla^2 G_\sigma$ wird der Operator häufig auch „Mexikanischer Hut“ genannt (vgl. Abb. 4.5). Der Marr-Hildreth-Operator benötigt als Parameter nur die Standardabweichung σ , da sich die Filtermaskengröße ableiten lässt. Bei einer inneren positiven Filterregion von $\omega = 2\sqrt{2}\sigma$, ist eine Filtergröße von 4ω ausreichend, wobei ein 3ω großer Filter schon zu

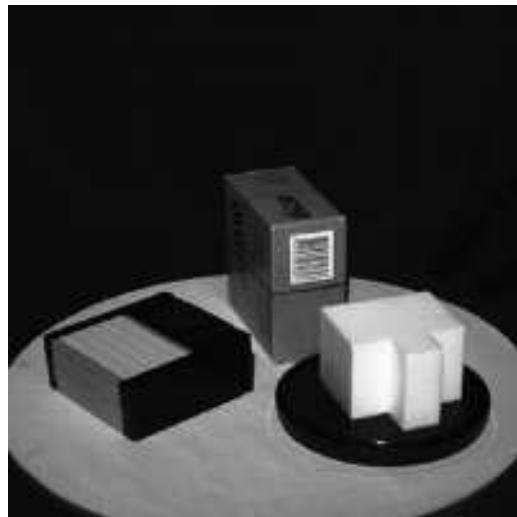
sichtbaren Fehlern führen kann [SOT89]. Bei der Nulldurchgangssuche wird innerhalb der 8-Nachbarschaft die Steigung mit einem Schwellwert für die Mindeststärke der Kante verglichen und ein Binärbild erzeugt.

4.2.3 Modifikationen

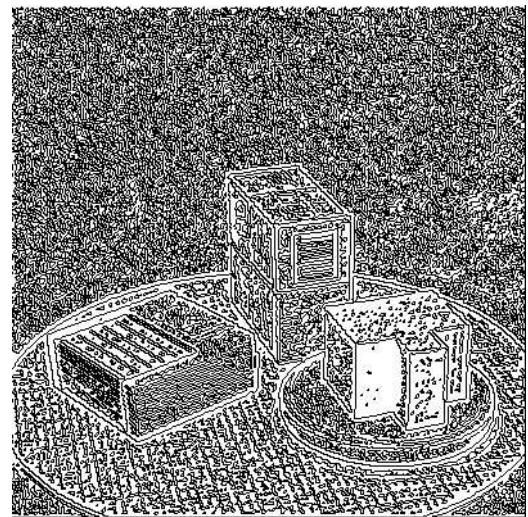
Der Skalierungsfaktor $\frac{1}{\pi\sigma^4}$ kann durch einen beliebigen für die Berechnung günstigen Faktor ersetzt werden, da es insgesamt nur auf die Detektion von Nullstellen ankommt. Um das Hintergrundrauschen von den echten Kanten unterscheiden zu können, haben wir zusätzlich über die erste Ableitung des Ausgangsbildes eine entsprechende Schwelle angelegt und binarisiert (vgl. Abb.4.6). Die Verknüpfung des binarisierten Gradientenbildes mit dem Ergebnis des LoG-Operators zeigt gute Ergebnisse, führt aber zu Unterbrechungen in den ansonsten geschlossenen Kanten.

4.2.4 Eigenschaften

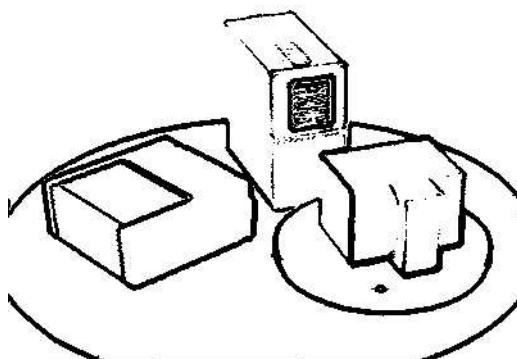
- Die detektierten Kanten sind aufgrund der Nulldurchgangssuche immer einen Pixel breit und bilden immer geschlossene oder am Bildrand endende Kurven [TOR86].
- Gänzlich fehlt diesem Operator ein Maß für die Stärke einer Kante. In leicht verrauschten Gebieten werden daher Pseudokanten detektiert.
- In verrauschten Bildern können prinzipiell keine Kreuzungspunkte mit einer ungeraden Zahl sich treffender Kanten detektiert werden.
- Eine Verschiebung der detektierten Kante gegenüber der tatsächlichen Kantenposition kann entstehen, wenn mehrere Kanten innerhalb des zentralen Bereiches des LoG-Operators liegen [BER84, LUN86].
- Da bei stärkerer Glättung die Lokalisation der Kanten schlechter wird, ist die Verwendung kleiner Filtermasken sinnvoll.



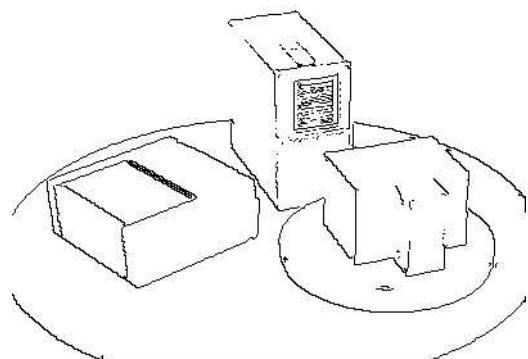
a.) Polyedrische Szene als Eingabebild.



b.) Ergebnis mit dem LoG-Operator.



c.) Binarisierter Gradient von a.).



d.) Kombination von b.) und c.).

Abbildung 4.6: Merkmalsextraktion mit dem modifizierten Marr-Hildreth-Operator.

4.2.5 Subpixel-Informationen

Der Marr-Hildreth-Operator liefert normalerweise pixelgenaue Informationen in Form eines Binärbildes, das die Nulldurchgänge der zweiten Ableitung darstellt. Um dennoch genauere Angaben über die Nullstellen zu erhalten, betrachten wir sie in den folgenden Grafiken der digitalisierten zweiten Ableitung (zur Vereinfachung eindimensional dargestellt):

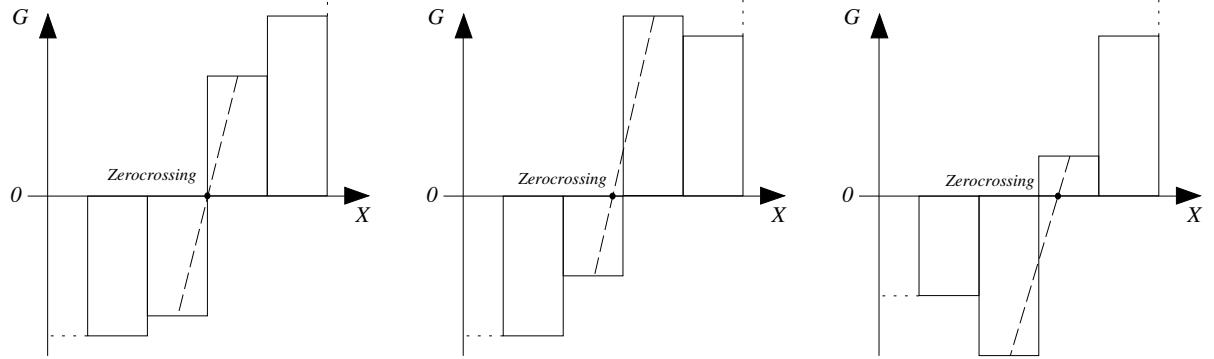


Abbildung 4.7: subpixelgenaue Nullstellen

Wie leicht zu erkennen ist, kann eine subpixelgenaue Position der Nullstellen mittels der Steigung der beiden Grauwerte die die Nullstelle erzeugen berechnet werden. Auf diese Weise kann die Abweichung des Zerocrossings von ihrer Repräsentation im Binärbild sowohl in X- als auch in Y-Richtung bestimmt werden, was zwar nicht immer exakt der wirklichen 2D-Position entspricht, jedoch zusätzliche Genauigkeit liefert.

4.3 Der Canny-Operator

Die Herleitung des Canny-Operators ist streng mathematisch formuliert [CAN83, CAN86]. Nach der Aufstellung einiger grundsätzlicher Annahmen und Optimierungs-Kriterien, wie z.B.

- geringe Fehlerrate,
- gute Lokalisation,
- nur eine Filterantwort auf eine einzige Kante,

wird durch Anwendung von Variationsmethoden und ihrer numerischen Lösung die Operator-Funktion gefunden. Die so für eine *Stufenkante* berechnete Filterfunktion wird mit Hilfe der ersten Ableitung der Gauß-Funktion approximiert.

4.3.1 Filterfunktion

Zur Kantendetektion wird zuerst die Bildfunktion f mit den *Richtungsableitungen der Gauß-Funktion* in x - und y -Richtung gefaltet:

$$D_x(x, y) = \frac{\partial}{\partial x} (G_\sigma(x, y) * f(x, y)) = \frac{\partial G_\sigma(x, y)}{\partial x} * f(x, y), \quad (4.7)$$

$$D_y(x, y) = \frac{\partial}{\partial y} (G_\sigma(x, y) * f(x, y)) = \frac{\partial G_\sigma(x, y)}{\partial y} * f(x, y). \quad (4.8)$$

Bis auf einen Skalierungsfaktor ist

$$G_\sigma(x, y) = \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (4.9)$$

die Gauß-Funktion mit der Standardabweichung σ . Für ihre Normierung verwendet man im allgemeinen den Skalierungsfaktor $\frac{1}{2\pi\sigma^2}$. So ergeben sich die ersten Richtungsableitungen der Gauß-Funktion zu

$$\frac{\partial G_\sigma(x, y)}{\partial x} = -\frac{x}{\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right), \quad (4.10)$$

$$\frac{\partial G_\sigma(x, y)}{\partial y} = -\frac{y}{\sigma^2} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right). \quad (4.11)$$

4.3.2 Gradientenoperator

In zwei Dimensionen besitzt eine Kante auch eine *Orientierung*, die der Richtung der Tangente an der Kontur entspricht. Diese Orientierung lässt sich mit Hilfe des Gradienten bestimmen, da er gerade senkrecht zur Kantenrichtung steht. Der *Gradient* der geglätteten Bildfunktion

$$D'(x, y) = D_x(x, y) \cdot e_x + D_y(x, y) \cdot e_y \quad (4.12)$$

zeigt immer in Richtung des stärksten Anstiegs und wird durch die bereits berechneten Komponenten D_x und D_y festgelegt. Der *Betrag des Gradienten*

$$D(x, y) = |\nabla(G_\sigma(x, y) * f(x, y))| = \sqrt{D_x(x, y)^2 + D_y(x, y)^2} \quad (4.13)$$

entspricht der Größe der Änderung der Gauß-geglätteten Bildfunktion und liefert damit ein gutes Maß für die *Stärke* der Kante.

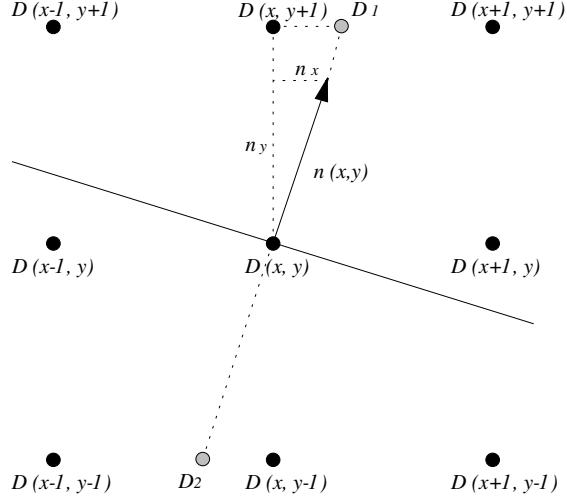


Abbildung 4.8: Maximumssuche anhand der Gradientenrichtung.

4.3.3 Kanten-Nachbearbeitung

Non-Maxima-Suppression

Im nächsten Schritt wird die vorhandene Information über die Gradientenrichtung und -stärke ausgenutzt, um Nicht-Kantenpunkte zu beseitigen. Dazu wird die Normale n zur Kante

$$n(x, y) = n_x(x, y) \cdot e_x + n_y(x, y) \cdot e_y \quad (4.14)$$

aus dem Gradienten und seinen Komponenten $D_x(x, y)$ und $D_y(x, y)$ folgendermaßen bestimmt:

$$n_x(x, y) = \frac{D_x(x, y)}{D(x, y)} \quad \text{und} \quad n_y(x, y) = \frac{D_y(x, y)}{D(x, y)} \quad (4.15)$$

Da in dem diskreten Gitter die Gradientenrichtung im allgemeinen nicht genau in die Richtung eines Gitterpunktes der 8-Nachbarschaft zeigt, muß die Richtung angenähert werden. Aus diesem Grunde werden beim Canny-Operator die zwei gesuchten angrenzenden Gradientenwerte näherungsweise durch lineare Interpolation berechnet (vgl. Abb. 4.8). Entsprechend berechnen sich die beiden *interpolierten Gradienten* D_1 und D_2 zu

$$D_1 = \frac{n_x}{n_y} \cdot D(x+1, y+1) + \frac{(n_y - n_x)}{n_y} \cdot D(x, y+1), \quad (4.16)$$

$$D_2 = \frac{n_x}{n_y} \cdot D(x-1, y-1) + \frac{(n_y - n_x)}{n_y} \cdot D(x, y-1). \quad (4.17)$$

Für die sieben anderen möglichen Richtungen ergeben sich die interpolierten Werte D_1 und D_2 analog. Die Auswahl des richtigen Oktanten erfolgt durch Fallunterscheidung anhand der Richtung der Normalen. Anschließend werden die Kantenpunkte im Gebiet der höheren Intensität gewählt, d.h ein Maximum liegt genau dann vor, wenn $D(x, y) \geq D_1$ und $D(x, y) \geq D_2$ ist. Nach einer Schwellwertbildung der so vorselektierten Kantenpunkte erhält man durch die Non-Maxima-Supression Kanten, die den Ort der stärksten Grauwertänderung beschreiben.

Die detektierten Kanten sind immer einen Pixel breit und dieser Algorithmus kann auch als spezielles Verfahren zur Skelettierung angesehen werden, welches nicht nur die geometrische Anordnung der Punkte verwendet, sondern auch die Informationen aus der vorangehenden Kantendetektionstufe. Aufgrund der sicherlich unrealistischen Annahme, daß nur eine Kante im Bereich der Filterfunktion liegt, entstehen Unterbrechungen und falsche Verbindungen im Bereich von Kreuzungen.

Diagonale Kanten wirken aufgrund der Verbindung über direkte Nachbarn etwas dicker als die orthogonalen Kanten. Durch eine Zusatzbedingung kann aber eine indirekte 8-Nachbarschaftsverbindung erzwungen werden. Bei unserer *Modifikation* wird der aktuelle Punkt p nur dann gesetzt, wenn für die bereits berechneten Nachbarn eine der folgenden Masken übereinstimmt :

<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>p</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td></tr></table>	0	0	0	0	p	x	x	x	x	<table border="1"><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>p</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td></tr></table>	0	0	0	1	p	x	x	x	x	<table border="1"><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>p</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td></tr></table>	1	0	0	0	p	x	x	x	x	<table border="1"><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>0</td><td>p</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td></tr></table>	0	1	0	0	p	x	x	x	x
0	0	0																																					
0	p	x																																					
x	x	x																																					
0	0	0																																					
1	p	x																																					
x	x	x																																					
1	0	0																																					
0	p	x																																					
x	x	x																																					
0	1	0																																					
0	p	x																																					
x	x	x																																					
<table border="1"><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>p</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td></tr></table>	0	0	1	0	p	x	x	x	x	<table border="1"><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>p</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td></tr></table>	0	0	1	1	p	x	x	x	x	<table border="1"><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>p</td><td>x</td></tr><tr><td>x</td><td>x</td><td>x</td></tr></table>	1	0	1	0	p	x	x	x	x										
0	0	1																																					
0	p	x																																					
x	x	x																																					
0	0	1																																					
1	p	x																																					
x	x	x																																					
1	0	1																																					
0	p	x																																					
x	x	x																																					

Abbildung 4.9: Masken für indirekte 8-Nachbarschaftsverbindung.

Hysteresis-Threshold

Über das vorangegangene Verfahren wurde die Anzahl der potentiellen Kantenpunkte erheblich reduziert. Aus der bisher berechneten Menge von möglichen Kantenpunkten muß durch ein geeignetes Schwellwertverfahren entschieden werden, ob es sich bei einem Punkt aufgrund seines Gradientenwertes um einen Kantenpunkt handelt oder nicht. Ein solches Verfahren, das nicht einen festen Wert sondern ein Intervall verwendet, ist das *Hysteresis-Threshold-Verfahren*.

Punkte an denen der Gradientenbetrag den hohen Schwellwert HT_h überschreitet zählen zu den sicheren Kantenpunkten, während Punkte mit Beträgen über dem niedrigeren Schwellwert HT_l der Fortsetzung von Kanten dienen. Mittels eines einfachen Konturverfolgungs-Algorithmus werden von sicheren Kantenpunkten ausgehend benachbarte mögliche Kantenpunkte zur Kontur hinzugefügt.

Das Verfahren ist sehr robust und selbst bei festen Schwellwerten auf verschiedene Bilder anwendbar. Die erzeugten Kanten sind wesentlich länger als bei herkömmlichen einfachen Schwellwertverfahren. Auf einer Kontur mit Gradientenbeträgen zwischen dem hohen und dem niedrigen Schwellwert reicht ein einziger sicherer Punkt um die gesamte Kontur anzuerkennen.

Constraint-Thinning

Bei der Kombination mit der *Non-Maxima-Supression* kommt es zu Fehlern, wenn sich mehrere Kanten im Einzugsbereich des Operators befinden. Mit dem *Constraint-Thinning-Verfahren* ist es nun möglich, diese Lücken in den meisten Fällen wieder zu schließen.

Mit Hilfe eines zusätzlichen Schwellwertverfahrens wird ein *Binärbild des Gradienten* erstellt. Anschließend wird dieses Binärbild mit einem *Skelettierungs-Verfahren* ausgedünnt, allerdings unter der Bedingung, daß die in den vorangegangenen Stufen markierten Maxima nicht gelöscht werden dürfen.

Das Schließen der Lücken mit dem Constraint-Thinning war recht problematisch, da die untersuchten Skelettierungsverfahren von Zhang/Suen [ZHS84], Zhang/Wang [ZHW88] und Sakai [SAK72] eine Vorezugsrichtung besitzen. Abhilfe schaffen konnte erst die parallele Verdünnung mit dem Kontur-Folge-Verfahren von Ji [JI89], das wie folgt skizziert werden kann:

Zuerst werden alle *Randpunkte* im *Binärbild* f bestimmt und in einem Puffer gespeichert. Ein Objektpunkt gilt als Randpunkt, wenn in der 4-Nachbarschaft mindestens ein Hintergrundpunkt existiert. Solange der Puffer mit den Randpunkten noch nicht leer ist, werden die nachfolgenden Schritte durchgeführt. An den Koordinaten der Randpunkte wird mit 3×3 Masken in f überprüft, ob es sich um einen *Skelettpunkt* handelt oder nicht. Als invariantes Kriterium wird folgendes Skelettmodell verwendet:

$x 0 x$	$x 1 x$	$x 0 1$
$1 p 1$	$0 p 0$	$x p 0$
$x 0 x$	$x 1 x$	$x x x$
$x x x$	$x x x$	$1 0 x$
$x p 0$	$0 p x$	$0 p x$
$x 0 1$	$1 0 x$	$x x x$

Abbildung 4.10: Masken zur Überprüfung der Skeletteigenschaft.

Im Falle eines Skelettpunktes wird dieser in das Ergebnisbild übertragen. Ansonsten wird mit folgenden 3×3 Masken in f getestet, ob es sich um einen *Endpunkt* handelt:

$0 1 0$	$0 0 0$	$0 0 0$	$0 0 0$
$0 p 0$	$0 p 1$	$0 p 0$	$1 p 0$
$0 0 0$	$0 0 0$	$0 1 0$	$0 0 0$

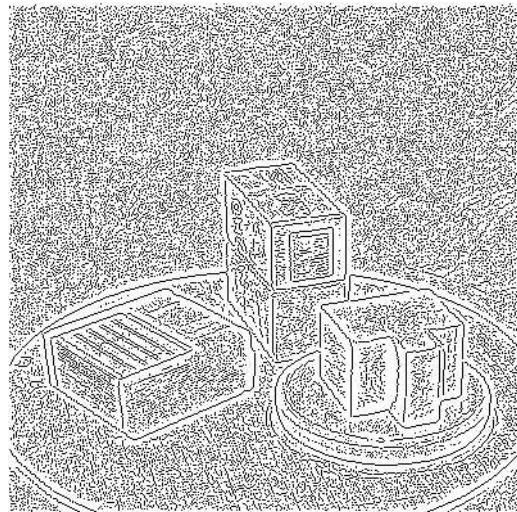
Abbildung 4.11: Masken für die Bestimmung von Endpunkten.

Ist der Punkt weder ein Skelett- noch ein Endpunkt, und ist er in dem vorangegangenen Prozeß noch nicht als sicherer Kantenpunkt fixiert worden (*Constraint*), dann darf er in f gelöscht werden. Anschließend wird der komplette Puffer gelöscht und untersucht, ob neue Randpunkte entstanden sind. Diese werden dann in den Puffer eingetragen und letzten Schritte wiederholt.

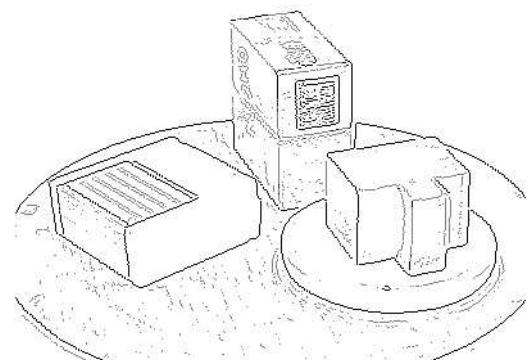
4.3.4 Eigenschaften

- Neben den festen Kantenpunkten schließen nun die Mittelachsenkonturen die Lücken, die durch die Non-Maxima-Supression entstanden sind.
- Der Schwellwert zur Binarisierung sollte etwas niedriger gewählt werden, wie der Schwellwert HT_h vom Hysteresis-Threshold-Verfahren.
- Fehlerhaft detektierte Kantenpunkte und einige seitliche Ausläufer („Fransen“) machen sich negativ bemerkbar.
- Fallen zwei Kanten bei der Binarisierung des Gradienten zusammen, so entstehen beim Skelettieren viele Verbindungen („Leitereffekt“).
- Es ist ein Skelettierungsverfahren empfehlenswert, daß keine feste Abtastrichtung vorausgesetzt. Ansonsten wird die Kontur verschoben, d.h. in den Lücken entstehen „Beulen“ oder die Kontur kann nicht bis auf 1-Pixelbreite abgetragen werden.
- Die Verwendung von verschiedenen Abtastrichtungen liefert kaum bessere Ergebnisse, da sich bei der Verknüpfung die jeweils zu einer anderen Seite verschobene Kante aufhebt, und so die Lücke nicht geschlossen wird.

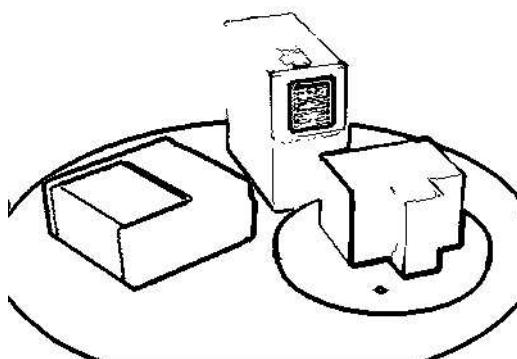
Aufgrund der gesamten Vorgehensweise liefert der Canny-Operator gute Ergebnisse bei der Anwendung auf reale Bilder (siehe Abb.4.12), doch leider wird die Geschlossenheit der Konturen nicht garantiert. Durch die Verwendung der ersten Ableitung ist die Rauschempfindlichkeit wesentlich geringer als z.B. beim Marr-Hildreth-Operator. Obwohl beim Canny-Operator mehrere Stufen zu durchlaufen sind, ist er durch die getrennte Berechnung der Richtungsableitungen um einiges schneller als der Marr-Hildreth-Operator, dessen Faltung mit relativ großen Masken aufwendiger ist.



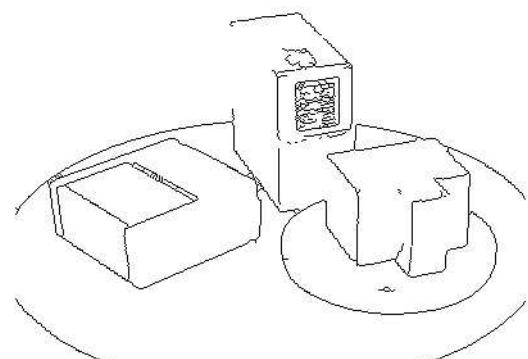
a.) Ergebnis der Non-Maxima-Suppression.



b.) Anwendung des Hysteresis-Threshold auf a.).



c.) Binarisierter Gradient des Eingabebildes mit den bereits erkannten Kanten kombiniert.



d.) Constraint-Thinning von c.).

Abbildung 4.12: Ergebnis der Merkmalsextraktion mit dem Canny-Kantendetektor.

4.4 Monotonie-Operator

Der *Monotonie-Operator* [KOZ84, BJA93] gibt die geometrischen Strukturen im Bild weitgehend unabhängig von der Beleuchtung bzw. dem Kontrast wieder. Er ist für eine Grauwertfunktion $f(m, n)$ folgendermaßen definiert:

$$M(m, n) = \sum_{\forall(i,j) \in \text{Maske}} \begin{cases} 1 & , \text{falls } f(m, n) > f(m + i, n + j) \\ 0 & , \text{sonst} \end{cases} \quad (4.18)$$

In einer Maske ist also der neue Wert des Zentralpixels definiert als die Anzahl der Pixel innerhalb des Maskenbereichs die kleiner oder gleich dem Grauwert des untersuchten Bildpunktes sind. Anhand einer 3×3 Maske soll dies auf einem 5×5 Pixel großen Bildausschnitt veranschaulicht werden:

Punkt:	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>20</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	→	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0																																																	
0	0	0	0	0																																																	
0	0	20	0	0																																																	
0	0	0	0	0																																																	
0	0	0	0	0																																																	
0	0	0	0	0																																																	
0	0	0	0	0																																																	
0	0	8	0	0																																																	
0	0	0	0	0																																																	
0	0	0	0	0																																																	
Linie:	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>10</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>10</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>10</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>10</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>10</td><td>0</td><td>0</td></tr> </table>	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	0	0	10	0	0	→	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>6</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>6</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>6</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>6</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>6</td><td>0</td><td>0</td></tr> </table>	0	0	6	0	0	0	0	6	0	0	0	0	6	0	0	0	0	6	0	0	0	0	6	0	0
0	0	10	0	0																																																	
0	0	10	0	0																																																	
0	0	10	0	0																																																	
0	0	10	0	0																																																	
0	0	10	0	0																																																	
0	0	6	0	0																																																	
0	0	6	0	0																																																	
0	0	6	0	0																																																	
0	0	6	0	0																																																	
0	0	6	0	0																																																	
Kante:	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>11</td><td>11</td><td>11</td></tr> <tr><td>0</td><td>0</td><td>11</td><td>11</td><td>11</td></tr> <tr><td>0</td><td>0</td><td>11</td><td>11</td><td>11</td></tr> <tr><td>0</td><td>0</td><td>11</td><td>11</td><td>11</td></tr> <tr><td>0</td><td>0</td><td>11</td><td>11</td><td>11</td></tr> </table>	0	0	11	11	11	0	0	11	11	11	0	0	11	11	11	0	0	11	11	11	0	0	11	11	11	→	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr> </table>	0	0	3	0	0	0	0	3	0	0	0	0	3	0	0	0	0	3	0	0	0	0	3	0	0
0	0	11	11	11																																																	
0	0	11	11	11																																																	
0	0	11	11	11																																																	
0	0	11	11	11																																																	
0	0	11	11	11																																																	
0	0	3	0	0																																																	
0	0	3	0	0																																																	
0	0	3	0	0																																																	
0	0	3	0	0																																																	
0	0	3	0	0																																																	
Ecke:	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>12</td><td>12</td><td>12</td></tr> <tr><td>0</td><td>0</td><td>12</td><td>12</td><td>12</td></tr> <tr><td>0</td><td>0</td><td>12</td><td>12</td><td>12</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	12	12	12	0	0	12	12	12	0	0	12	12	12	→	<table border="1" style="border-collapse: collapse; width: 100%; height: 100%;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>5</td><td>3</td><td>3</td></tr> <tr><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	0	0	0	0	0	5	3	3	0	0	3	0	0	0	0	3	0	0
0	0	0	0	0																																																	
0	0	0	0	0																																																	
0	0	12	12	12																																																	
0	0	12	12	12																																																	
0	0	12	12	12																																																	
0	0	0	0	0																																																	
0	0	0	0	0																																																	
0	0	5	3	3																																																	
0	0	3	0	0																																																	
0	0	3	0	0																																																	

Die Vorteile des Monotonieoperators sind

- der geringe Rechenaufwand (nur Vergleichsoperationen),
- die Unabhängigkeit von absoluten Grauwerten und dem Kontrast,
- die Klassifikation von geometrischen Strukturen im Bild, was die Zuordnung von Merkmalen vereinfacht.

Nach Tests mit dem Monotonieoperator auf realen Bildern stellt sich sofort heraus, daß die direkte Anwendung zu keinen verwertbaren Ergebnissen führt. Der Einfluß von Rauschen auf die Funktionsweise des Operators erzeugt eine ‘unzureichende’ Klassifikation (Abb. 4.13). Der Operator wird von uns zur Extraktion lokaler Maxima in vorverarbeiteten Bildern benutzt (siehe Abschnitt 4.6).

4.5 Corner Response Funktion

Zur kombinierten Gewinnung von ‘geraden’ Kanten- und Eckenmerkmalen wird die sogenannte *Corner Response Funktion* ($CRF(x, y)$) von [BRA90] verwendet, sie ist definiert durch

$$\begin{aligned} A(x, y) &= W(u, v) * (\nabla_x I(x, y))^2 \quad , \\ B(x, y) &= W(u, v) * (\nabla_y I(x, y))^2 \quad , \\ C(x, y) &= W(u, v) * (\nabla_x I(x, y) \cdot \nabla_y I(x, y)) \end{aligned}$$

und

$$CRF(x, y) = A(x, y) \cdot B(x, y) - C^2(x, y) - \kappa \cdot (A(x, y) + B(x, y))^2 \quad ,$$

wobei $I(x, y)$ das Grauwertbild an der Stelle (x, y) und $W(u, v)$ ein Gauß’scher Tiefpaß ist, der durch eine 2D-Binomialmaske realisiert wird. Die Faltung mit dieser Maske ist durch die sequenzielle Anwendung von 1D-Binomialfiltern mit den Koeffizienten $(1, 6, 15, 30, 15, 6, 1)$ (oder $(1, 4, 6, 4, 1)$) in X- und Y-Richtung durchführbar. Die Gradienten werden durch die Faltung des Bildes mit einfachen eindimensionalen FIR-Filtern $(-1, 0, 1)$ erzeugt. Die $CRF(x, y)$ ist invariant gegen Bildrotation.

Um eine günstige Trennung der Kanten von den Ecken zu erhalten wird für den freien Parameter κ ein Wert von 0.04 gewählt. Auf diese Weise ergeben sich an Grauwertecken positive Werte und an geraden Kanten negative Werte [HST88, BRE90] (Abb. 4.14).

4.6 Anwendung

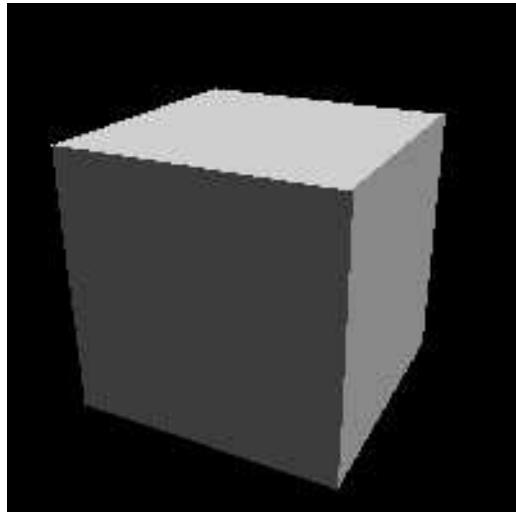
Das Resultat der Corner Response Funktion ist jedoch weiterhin ein Grauwertbild, in dem die Maxima bzw. Minima bedingt durch die absoluten Grauwerte und den Kontrast des Ausgangsbildes unterschiedlich hell auftreten. Dies hat zur Folge, daß eine einfache globale Schwellwertbildung zur Selektion nicht ausreicht.

In unserer Anwendung werden lediglich die punktförmigen Merkmale weiter verwendet, d.h. durch Selektion des positiven Wertebereichs der Corner Response Funktion. Für die Extraktion der gesuchten bzw. benötigten Merkmale liefert eine reine Schwellwertbildung nur Teile bzw. flächenformige Segmente. An dieser Stelle haben wir den Monotonieoperator zur Extraktion der Maxima eingesetzt.

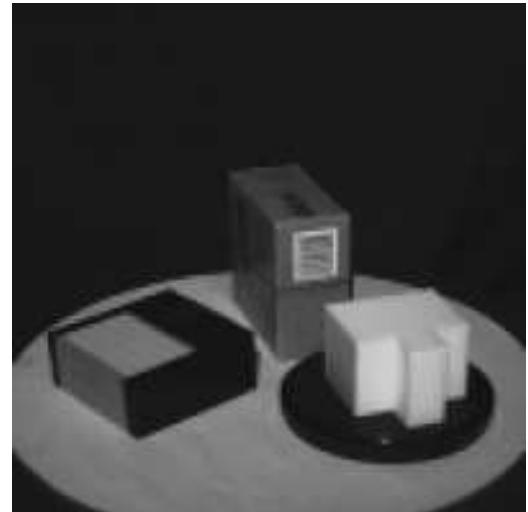
Zur Gewinnung der ‘Selektierten Maxima’ in Abb. 4.15 wird folgendermaßen vorgegangen:

- Die $CRF > 0$ mit einem globalen Schwellwert zur Rauschunterdrückung binarisieren.
- Die $CRF > 0$ mit dem Monotonieoperator falten.
- Das Ergebnis der Faltung invertieren und mit ihm alle Punkte > 0 aus dem binarisierten Bild selektieren.

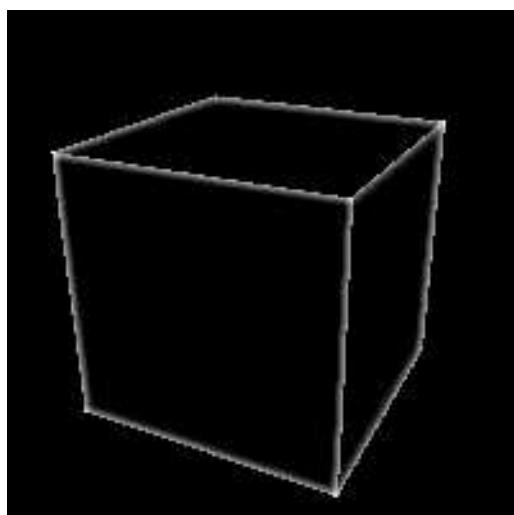
4.7 Beispiele



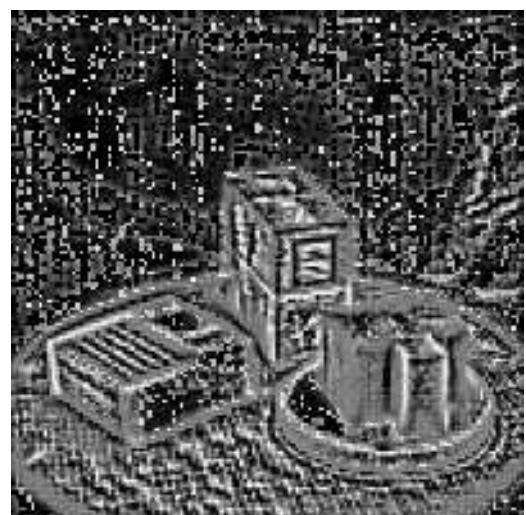
künstliches Eingabebild



reale Polyederszene



Anwendung auf rauschfreies Bild



Ergebnis bei verrauschtem Realbild

Abbildung 4.13: Darstellung der Rauschempfindlichkeit des Monotonieoperators

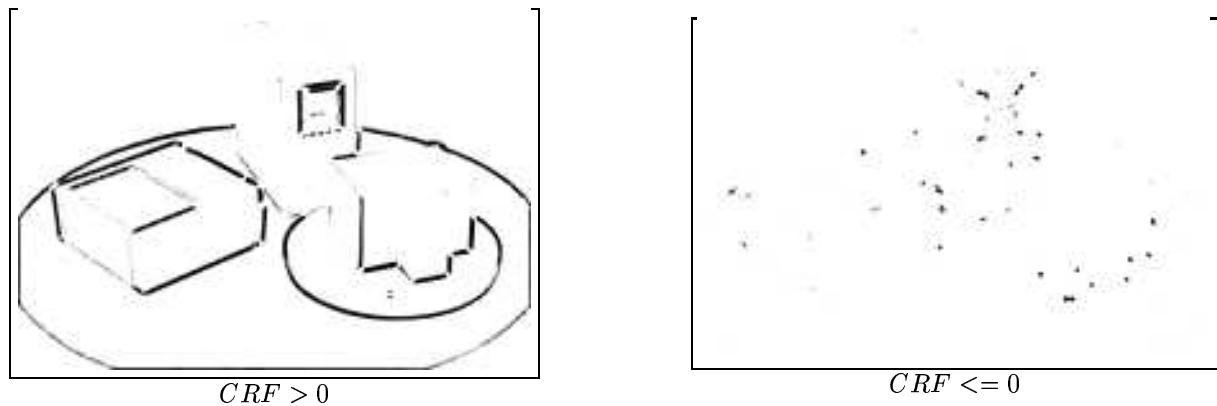


Abbildung 4.14: Die Ergebnisse der Corner Response Funktion

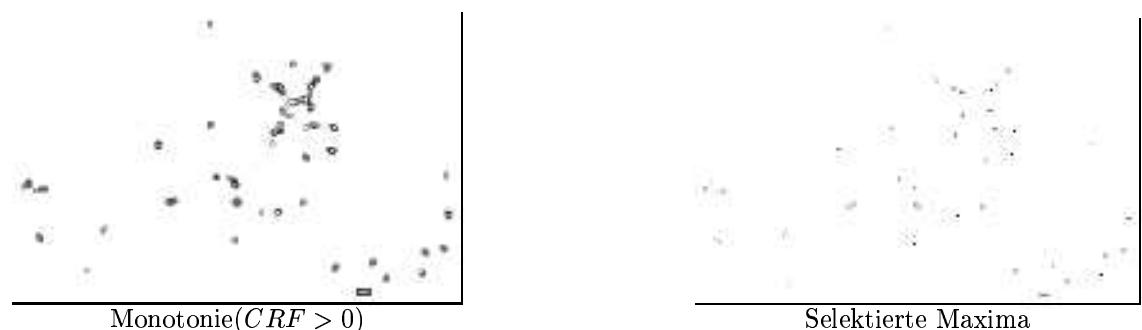


Abbildung 4.15: Die Gewinnung von punktförmigen Merkmalen durch die Kombination der Corner Response Funktion mit dem Monotonieoperator

Literaturverzeichnis

- [BER84] **Berzins, V.**: *Accuracy of Laplacian Edge Detector*, Computer Vision, Graphics and Image Processing, pp. 195-210, 1984.
- [BRA90] **Brand, A. und Zaig, D.**: *Ermittlung v. Eigenbewegung u. Tiefeninformation a. monokularen Bildfolgen*, Siemens AG, Zentralabt. Forschung u. Entwicklung DAGM-Symposium Mustererkennung 1990
- [BRE90] **Breit, H.**: *Bestimmung d. Kameraeigenschaften u. Gewinnung v. Tiefendaten*, Diplomarbeit am Lehrstuhl f. Nachrichtentechnik d. TU München. 30. Sept. 1990
- [CAN83] **Canny, J.**: *Finding Edges And Lines In Images*, Technical Report 720, MIT Artificial Intelligence Laboratory, May 1983.
- [CAN86] **Canny, J.**: *A Computational Approach to Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 679-698, November 1986.
- [HST88] **Harris, C.G. und Stephens, M.J.**: *A combined corner and edge detector*, Proc. of the 4th Alvey Vision Conference, August 1988, pp. 147-152.
- [JI89] **Ji, Yu**: *Ein neues Verfahren zur Verdünnung von Binärbildern*, Mustererkennung 1989, Springer, pp. 72-76, Oktober 1989.
- [MAR80] **Marr, D und Hildreth, E**: *Theory of edge detection*, Proceedings of Royal Society of London, Volume 207, pp. 187-217, 1980.
- [HIL83] **Hildreth, Ellen C.**: *The Detection of Intensity Changes by Computers and Biological Vision Systems*, Computer Vision, Graphics and Image Processing, pp. 1-27, 1983.
- [KOZ84] **Kories, R.; Zimmermann, G.**: *Motion detection in image sequences: an evaluation of feature detectors* Proc. Int. VConf. Patt. Recogn., Montreal 1984 pp. 778-780
- [LUN86] **Lunscher, W. und Beddoes, M.**: *Optimal Edge Detector Design I: Parameter Selection and Noise Effects.*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 164-177, März 1986.
- [MIC89] **De Micheli, E., Caprile, B. et al.**: *Localization and Noise in Edge Detection.*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1106-1117, Oktober 1989.
- [SAK72] **Sakai, et al**: Computer Graphics and Image Processing, pp. 81-96, 1972.
- [SOT89] **Sotak, G. und Boyer, K.**: *The Laplacian-of-Gaussian Kernel: A Formal Analysis and Design Procedure for Fast, Accurate Convolution and Full-Frame Output*, Computer Vision, Graphics and Image Processing, pp. 147-189, 1989.

- [TOR86] **Torre, V. und Poggio, T.**: *On Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 147-163, März 1986.
- [WAH84] **Wahl, F.M.**: *Digitale Bildverarbeitung*, Nachrichtentechnik 13, Springer Verlag, 1984.
- [ZHS84] **Zhang, T. und Suen, C.**: *A Fast Parallel Algorithm for Thinning Digital Patterns*, Communications of the ACM, pp. 236-239, März 1984.
- [ZHW88] **Zhang, Y. und Wang, P.**: *A Maximum Algorithm for Thinning Digital Patterns*, IEEE 9th International Conference on Pattern Recognition, pp. 942-944, November 1988.

Kapitel 5

Verschiebungsvektor-Bestimmung

Zu Beginn dieses Kapitels werden die Vor- und Nachteile bekannter Verfahren zur Berechnung von dichten Verschiebungsvektorfeldern vorgestellt. Als Grundlage dient eine Untersuchung der Leistungsfähigkeit von sogenannten *Optical-Flow-Techniken* durch *Barron und Fleet* [BAR92].

Um einen möglichst optimalen Ansatz für diese Berechnung zu finden, werden im nachfolgenden Abschnitt (5.2) die speziellen Bedingungen der Drehtellergometrie hergeleitet, mit deren Hilfe dann die Bestimmung für ausgezeichnete Punkte (*Merkmale*) durchgeführt wird. Am Ende des Kapitels wird die Gewinnung von dichten Verschiebungsvektorfeldern durch bilineare *Interpolation* beschrieben (Abschnitt 5.4.2).

5.1 Techniken zur Bestimmung des optischen Flußes

Differentielle Techniken

Vorteile :

- Die differentiellen Methoden berechnen die Geschwindigkeit durch räumlich-zeitliche Ableitungen der Grauwertbilder, die teilweise durch Tief- bzw. Bandpaßfilter vorverarbeitet werden.

Nachteile :

- Eine häufige Voraussetzung ist, daß die Verschiebung zwischen zwei Bildern unter einem Pixel liegt.
- Eine exakte numerische Differentiation ist bei Rauschen nahezu unmöglich.
- Die Verfahren sind sehr sensibel bei der Wahl der Parameter und Schwellwerte.
- Die Anzahl der verwendeten Bilder ist mit 12 relativ hoch.

Referenzen :

- *Horn und Schunck* [HOR81] kombinierten eine Gradientenbedingung mit einem globalen Glattheitsterm und bestimmten den Optischen Fluß durch Minimierung. So kann für jeden Punkt im Bild ein Verschiebungsvektor bestimmt werden. Jedoch ist die Genauigkeit durch die verwendete numerische Differentiation allgemein schlecht. Eine räumlich-zeitliche Gaußglättung konnte die Ergebnisse nur optisch verbessern.
- *Lukas und Kanade* [LUC81, LUC84] verwendeten ein Optimierungsverfahren für lokale Gebiete. Die produzierten Schätzungen der Bewegung sind vielversprechend und die Technik stellt konzeptionell und rechentechnisch die einfachste Methode dar. Dennoch erreicht sie nach der Phasen-basierten Methode die höchste Genauigkeit und ist einigermaßen stabil bei der Parameterwahl.

- *Uras, Girosi, Verri und Torre* [URA88] setzten in ihrer Technik die zweite Ableitung ein und bestimmten die Verschiebung anhand der Hessischen Matrix. Ohne die Angabe eines Schwellwertes kann für jeden Bildpunkt ein Verschiebungsvektor berechnet werden. Durch die Verwendung einer Schwelle werden die Ergebnisse einigermaßen genau, aber dann erhält man ein sehr dünn besetztes Verschiebungsvektorfeld.

Region-basiertes Matching

Vorteile :

- Das Matching-Verfahren benötigt nur zwei Bilder.
- Für jeden Bildpunkt kann ein Verschiebungsvektor bestimmt werden.
- Im Test ist es das beste Verfahren für dichte Verschiebungsvektorkarten.

Nachteile :

- Nur bei translatorischen Bewegungen zeigt das Verfahren seine Stärken.

Referenzen :

- Das Verfahren von *Anandan* [ANA87, ANA89] basiert auf einer Laplace'schen Pyramide, deren Ebenen durch Glättung und überlappende Projektion erstellt werden. Mit einer speziellen Matching-Strategie werden von der obersten Ebene angefangen bis zum Bild auf Ebene 0 die Verschiebungsvektoren gefunden.

Energie-basierte Methoden

Vorteile :

- Die Energie-basierten Techniken verwenden die Energie von bewegungsorientierten Filtern. Sie werden häufig auch als frequenzbasiert bezeichnet, da die Filter im Fourier-Raum angewendet werden.

Nachteile :

- Es werden mindestens 7 Bilder benötigt.
- Durch die große Anzahl an Filtern sind es sehr aufwendige Methoden.

Referenzen :

- *Heeger* [HEE88] verwendet eine Gauß'sche Pyramide und setzt 12 verschiedene Garbor Filter ein, um den Optischen Fluß durch eine Optimierung der Filterenergie im Frequenzraum zu berechnen.

Phasen-basierte Methoden

Vorteile :

- Im dem Vergleich der Techniken erreichen sie die höchste Genauigkeit.
- Das Ergebnis ist einigermaßen robust gegenüber den benötigten Parametern und Schwellwerten.

Nachteile :

- Die Anzahl der benötigten Bilder für die räumlich-zeitliche Filterung ist mit 21 am höchsten.
- Es sind sehr aufwendige Methoden mit einer großen Anzahl an Filtern.

Referenzen :

- *Fleet und Jepson* [FLE90, FLE92] verwenden Bandpaßfilter, um das Eingangssignal in Größe, Geschwindigkeit und Orientierung der Bewegung aufzuteilen.

Allgemein erzeugen die Verfahren mit globalen Glattheitsbedingungen optisch ansprechende Ergebnisse, sie sind aber für die Gestaltsrekonstruktion nicht präzise genug. Die Varianten, welche eine ausreichende Genauigkeit anbieten, sind ausgesprochen zeit- und rechenintensiv. Als Ergebnis erhält man schließlich ein äußerst dünn besetztes Verschiebungsvektorfeld.

5.2 Motionkonstraint

Im folgenden wird das aus Stereoverfahren bekannte Epipolarkonstraint [HOR86] der Korrespondenzanalyse für die spezifische Geometrie des Drehtellers modifiziert. Diese Bedingung beruht auf der Tatsache, daß ein im Bild betrachteter Weltpunkt auf dem Sichtstrahl - vom Fokus über den Bildpunkt hinaus - liegt; Voraussetzung ist die kamerazentrierte Perspektivprojektion (vgl. Abb. 5.1). Dieser Strahl bzw. diese Gerade und deren Projektion im zweiten Bild¹ der Folge, stellen eine Beziehung zwischen den Abbildern von Merkmalen dar. Für die Korrespondenzanalyse (d.h. die Verschiebungsvektor-Bestimmung) ist dies eine Einschränkung im Suchraum, der von zwei Dimensionen auf eine Dimension reduziert wird.

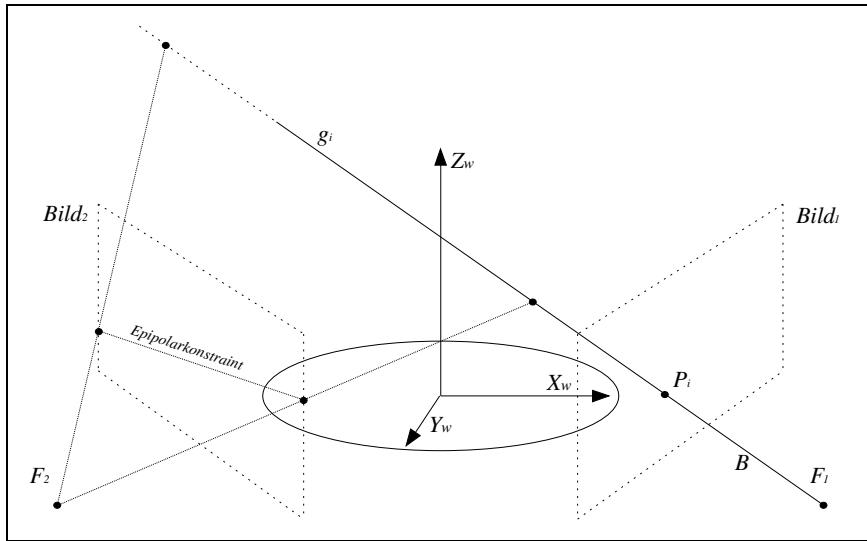


Abbildung 5.1: Epipolarkonstraint

Die *unendliche* Halbgerade g_i bzw. deren *endliche* Projektion im zweiten Bild kann jedoch noch weiter eingeschränkt werden, wenn Kenntnisse über den Objektraum verfügbar sind. Im Falle der Drehtellergeometrie kann als weitere Bedingung ein zylindrischer Bereich in dem sich das Objekt befindet angenommen werden. Als Radius des Zylinders, dessen 'Haupt-Rotationsachse' der Tellerdrehachse entspricht, sei r definiert. Die Abb. 5.2 veranschaulicht die zusätzliche Einschränkung.

Der auf diese Weise eingeschränkte Suchraum stellt eine Strecke bzw. einen Geradenabschnitt im Raum und auch in den weiteren Bildern der Folge dar. Die Endpunkte der Strecke sind die Schnittpunkte der Geraden mit dem Zylindermantel. Neben der so erlangten Beschränkung in Richtung der X- und Y-Achse, kann auch eine minimale Höhe der Z-Achse angegeben werden, die durch die Drehtellerebene (siehe Abb. 5.3) bestimmt ist.

Die Bewegungsbedingung ist wie folgt herzuleiten:

Sei g_i eine Gerade, F_1 der Brennpunkt, B die Brennweite, P_i ein Bildpunkt, r der Radius des Zylinders und Z_{min} die Drehtellerhöhe:

1. Die Bestimmung der Schnittpunkte S_{1i} und S_{2i} von Gerade g_i und Zylindermantel erfolgt durch

$$g_i : \quad F_1 + \lambda(P_i - F_1) = F_1 + \lambda E_i = (X, Y)^T \quad \text{und} \quad r = \sqrt{X^2 + Y^2}$$

¹und natürlich auch in allen weiteren Bildern der Folge

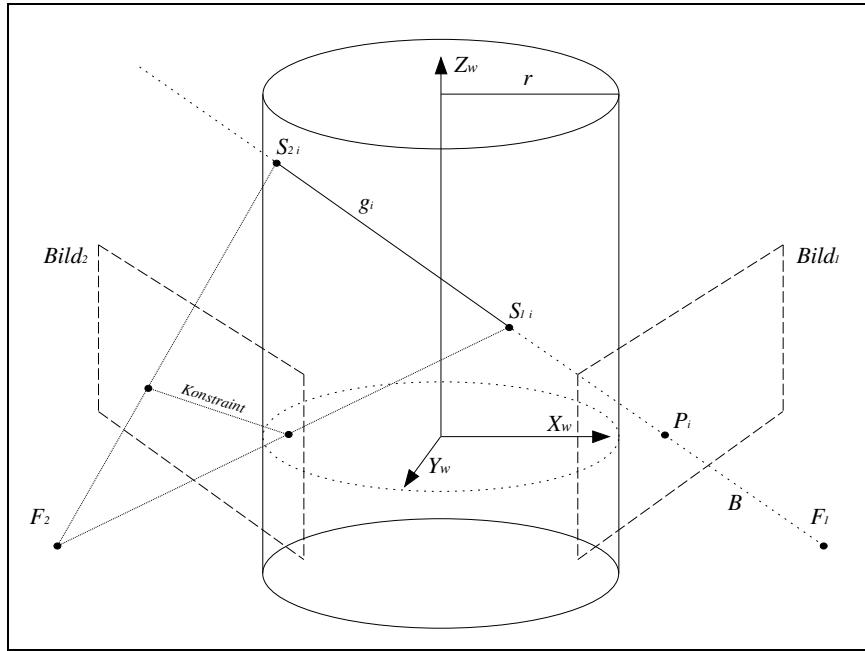


Abbildung 5.2: Zylinderkonstraint

wobei $E_i = P_i - F_1$ den Richtungsvektor der Geraden repräsentiert. Um das gesuchte λ zu berechnen, wird die folgende Gleichung umgeformt:

$$\begin{aligned} r &= \sqrt{(F_{1x} + \lambda E_{x_i})^2 + (F_{1y} + \lambda E_{y_i})^2} \\ r^2 &= (F_{1x} + \lambda E_{x_i})^2 + (F_{1y} + \lambda E_{y_i})^2 \\ r^2 &= F_{1x}^2 + 2\lambda F_{1x} E_{x_i} + (\lambda E_{x_i})^2 + F_{1y}^2 + 2\lambda F_{1y} E_{y_i} + (\lambda E_{y_i})^2 \\ 0 &= \lambda^2(E_{x_i}^2 + E_{y_i}^2) + \lambda(2(F_{1x} E_{x_i} + F_{1y} E_{y_i}) + (F_{1y}^2 + F_{1x}^2 - r^2)) \end{aligned}$$

Die letzte Gleichung ist eine *normale* quadratische Gleichung in λ , die sich direkt lösen lässt. Ihre Ergebnisse sich folgendermaßen zu interpretieren:

- Zwei reelle Lösungen für λ : \Rightarrow zwei Schnittpunkte
 \Rightarrow Korrespondenzsuche auf einem Geradenabschnitt
- Eine reelle Lösungen für λ : \Rightarrow ein Tangentialpunkt am Zylindermantel
 \Rightarrow eindeutiger Korrespondenzpunkt
- Komplexe Lösungen für λ : \Rightarrow die Gerade schneidet den Zylinder nicht
 \Rightarrow ein Punkt außerhalb des Rotationsvolumen
 $\Rightarrow P_i$ ist kein Objektpunkt

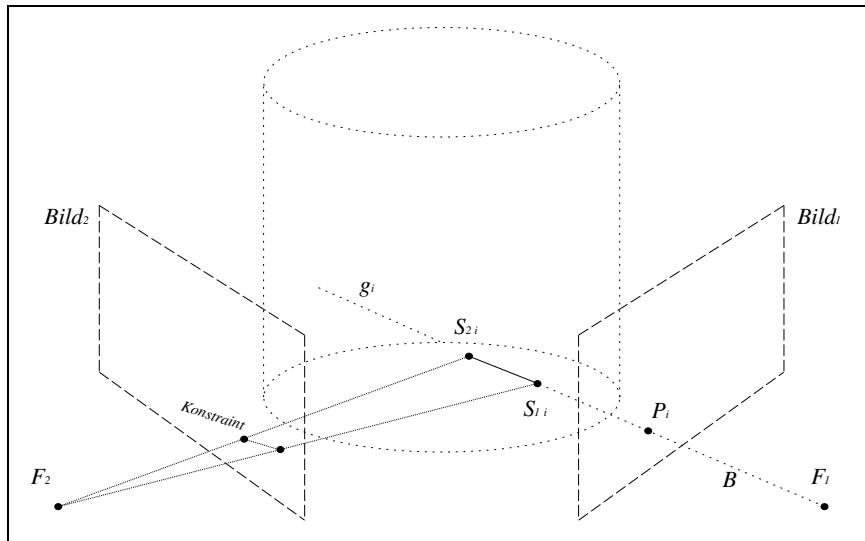


Abbildung 5.3: Motionkonstraint

Nach der Bestimmung von λ können jetzt die Weltkoordinaten der Schnittpunkte $S_{(1,2)i}$ berechnet werden:

$$S_{(1,2)i} = F + \lambda_{(1,2)} E_i$$

2. Die Überprüfung der zulässigen Höhe Z_{min} :

Wenn ein Schnittpunkt auf dem Zylindermantel unterhalb der Drehtellerebene liegt, ist seine Z-Komponente kleiner als Z_{min} . Es gilt folgende Fälle zu unterscheiden:

- Beide Punkte liegen unterhalb : \Rightarrow keine Schnittpunkte
 $\Rightarrow P_i$ ist kein Objektpunkt
- Ein Punkt liegt unterhalb : \Rightarrow neuer Schnittpunkt mit der Drehtellerebene
 \Rightarrow kein, ein oder zwei Schnittpunkte
- Kein Punkt liegt unterhalb : \Rightarrow kein, ein oder zwei Schnittpunkte

Die Berechnung des Schnittpunktes der Geraden g_i mit der Drehtellerebene führt wiederum zur Bestimmung des freien Parameter λ , sie lautet für $g_i := F_1 + \lambda E_i$:

$$\begin{aligned} F_{1z_i} + \lambda E_{z_i} &= Z_{min} \\ \lambda &= \frac{Z_{min} - F_{1z_i}}{E_{z_i}} \end{aligned}$$

Und dann wiederum:

$$S_{(1,2)i} = F_1 + \lambda_{(1,2)} E_i$$

Um das Motionconstraint benutzen zu können, müssen die beiden Schnittpunkte bzw. die sie verbindende Strecke im Weltkoordinatensystem in das zweite Bild projiziert werden. Um diese Koordinatentransformation durchführen zu können, ist die Kenntnis der Kameraposition nach der Rotation des Drehtellers nötig. Ohne dieses Wissen ist der Suchraum nicht mehr auf eine Strecke sondern auf eine Fläche beschränkt. Die Überführung vom Welt- in das Kamerakoordinatensystem der zweiten Aufnahme ist bereits aus Abschnitt 2.2 bekannt:

3. Projektion in das zweite Bild:

$$K_{S_{(1,2)i}} = R R_\Delta S_{(1,2)i} + T$$

mit $K_{S_{(1,2)i}}$ dem Punkt im neuen **KKS** sowie R_Δ der Drehtellerrotation.

Abschließend werden die Schnittpunkte aus dem **KKS** in das zweite Bild projiziert, wobei darauf zu achten ist, daß die entstehende Linie zwischen den beiden Punkten im Bild liegt:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} B \frac{X_K}{Z_K} \\ B \frac{Y_K}{Z_K} \end{pmatrix}$$

Jetzt stehen Einschränkungen, die auf die Drehtellergeometrie abgestimmt wurden, für die Korrespondenzanalyse von Punkten zur Verfügung.

5.3 Ausgezeichnete Verschiebungsvektoren

Durch die für die angestrebte Gestaltsrekonstruktion unzureichenden Ergebnisse der in Abschnitt 5.1 vorgestellten Verfahren, die einfache Geometrie des Drehtellers, die Kenntnis der Rotationsbewegung und durch die Komplexität ihrer Abbildung bedingt, haben wir uns entschlossen zunächst nur für bestimmte Merkmale in den Bildern der Folge die Verschiebungsvektoren zu bestimmen. *Kanten und Eckpunkte* stellen diese ausgewählten Merkmale dar.

Die Eckpunktinformationen werden mittels *Corner Response Funktion* (Abschnitt 4.5) bestimmt, die Kantenmerkmale liefert der *Canny Edge Operator* (Abschnitt 4.3). Wie in 5.2 beschrieben, liegt das Abbild eines Punktes aus dem Bild zum Zeitpunkt t_1 nach der Rotation R_Δ mit dem Winkel ψ_Δ auf eine Strecke im Bild zum Zeitpunkt $t_{i>1}$. Es wird jetzt der Punkt, der der Strecke am nächsten liegt als korrespondierender Punkt ausgewählt, unter der Annahme, daß keine Verdeckungen auftreten und keine Mehrdeutigkeiten existieren. Da die Kantenbilder nur pixelgenaue Informationen liefern, werden die Koordinaten eines Punktes innerhalb einer 8-Nachbarschaft gemittelt. Die Abb. 5.4 demonstrieren diese Vorgehensweise.

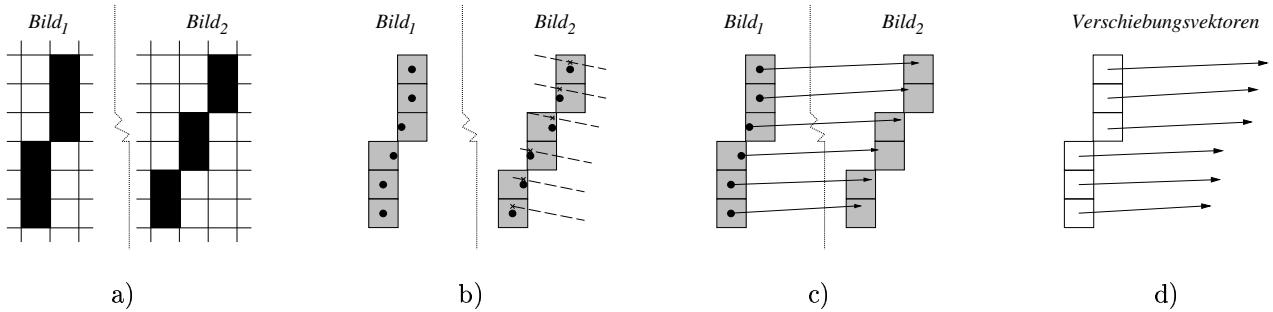


Abbildung 5.4: a) Bildausschnitt korrespondierender Kanten, b) in der 8-Nachbarschaft gemittelte Punktkoordinaten mit eingezeichnetem Motionkonstraint (gestrichelte Linie), c) die aus b) resultierenden Verschiebungsvektoren, d) die Vektoren, die in die das Verschiebungsvektorfeld eingetragen werden.

Probleme

Fehlerhafte Korrespondenzen zwischen Kantenpunkten im ersten und zweiten Bild treten auf, wenn die gesuchte Korrespondenz auf einer Geraden parallel zur Konstraintstrecke liegt, die korrespondierende Kante verdeckt ist bzw. nicht detektiert wurde oder Mehrdeutigkeiten existieren.

Lösungen

Fehlerhafte Zuordnungen bei Auftreten von Mehrdeutigkeiten werden durch die Betrachtung der 8-Nachbarschaft in der Verschiebungsvektorkarte eingeschränkt. Da innerhalb einer 3×3 Maske nur die Punkte einer Kante (bzw. deren V-Vektoren) liegen, kann eine Glattheitsanforderung in Form einer maximalen Abweichung ϱ vom Umgebungsmittelwert definiert werden. Sei \vec{v} der Verschiebungsvektor an der Stelle (x, y) im Bild, dann gilt für Kantenvektoren

$$\|\vec{v}(i, j) - \text{Mean}(\{\vec{v}(x, y) | x \in [i-n, i+n], y \in [j-n, j+n]\})_{[n=1,2,\dots]} \| < \varrho \quad (\text{Glattheitsanforderung}).$$

Eine weitere Möglichkeit ist die Ausnutzung der Bildfolge. Rotationen um die Z-Achse des Drehtellers mit kleinem Winkel ($\psi_\Delta < 10^\circ$) verringern aufgrund kürzerer Konstraintstrecken im zweiten Bild Mehrdeutigkeiten. Andererseits ergeben sich auf diese Weise nur kurze Verschiebungsvektoren, die mit abnehmender

Länge, bedingt durch die Diskretisierung bei der Bestimmung, ungenauer werden. Es ist also wünschenswert möglichst lange Verschiebungsvektoren, zu erhalten, was jedoch bei direkter Berechnung den grund-sätzlichen Matchingproblemen entgegensteht. Eine Lösung ist die sequenzielle Verknüpfung mehrerer Vektorbilder durch Addition (siehe Abb. 5.5).

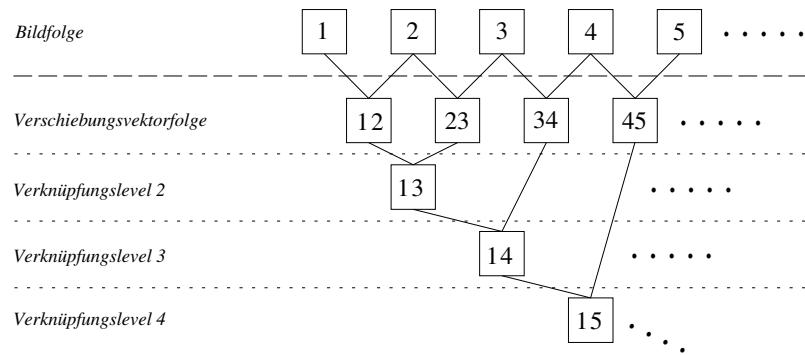


Abbildung 5.5: Die Verknüpfung der Verschiebungsvektorkarten wird durch eine punktweise Vektoraddition durchgeführt und beziehen sich immer auf ein Referenzbild (hier Bild 1).

Die entstehenden Verschiebungsvektorkarten dienen zur Prädiktion im Matchingprozeß über größere Rotationsbewegungen. Die Abb. 5.6 zeigt die ausgezeichneten Verschiebungsvektoren zweier Szenen.

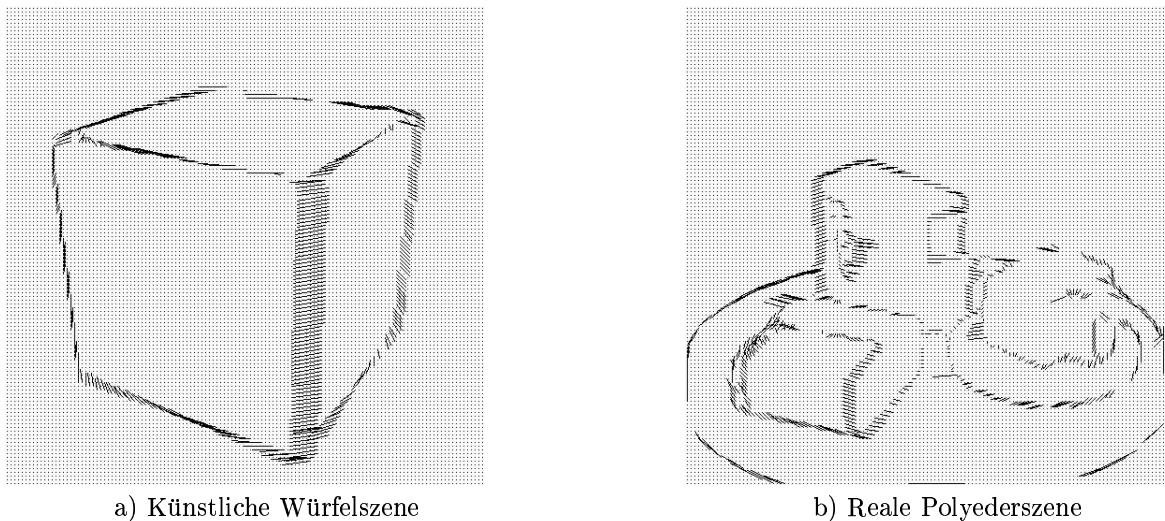


Abbildung 5.6: a) Die Verschiebungsvektorkarten wurden aus vier aufeinanderfolgenden Bildern mit je- weils 5 Grad Drehung kombiniert. b) Es wurden drei Bilder der realen Szene in 10 Grad Schritten aufgezeichnet und die Verschiebungsvektoren ihrer Kantenbilder verknüpft.

5.4 Verschiebungsvektorfelder

Die im Abschnitt 5.3 berechneten Verschiebungsvektorkarten sind nur dünn besetzt und für die angestrebte Gestaltrekonstruktion nicht ausreichend. Es müssen die Verschiebungsvektoren für die Flächen zwischen den Kanten berechnet werden. Da es sich bei diesen Flächen oft um *homogene* Grauwertbereiche handelt, ist eine direkte Berechnung unmöglich. Unter der Annahme von planaren Oberflächen lassen sich die Verschiebungsvektoren *bilinear interpolieren*. Ein Problem, das in diesem Zusammenhang auftritt, ist die Kenntnis welche Punkte zu welcher Fläche gehören. Nicht immer können die Kanten eines Segments vollständig detektiert werden und es somit ausreichend einschränken, es muß daher eine *Segmentierung* durchgeführt werden.

5.4.1 Segmentierung

Die berechneten Bilder der Würfelszene haben ideale Kanten und Flächen und benötigen daher keine Segmentierung. Die aufgenommenen Realszenen wurden interaktiv segmentiert. Versuche mit Segmentierungsalgorithmen, wie z.B. dem Pyramidenverknüpfungsverfahren von [BUR84], führten zu ersten brauchbaren Ansätzen sind jedoch im Rahmen dieser Arbeit nicht weiter verfolgt worden.

5.4.2 Interpolation

Die *bilineare Interpolation* wird zwischen zwei Kantenpunkten durchgeführt, wobei darauf geachtet werden muß, daß der berechnete Verschiebungsvektor innerhalb der Fläche liegt. Sie unterteilt sich in mehrere Prozesse: Zeilenweise Interpolation, spaltenweise Interpolation und eventuell Konsistenzüberprüfung, Nachbarschaftsinterpolation und Glättung. Alle diese Abläufe sind auf Segmente bzw. schon existierende Verschiebungsvektoren beschränkt.

Zeilenweise/spaltenweise Interpolation

Im Idealfall wäre die Interpolation in eine der beiden Richtungen ausreichend. Die aus Kantenbildern gewonnenen Vektoren dienen als Stützpunkte, da die Kantendetektion jedoch nicht immer vollständig ist, kommt es zu Ausfällen und Fehlinterpolationen. Als Lösung dieses Problems wird sowohl zeilenweise als auch spaltenweise bilinear interpoliert.

Konsistenzüberprüfung

Bei der Konsistenzüberprüfung wird innerhalb einer definierten Nachbarschaft der Verschiebungsvektor mit dem Mittelwert aus der Umgebung verglichen. Wenn die Länge des Differenzvektors eine prozentuale Schranke bzgl. der Länge des Mittelwerts überschreitet, wird der Verschiebungsvektor abgelehnt bzw. gelöscht. Dies entspricht einer Überprüfung der zuvor definierten Glattheitsanforderung.

Nachbarschaftsinterpolation, Glättung

Beide Prozesse arbeiten wiederum innerhalb einer definierten Nachbarschaft (z.B 8-Nachbarschaft). Sie unterscheiden sich lediglich darin, daß bei der Glättung der untersuchte Vektor schon existieren muß und in die Mittelwertberechnung einfließt, bei der Interpolation jedoch fehlt und neu bestimmt wird.

Zwei Ergebnisse dieser Interpolation werden in der nachfolgenden Abb. 5.7 dargestellt. In beiden Fällen wurden alle Prozesse durchlaufen.

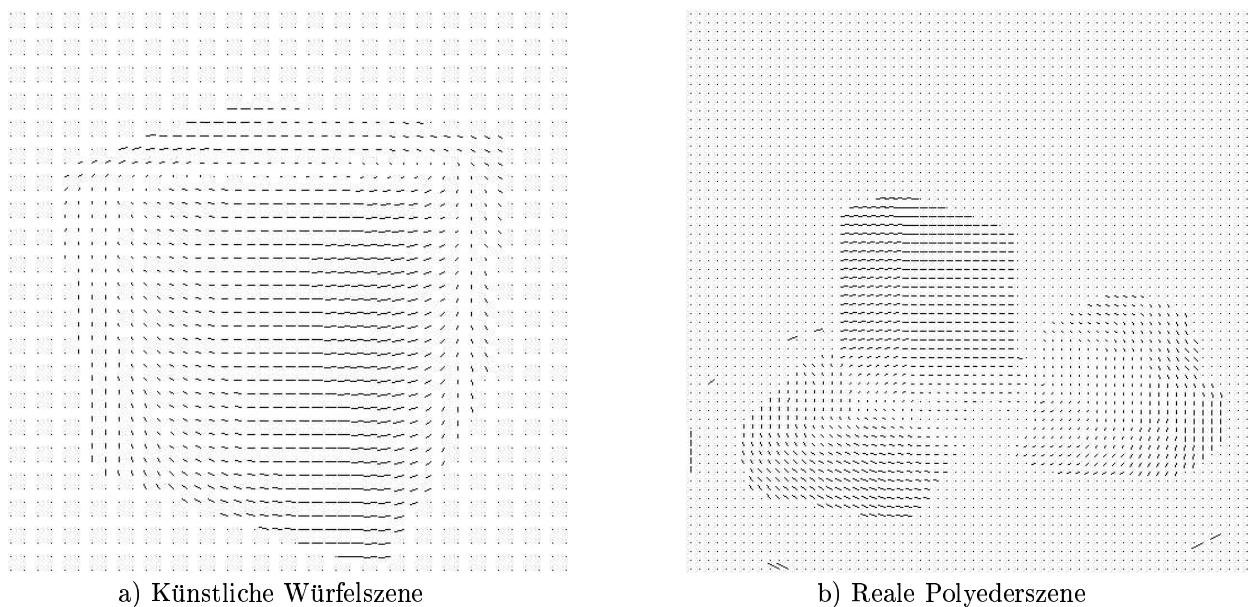


Abbildung 5.7: Bilinear interpolierte Verschiebungsvektorfelder der Kantenvektoren aus Abb. 5.6

Literaturverzeichnis

- [ANA87] **Anandan P.**: PhD Dissertation, COINS TR 87-21, University of Massachusetts, Amherst, 1987.
- [ANA89] **Anandan P.**: International Journal of Computer Vision, pp. 238-310, 1989.
- [BAR92] **Barron, J. und Fleet D.**: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 236-242, 1992.
- [BUR84] **Burt P.J.**: The Pyramid as a structure for efficient computation, In: Rosenfeld, A. (Ed.): Multiresolution image processing and analysis, Berlin, Springer, 1984.
- [FLE90] **Fleet D. und Jepson A.**: International Journal of Computer Vision, pp. 77-104, 1990.
- [FLE92] **Fleet D.**: *Measurement of Image Velocity* Kluwer Academic Publ., Norwell, 1992.
- [HEE88] **Heeger D.**: International Journal of Computer Vision, pp. 279-302, 1988.
- [HOR81] **Horn B. und Schunk B.**: Artificial Intelligence 17, pp. 185-204, 1981.
- [LUC84] **Lucas, B.**: PhD Dissertation, Dept. Computer Science, Carnegie-Mellon University, 1984.
- [LUC81] **Lucas, B. und Kanade, T.**: Proceedings DARPA IU Workshop, pp. 121-130, 1981.
- [NAG83] **Nagel H.**: Computer Graphics and Image Processing 21, pp. 85-117, 1983.
- [URA88] **Uras S., Girosi F., Verri A. und Torre V.**: Biological Cybernetics 60, pp. 79-97, 1988.

Kapitel 6

Gestaltsrekonstruktion

6.1 Tiefe aus Bewegung

In den nachfolgenden Abschnitten wird das Problem der Tiefenberechnung behandelt. *Motion* liegt hier in der Form von Bildpunkt-Verschiebungsvektoren vor, die in ein Gleichungssystem, dessen Lösung die Tiefe der entsprechenden Punkte ergibt, eingesetzt werden. Die folgenden Fallunterscheidungen werden behandelt:

- Überbestimmtes Gleichungssystem mit Kenntnis des Rotationswinkels
- Gleichungssystem ohne Kenntnis des Rotationswinkels

6.1.1 Überbestimmtes Gleichungssystem mit Kenntnis der Rotation

Für die Transformation eines Punktes W vom Weltkoordinatensystem **WKS** in einen Punkt K des Kamerakoordinatensystems **KKS** liegt folgende Gleichung vor:

$$RW + T = K$$

mit R als 3×3 Rotationsmatrix und T als Translationsvektor¹. Für einen Punkt W im **WKS** gilt nach einer Rotation R_Δ des Drehtellers:

$$\begin{array}{lll} \text{voher:} & RW + T & = K_1 \\ \text{nachher:} & RR_\Delta W + T & = K_2 \end{array}$$

K_1 ist der Punkt im **KKS** zum Zeitpunkt t_1 und K_2 zum Zeitpunkt t_2 nach der Rotation. Diese beiden Gleichungen lassen sich wie folgt umformen und kombinieren:

$$\begin{array}{llll} \begin{array}{lll} K_1 & = & RW + T & \parallel RR_\Delta W + T & = & K_2 \\ K_1 - T & = & RW & RR_\Delta W & = & K_2 - T \\ R^T(K_1 - T) & = & W & W & = & (RR_\Delta)^T(K_2 - T) \end{array} \\ \\ \underbrace{\begin{array}{lll} R^T(K_1 - T) & = & (RR_\Delta)^T(K_2 - T) \\ R^T K_1 - R^T T & = & (RR_\Delta)^T K_2 - (RR_\Delta)^T T \\ \underbrace{(RR_\Delta)^T T - R^T T}_{\vec{b}} & = & (RR_\Delta)^T K_2 - R^T K_1 \\ \vec{b} & = & (RR_\Delta)^T K_2 - R^T K_1 \end{array}}_{\vec{b}} \end{array}$$

¹sowohl R als auch T sind während der Kamerakalibrierung bestimmt worden

Aufgrund der kamerazentrierten Perspektivprojektion mit der Brennweite B ergibt sich für einen *idealen* Bildpunkt (X_{P_i}, Y_{P_i}) zum Zeitpunkt t_i , $i \geq 1$,

$$X_{P_i} = \frac{X_{K_i}B}{Z_{K_i}} \quad Y_{P_i} = \frac{Y_{K_i}B}{Z_{K_i}}$$

bzw. umgekehrt für den entsprechenden Kamerapunkt

$$X_{K_i} = \frac{X_{P_i}Z_{K_i}}{B} \quad Y_{K_i} = \frac{Y_{P_i}Z_{K_i}}{B}$$

Diese Gleichungen gelten nur für die *idealen, unverzerrten* Punktkoordinaten (X_{P_i}, Y_{P_i}) in der Bildebene. Für verzerrte Bildpunktkoordinaten (X_{D_i}, Y_{D_i}) müssen jedoch die folgenden Entzerrungsgleichungen (siehe Abschnitt 2.3) angewendet werden:

$$X_{P_i} = X_{D_i}(1 + \kappa_1 r^2 + \kappa_2 r^4), \quad Y_{P_i} = Y_{D_i}(1 + \kappa_1 r^2 + \kappa_2 r^4) \quad \text{und} \quad r = \sqrt{X_{D_i}^2 + Y_{D_i}^2}.$$

Mit κ_1 und κ_2 als Distorsionskoeffizienten können die auf diese Weise entzerrten Bildpunkte P_i zur Berechnung der Punkte K_i im **KKS** herangezogen werden:

$$K_i = \begin{pmatrix} X_{K_i} \\ Y_{K_i} \\ Z_{K_i} \end{pmatrix} = \begin{pmatrix} \frac{X_{P_i}Z_{K_i}}{B} \\ \frac{Y_{P_i}Z_{K_i}}{B} \\ Z_{K_i} \end{pmatrix} = Z_{K_i} \begin{pmatrix} \frac{X_{P_i}}{B} \\ \frac{Y_{P_i}}{B} \\ 1 \end{pmatrix} = Z_{K_i} E_i$$

E_i stellt eine abkürzende Schreibweise für den entsprechenden Vektor dar. Unter Ausnutzung der Ausdrücke

$$\vec{b} = (RR_\Delta)^T K_2 - R^T K_1 \quad \text{und} \quad K_i = Z_{K_i} E_i$$

liegt eine Vektorgleichung bzw. ein Gleichungssystem bestehend aus drei Gleichungen mit den beiden Unbekannten Z_{K_2} und Z_{K_1} vor:

$$\vec{b} = (RR_\Delta)^T Z_{K_2} E_2 - R^T Z_{K_1} E_1$$

$$\vec{b} = Z_{K_2} \underbrace{(RR_\Delta)^T E_2}_{\vec{c}} - Z_{K_1} \underbrace{R^T E_1}_{\vec{d}}$$

$$\vec{b} = Z_{K_2} \vec{c} - Z_{K_1} \vec{d}$$

$$\vec{b} = \underbrace{\left(\vec{c} \ (-\vec{d}) \right)}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} Z_{K_2} \\ Z_{K_1} \end{pmatrix}}_{\vec{z}}$$

$$\vec{b} = \mathbf{A} \cdot \vec{z}$$

Dieses *überbestimmte lineare Gleichungssystem* kann mit Hilfe der Pseudoinversen und *Pivoting* numerisch stabil berechnet werden:

$$\begin{aligned} \vec{b} &= \mathbf{A} \vec{z} \\ \mathbf{A}^T \vec{b} &= \mathbf{A}^T \mathbf{A} \vec{z} \\ (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \vec{b} &= \vec{z} \end{aligned}$$

Für gegebene R, T (Kalibrierung) und R_Δ (bekannte Rotation der Drehscheibe) sind also zunächst die Werte Z_{K_1} und Z_{K_2} (d.h. Vektor \vec{z}) aus den Abbildpunkten E_1 und E_2 zu berechnen. Die Kalibrierungsparameter B, κ_1, κ_2 werden zur Berechnung benötigt. Die entstehenden Lösungen sind die Tiefenwerte im Kamerakoordinatensystem und müssen noch ins Weltkoordinatensystem transformiert werden. Die beiden Punkte K_1 und K_2 (d.h. $Z_{K_1}E_1$ und $Z_{K_2}E_2$) stellen den gleichen Weltpunkt W dar. Sie können folgendermaßen rückprojiziert und zur Kontrolle der Genauigkeit miteinander verglichen werden:

$$W = R^T(Z_{K_1}E_1 - T) = (RR_\Delta)^T(Z_{K_2}E_2 - T).$$

6.1.2 Gleichungssystem ohne Kenntnis des Rotationswinkel

Bei der Betrachtung der Tiefenbestimmung unter Kenntnis der Rotation führten die Überlegungen zu einem überbestimmten Gleichungssystem bestehend aus 3 Gleichungen mit zwei Unbekannten. Durch das ‘Mehr’ an Informationen in diesem System motiviert, versuchen wir jetzt eine weitere Unbekannte einzuführen: Den Rotationswinkel ψ_Δ der Rotation R_Δ .

Bei flüchtiger Betrachtung des Gleichungssystems stellt man fest, daß es für die drei Variablen Z_{K_1}, Z_{K_2} und ψ_Δ seine Gestalt und Linearität verliert. Um dieses Problem zu bewältigen, muß ein neuer Ansatz gefunden werden. Im folgenden werden zwei Lösungen dargestellt, die jedoch beide ihre Schwächen haben.

Die neuen Lösungswwege sind zu Beginn dem Vorhergehenden sehr ähnlich. Für die Transformation eines Punktes W vom Weltkoordinatensystem **WKS** in einen Punkt K des Kamerakoordinatensystems **KKS** liegen folgende Gleichungen mit R als 3×3 Rotationsmatrix und T als Translationsvektor sowie einer Rotation R_Δ des Drehellers vor (siehe Abschnitt 6.1.1):

$$\begin{array}{lll} \text{voher:} & RW + T & = K_1 \\ \text{nachher:} & RR_\Delta W + T & = K_2 \end{array}$$

wobei K_1 der Punkt im **KKS** zum Zeitpunkt t_1 und K_2 zum Zeitpunkt t_2 nach der Rotation R_Δ ist. Diese beiden Gleichungen lassen sich wiederum wie folgt umformen und kombinieren:

$$\begin{array}{llll} K_1 & = & RW + T & = K_1 \\ K_1 - T & = & RW & = K_1 - T \\ R^T(K_1 - T) & = & W & = R_\Delta^T R^T(K_2 - T) \\ \\ R^T(K_1 - T) & = & R_\Delta^T R^T(K_2 - T) \\ R^T K_1 - R^T T & = & R_\Delta^T (R^T K_2 - R^T T) & \end{array}$$

Auch hier ergibt sich für einen Punkt K_i im Kamerakoordinatensystem aufgrund der kamerazentrierten Perspektivprojektion mit der Brennweite B und einem Punkte (X_{P_i}, Y_{P_i}) im Bild

$$K_i = \begin{pmatrix} X_{K_i} \\ Y_{K_i} \\ Z_{K_i} \end{pmatrix} = \begin{pmatrix} \frac{X_{P_i}Z_{K_i}}{B} \\ \frac{Y_{P_i}Z_{K_i}}{B} \\ \frac{1}{Z_{K_i}} \end{pmatrix} = Z_{K_i} \begin{pmatrix} \frac{X_{P_i}}{B} \\ \frac{Y_{P_i}}{B} \\ 1 \end{pmatrix} = Z_{K_i} E_i$$

wobei E_i eine abkürzende Schreibweise für den entsprechenden Vektor darstellt.

Die Linsenverzerrung kann, wie im vorhergehenden Verfahren mit Kenntnis des Rotationswinkels, an dieser Stelle einbezogen werden, wird jedoch aus Gründen der Übersicht im folgenden weggelassen. Nach einigen Vereinfachungen ergibt sich:

$$\begin{aligned} \underbrace{Z_{K_1}}_{z_1} \underbrace{R^T E_1}_{\vec{a}} - \underbrace{R^T T}_{\vec{c}} &= R_\Delta^T (\underbrace{Z_{K_2}}_{z_2} \underbrace{R^T E_2}_{\vec{b}} - \underbrace{R^T T}_{\vec{c}}) \\ z_1 \vec{a} - \vec{c} &= R_\Delta^T (z_2 \vec{b} - \vec{c}) \end{aligned}$$

Die zuvor erwähnte Problematik stellt die Rotationsmatrix R_Δ dar. Sie ist zwar nicht vollständig besetzt - es handelt sich *nur* um eine Rotation um die Z-Achse - jedoch kommt der unbekannte Winkel ψ_Δ mehrfach kombiniert in trigonometrischen Funktionen vor.

1. Direkte Lösung

Die inverse Rotationsmatrix um die Z-Achse hat folgende Gestalt:

$$R_\Delta^T = \begin{pmatrix} \cos(\psi_\Delta) & -\sin(\psi_\Delta) & 0 \\ \sin(\psi_\Delta) & \cos(\psi_\Delta) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Mit der zuvor hergeleiteten Gleichung

$$z_1 \vec{a} - \vec{c} = R_\Delta^T (z_2 \vec{b} - \vec{c})$$

ergibt sich das Gleichungssystem:

$$z_1 a_x - c_x = (z_2 b_x - c_x) \cos(\psi_\Delta) - (z_2 b_y - c_y) \sin(\psi_\Delta) \quad (1)$$

$$z_1 a_y - c_y = (z_2 b_x - c_x) \sin(\psi_\Delta) + (z_2 b_y - c_y) \cos(\psi_\Delta) \quad (2)$$

$$z_1 a_z - c_z = z_2 b_z - c_z \quad (3)$$

Die dritte Gleichung nach z_2 aufgelöst, in (1) und (2) eingesetzt, ergibt:

$$\begin{aligned} z_1 a_x - c_x &= \left(\frac{z_1 a_z b_x}{b_z} - c_x \right) \cos(\psi_\Delta) - \left(\frac{z_1 a_z b_y}{b_z} - c_y \right) \sin(\psi_\Delta) \\ z_1 a_y - c_y &= \left(\frac{z_1 a_z b_x}{b_z} - c_x \right) \sin(\psi_\Delta) + \left(\frac{z_1 a_z b_y}{b_z} - c_y \right) \cos(\psi_\Delta) \end{aligned}$$

Das entstandene Gleichungssystem mit den beiden Unbekannten ψ_Δ und z_1 lässt sich eindeutig lösen. Die Lösungen² lauten:

$$\begin{aligned} \psi_\Delta &= 2 \arctan \left(\frac{c2(a_y b_z - b_y a_z) + c1(a_x b_z - b_x a_z)}{c2(a_x b_z + b_x a_z) - c1(a_y b_z + b_y a_z)} \right) \\ z_1 &= \frac{b_z(c_x - c_x \cos(\psi_\Delta) + c_y \sin(\psi_\Delta))}{a_x b_z - a_z b_x \cos(\psi_\Delta) + a_z b_y \sin(\psi_\Delta)} \\ z_2 &= z_1 \frac{a_z - c_z}{b_z - c_z} \end{aligned}$$

Mit diesen Lösungen können jetzt die Weltkoordinaten des Punktes W wie im Abschnitt zuvor berechnet werden:

$$W = R^T(z_1 E_1 - T) = (R R_\Delta)^T(z_2 E_2 - T).$$

²Die Lösungen wurden mit *MapleV*[®], einem mathematischen Symbolmanipulationsprogramm, errechnet.

2. Zylinderkoordinatensystem-Lösung

Eine weitere Möglichkeit ist die Betrachtung der Rotation in einer anderen Koordinatendarstellung. Für die spezifische Bewegung um die Z-Achse des Drehtellers bietet sich der Wechsel in das *Zylinderkoordinatensystem* an. Es ist wie folgt definiert [BRO87]:

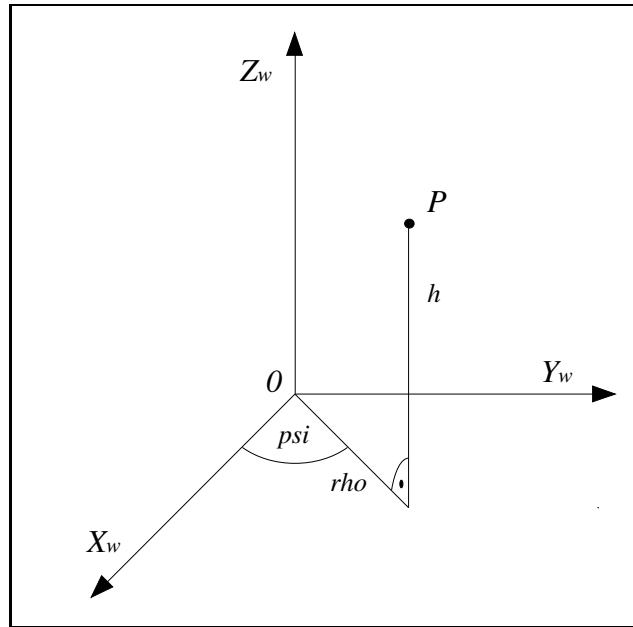


Abbildung 6.1: Kartesisches- und Zylinderkoordinatensystem

Ein Punkt ist demnach durch (ψ, ρ, h) mit h als Höhe und ψ als Winkel zwischen X-Achse und ρ eindeutig beschrieben. Für die Umrechnung dieser beiden Koordinatensysteme ineinander gilt folgendes:

$$\begin{aligned} x &= \rho \cos \psi \\ y &= \rho \sin \psi \\ z &= h \\ \rho &= \sqrt{x^2 + y^2} \\ \psi &= \arctan(\frac{y}{x}) \end{aligned}$$

Der Vorteil dieser Darstellung ist, daß sich die spezielle Rotationsmatrix R_Δ um die Z-Achse in eine Vektoraddition verwandelt. Sei P ein Punkt, dann gilt in diesem Fall:

$$R_\Delta \cdot P_{\text{kartesisch}} = P_{\text{zylindrisch}} + \begin{pmatrix} \psi_\Delta \\ 0 \\ 0 \end{pmatrix}$$

Auf die zuvor hergeleitete Gleichung bezogen:

- kartesisch:

$$z_1 \vec{a} - \vec{c} = R_\Delta^T (z_2 \vec{b} - \vec{c})$$

- zylindrisch:

$$\begin{pmatrix} \arctan\left(\frac{z_1 a_y - c_y}{z_1 a_x - c_x}\right) \\ \sqrt{(z_1 a_x - c_x)^2 + (z_1 a_y - c_y)^2} \\ (z_1 a_z - c_z) \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{z_2 b_y - c_y}{z_2 b_x - c_x}\right) \\ \sqrt{(z_2 b_x - c_x)^2 + (z_2 b_y - c_y)^2} \\ (z_2 b_z - c_z) \end{pmatrix} + \begin{pmatrix} -\psi_\Delta \\ 0 \\ 0 \end{pmatrix}$$

daraus ergibt sich das folgende vereinfachte Gleichungssystem:

$$z_1 a_z = z_2 b_z \quad (1)$$

$$(z_1 a_x - c_x)^2 + (z_1 a_y - c_y)^2 = (z_2 b_x - c_x)^2 + (z_2 b_y - c_y)^2 \quad (2)$$

$$\psi_\Delta = \arctan\left(\frac{z_2 b_y - c_y}{z_2 b_x - c_x}\right) - \arctan\left(\frac{z_1 a_y - c_y}{z_1 a_x - c_x}\right) \quad (3)$$

Aus den ersten beiden Gleichungen lässt sich z_1 und z_2 bestimmen. Die erste Gleichung wird nach z_2 aufgelöst,

$$z_2 = z_1 \frac{a_z}{b_z},$$

dies in die zweite Gleichung eingesetzt und nach z_1 umgeformt:

$$\begin{aligned} (z_1 a_x - c_x)^2 + (z_1 a_y - c_y)^2 &= (z_1 b_x \frac{a_z}{b_z} - c_x)^2 + (z_1 b_y \frac{a_z}{b_z} - c_y)^2 \\ (z_1 a_x)^2 + (z_1 a_y)^2 - 2z_1 a_x c_x - 2z_1 a_y c_y &= (z_1 b_x \frac{a_z}{b_z})^2 + (z_1 b_y \frac{a_z}{b_z})^2 - 2z_1 b_x \frac{a_z}{b_z} c_x - 2z_1 b_y \frac{a_z}{b_z} c_y \\ z_1^2 (a_x^2 + a_y^2 - (b_x \frac{a_z}{b_z})^2 - (b_y \frac{a_z}{b_z})^2) &= 2z_1 (a_x c_x + a_y c_y - b_x \frac{a_z}{b_z} c_x - b_y \frac{a_z}{b_z} c_y) \\ z_1 &= 2 \frac{(a_x c_x + a_y c_y - b_x \frac{a_z}{b_z} c_x - b_y \frac{a_z}{b_z} c_y)}{(a_x^2 + a_y^2 - (b_x \frac{a_z}{b_z})^2 - (b_y \frac{a_z}{b_z})^2)} \end{aligned}$$

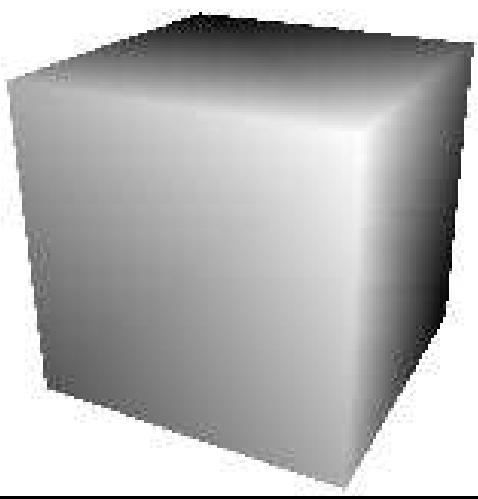
Mit den Ergebnissen für z_1 und z_2 ergibt sich aus der dritten Gleichung der gesuchte Winkel ψ_Δ . Der Weltpunkt wird wie im letzten Abschnitt berechnet.

6.2 Praxis

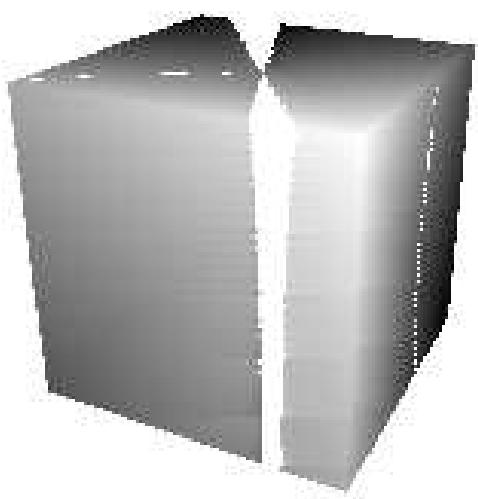
In der Praxis hat sich das Verfahren mit Kenntnis des Rotationswinkels als ‘stabil’ herausgestellt, ganz im Gegensatz zu den beiden Verfahren ohne Rotationswinkel. Sowohl die ‘direkte’ als auch die ‘Zylinderkoordinatensystem-Lösung’ reagieren in bestimmten Bereichen instabil und produzieren unbrauchbare Lösungen (siehe Abbildung 6.2).

Im zweiten Lösungsansatz treten die Ausfälle in der Tiefenkarte immer auf, wenn die Richtung der Verschiebungsvektoren von Punkten horizontal bzw. vertikal bezüglich des Bildes ist. Das heißt wenn der Quotient $\frac{u}{v} \gg 1$ oder $\frac{u}{v} \ll 1$ ist ((u, v) = Verschiebungsvektor im Bild). Untersuchungen der Gleichungen konnten jedoch keinerlei rechentechnische Instabilität aufzeigen.

Eine Lösung dieses Problems ist die Kombination der Verfahren *ohne* und *mit* Rotationskenntnis. Über die ‘direkte Lösung’ wird der Rotationswinkel ψ_Δ für alle Verschiebungsvektoren berechnet. Aufgrund



ideale Tiefenkarte zum Vergleich



berechnete Tiefenkarte nach dem 2. Ansatz

Abbildung 6.2: Tiefenberechnung ohne Kenntnis der Rotation

der Drehtellergeometrie wird jeder Objektpunkt mit dem gleichen Winkel rotiert, was eine Mittelwertbildung ermöglicht. Anhand dieses Mittelwertes und der Varianz werden die ψ_{Δ} , die innerhalb einer Epsilonumgebung mit der Größe der Standartabweichung σ um den Mittelwert $Mean_1$ liegen, erneut zur Mittelung herangezogen.

$$\psi_{\Delta} = Mean_2(\{\psi_{i\Delta} \mid \|Mean_1(\{\psi_{j\Delta} \mid j \text{ ist Objektpunkt}\}) - \psi_{i\Delta}\| \leq \sigma \text{ und } i \text{ ist Objektpunkt}\})$$

Mit dieser Näherung des Winkels kann jetzt das numerisch stabilere Verfahren in Abschnitt 6.1.1 angewendet werden.

Kapitel 7

Ergebnisse

7.1 Gestaltsanalyse mit synthetischen Verschiebungsvektoren

Da die Verwendung von bereits existierenden Programmen zur Bestimmung des optischen Flußes nicht befriedigend verlief, haben wir vorerst synthetische Verschiebungsvektoren von einem *Simulator* generieren lassen. Dabei kann ein Polyeder eingelesen werden und unter Perspektivprojektion aus einem beliebigen Winkel betrachtet werden (siehe Abb. 7.1). Die Kalibrierungsdaten sind somit bekannt. Der Polyeder kann nun um die Z-Achse des Objekts gedreht werden, die Verschiebung im Bild bestimmt und anschließend als Verschiebungsvektorkarte gespeichert werden (vgl. Abb. 7.2).

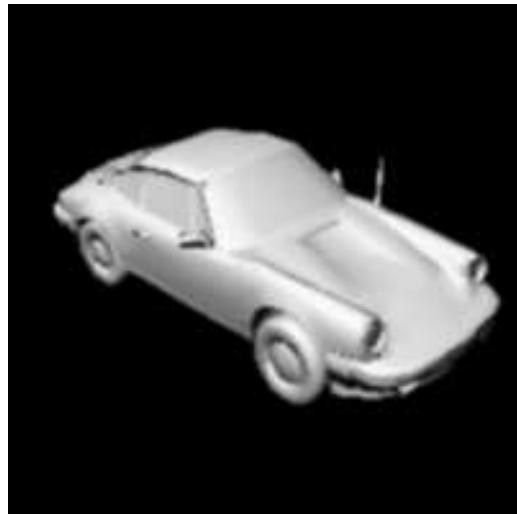
Die Bestimmung der Gestalt kann bei bekanntem Drehwinkel ohne wesentlichen Verlust durchgeführt werden, wie die rekonstruierte Tiefenkarte in Abb. (7.3) zeigt. Das unterstreicht die Seitenansicht des rekonstruierten Polyeders in Abb. (7.4), wo die korrekte Tiefenbestimmung besonders deutlich wird. Für den direkten Vergleich mit dem Grauwertbild ist dieser Polyeder in Abb. (7.5) mit einem Beleuchtungsmodell schattiert worden. Die dichten und optimalen Verschiebungsvektoren erlauben also eine *ideale Berechnung* der Tiefe.

Um die Bedingungen realistischer zu gestalten, kann die Bestimmung der Verschiebungsvektoren gestört werden (vgl. Abb. 7.6). Die einzelnen Komponenten des Vektors werden dabei unabhängig von einem Zufallsgenerator mit normalverteiltem Rauschen unterlegt. Die für Abb. (7.7) und (7.8) erzeugten Fehler sind in der Tabelle (7.1) aufgelistet.

Störung der Verschiebungsvektoren		
Standardabweichung σ	0.15	0.5
Durchschnittlicher Fehler der Orientierung	Vektorlänge	0.188 mm
	Orientierung	5.560 grd
Maximale Abweichung der Orientierung	Vektorlänge	0.657 mm
	Orientierung	178.492 grd
		16.249 grd
		2.191 mm
		179.732 grd

Tabelle 7.1: Rauschen bei der synthetischen Verschiebungsvektorbestimmung.

Es hat sich herausgestellt, daß eine aus diesen Verschiebungsvektoren rekonstruierte Tiefenkarte durch eine Gaußglättung wesentlich verbessert werden kann. Die Grundzüge der Oberflächengestalt können bei der in Abb. (7.8) dargestellten Störung mit einer Glättung der Tiefenkarte wiederhergestellt werden (vgl. Abb. 7.9).



Ein Sportwagen.



Büste von Beethoven.

Abbildung 7.1: Grauwertbilder von synthetisch generierten Objekten.

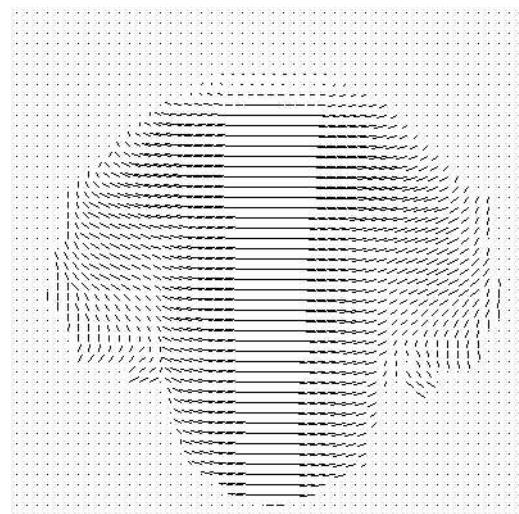
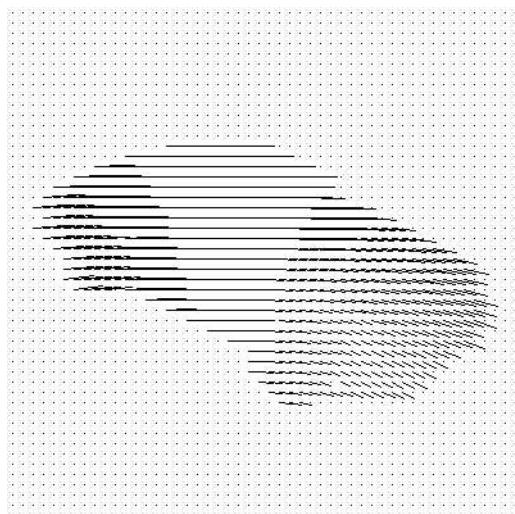


Abbildung 7.2: Durch Simulation bestimmte ideale Verschiebungsvektoren.

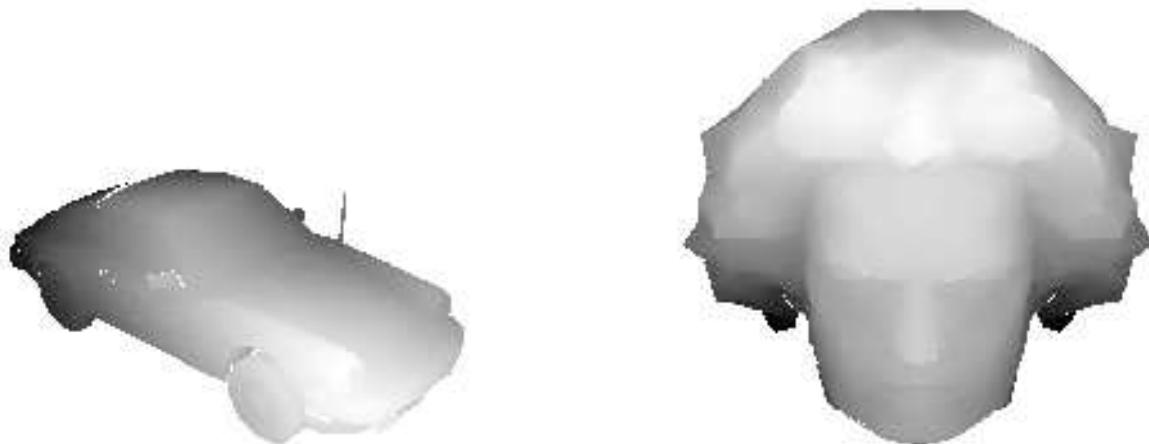


Abbildung 7.3: Aus idealen Verschiebungsvektoren rekonstruierte Tiefenkarte.

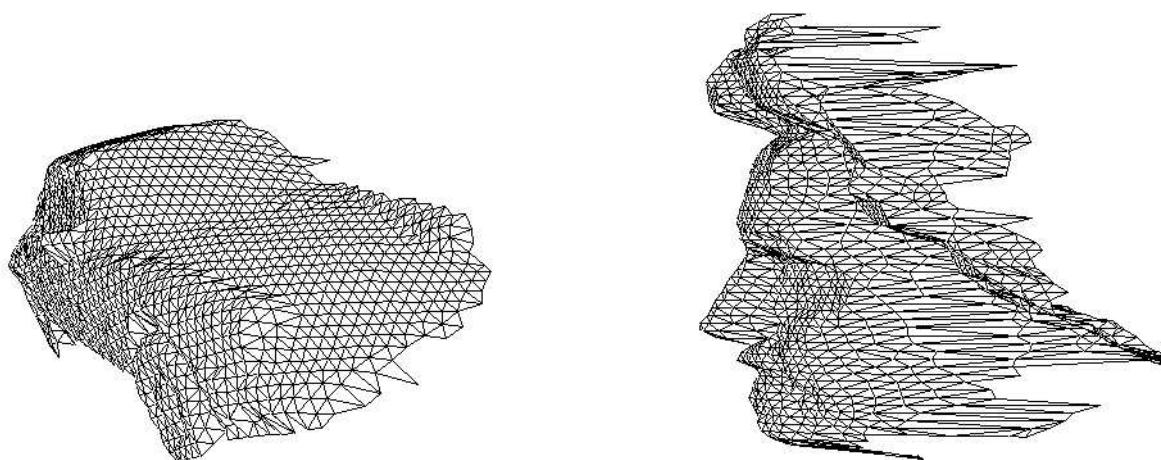


Abbildung 7.4: 3D-Visualisierung der rekonstruierten Tiefenkarte.

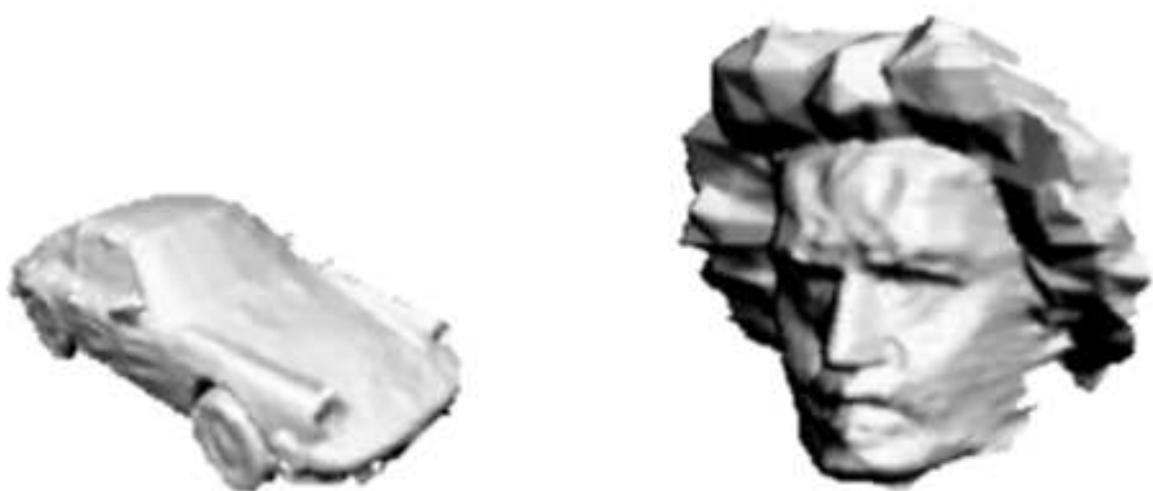


Abbildung 7.5: Das Ergebnis der Gestaltserkennung bei idealen Verschiebungsvektoren.

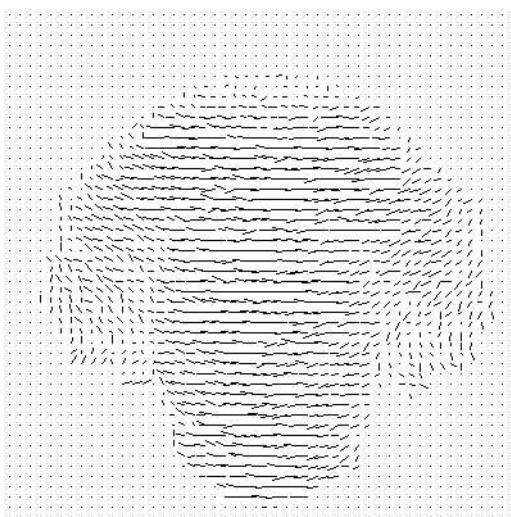


Abbildung 7.6: Störung der Verschiebungsvektoren mit Standardabweichung $\sigma = 0.5$.

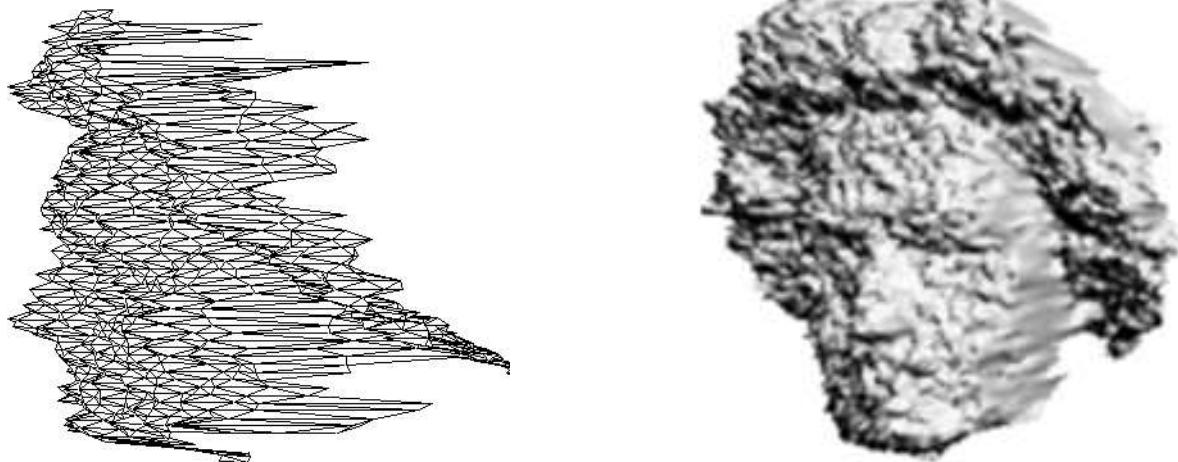


Abbildung 7.7: Rekonstruktion bei verrauschten Verschiebungsvektoren mit $\sigma = 0.15$.

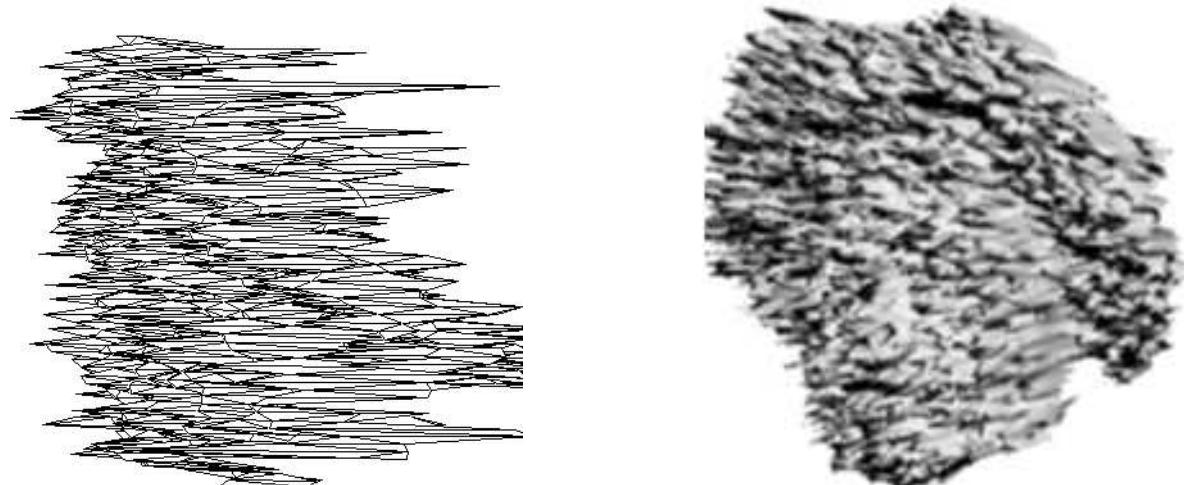


Abbildung 7.8: Rekonstruktion mit gestörten Verschiebungsvektoren bei $\sigma = 0.5$.

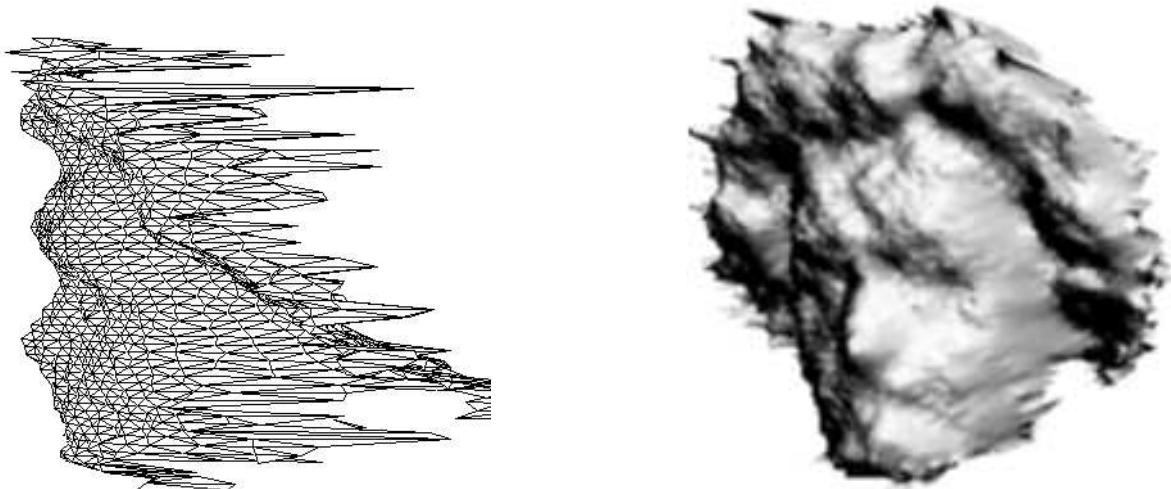


Abbildung 7.9: Ergebnis nach Gaußglättung der Tiefenkarte mit einer 7×7 Maske.

7.2 Rekonstruktion der Gestalt mit Optical-Flow-Techniken

Als Grundlage für diese Untersuchungen lagen uns die Implementationen der von *Barron und Fleet* [BAR92] beschriebenen Verfahren vor. Für den Einsatz dieser Optical-Flow-Techniken zur Bestimmung der Verschiebungsvektoren sollen zunächst synthetische Grauwertbilder eines einfachen polyedrischen Objekts betrachtet werden. Dazu wird ein Würfel zusammen mit der Lichtquelle um 1 *grd* gedreht, um eine ideale und rauschfreie Grauwertzuordnung zu ermöglichen. In Abb. (7.10) sind typische Ergebnisse bei der Anwendung der Optical-Flow-Techniken dargestellt.

Das Verfahren von *Anandan* [ANA89] detektiert für zwei Eingabebilder (200×200 *pix*) zwar eine Verschiebung an Grauwertkanten, aber die Richtung ist nur selten korrekt bestimmt. Die Breite und Kontinuität der berechneten Verschiebungsvektoren verdeutlicht die starke Glättung bei dieser Technik. Für das differentielle Verfahren von *Lucas und Kanade* [LUC81] wurden 15 Bilder generiert. Die resultierende Verschiebungsvektorkarte ist allerdings unzureichend dünn besetzt.

Das Problem der geringen Dichte entsteht gerade durch homogene Fächen, da nur an Grauwertkanten Korrespondenzen gefunden werden können. Um auch in diesen Gebieten Vektoren bestimmen zu können, wurden die synthetischen Objekte mit verschiedenen Texturen überzogen (vgl. Abb. 7.11).

Daß die Anzahl der Verschiebungsvektoren dadurch tatsächlich erheblich gesteigert werden kann, beweist die Abb. (7.12). Während bei dem Verfahren von *Anandan* die Qualität noch unzureichend ist, zeigt das Verfahren von *Lucas und Kanade* bei dem texturierten Würfel wenigstens bei einer Fläche seine Fähigkeiten.

Das in Abb. (7.13) dargestellte Beispiel, wo die Gestalt einer realen Büste rekonstruiert wird, ist nicht repräsentativ für die Gestaltanalyse mit Optical-Flow-Techniken. Nur das Verfahren von *Anandan* berechnete mit einer ganz speziellen Parameterkombination das vorliegende Resultat. Diese Einstellungen konnten nicht konstruktiv ermittelt werden und ergaben sich aus einer umfassenden Testreihe.

Für Eingabebilder der Größe 512×512 *pix* wurde eine Pyramide mit 4 hierarchischen Ebenen gewählt und das 3×3 Korrelationsfenster wurde über 10 Relaxationsschritte angewendet. Die auf diese Weise experimentell gewonnenen Parameter waren nicht auf andere Aufnahmen übertragbar. So ist das reale Beispiel in Abb. (7.13) eher als optimistische Grenze der untersuchten Optical-Flow Verfahren anzusehen.

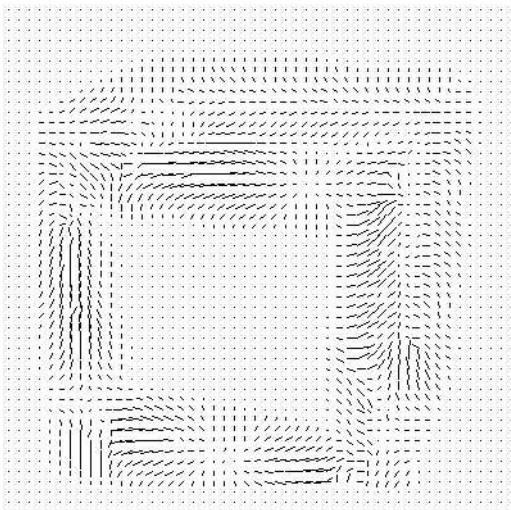
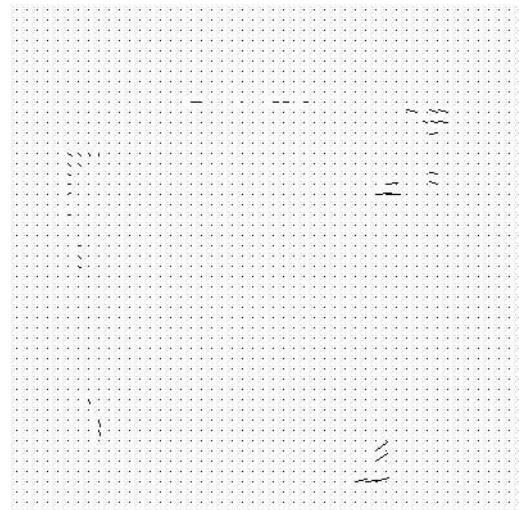
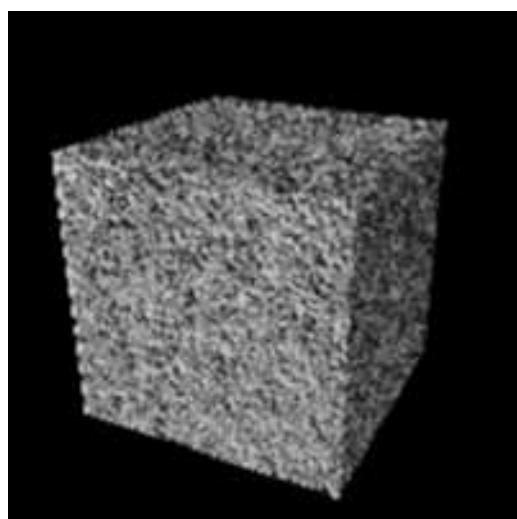
Optischer Fluß von *Anandan*.Ergebnis mit *Lucas-Kanade*.

Abbildung 7.10: Anwendung von Optical-Flow Verfahren auf synthetischen Würfel nach Rotation um 1grd.

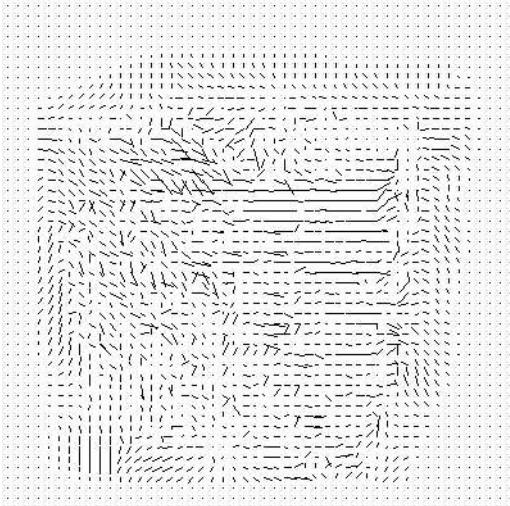


Würfel mit Oberflächentextur.

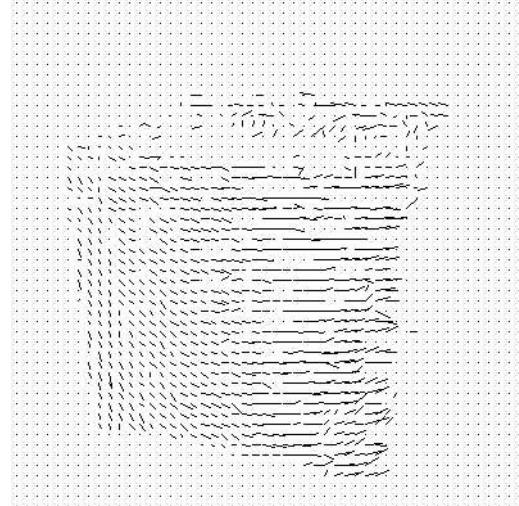


Büste mit marmorierter Textur.

Abbildung 7.11: Einsatz von Texturen zur Verbesserung der Verschiebungsvektorbestimmung.



Resultat mit *Anandan*.

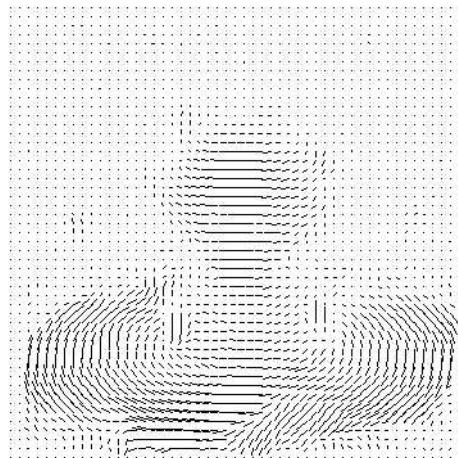
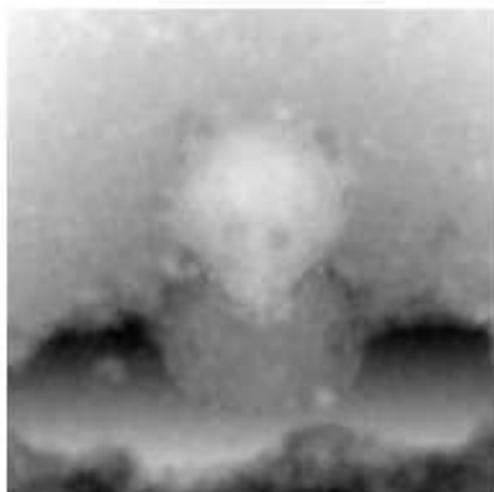


Ergebnis mit *Lucas-Kanade*.

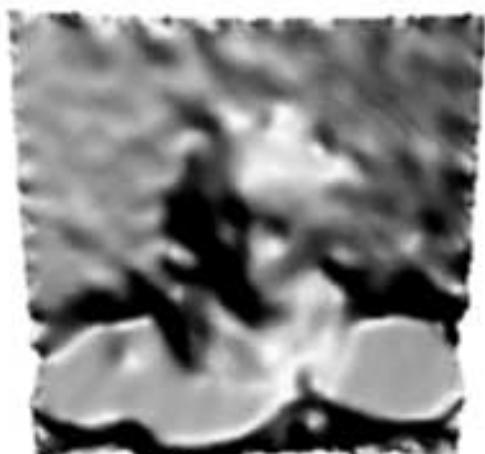
Abbildung 7.12: Anwendung der Optical-Flow Verfahren bei texturierten Oberflächen.



Grauwertbild einer realen Szene.

Optischer Fluß mit *Anandan*.

Rekonstruierte Tiefenkarte.



Geglättete 3D-Visualisierung.

Abbildung 7.13: Rekonstruktion einer realen Szene unter Verwendung des Optical-Flow Verfahrens von *Anandan* nach einer $5\text{ }grd$ Drehung.

7.3 Merkmalsbasierte Gestaltrekonstruktion

In den nachfolgenden Abb. 7.14 und 7.15 sind die Ergebnisse der merkmalsbasierten Gestaltrekonstruktion für die bekannten künstlich generierten Bilder der Würfelszene und die im Labor aufgenommenen Bilder der komplexen Realszene dargestellt. Auf beide Bildfolgen wurden die Verfahren zur Gewinnung von Merkmalen sowie ausgezeichneter Verschiebungsvektoren angewendet. Im Interpolationsprozeß ist in beiden Fällen eine Konsistenzüberprüfung und Glättung durchgeführt worden. Die Qualität der nicht dargestellten Kantenvektoren entspricht den in Abschnitt 5.3 abgebildeten Verschiebungsvektorkarten. In der 3D-Visualisierung ist auf die Glättung der Tiefenkarte verzichtet worden.

Künstliche Würfelszene

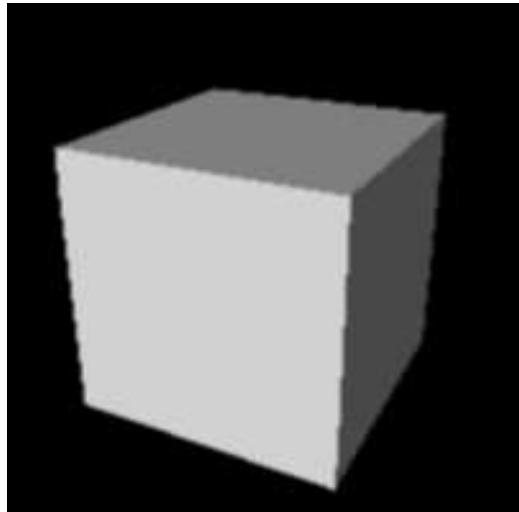
Wie in Abschnitt 5.3 beschrieben, beruht die Würfelszene auf der Vektorfeldkombination von vier Bildern, die im Abstand von jeweils 5 Grad aufgenommen wurden. Die Abb. 7.14 zeigen, daß die Bestimmung der Gestalt aus Kanten und Eckpunkten bei bekanntem Drehwinkel (hier 15 Grad) mit annehmbaren Verlusten bewältigt werden kann. Die auftretenden Fehler sind auf falsche Kantenverschiebungsvektoren zurückzuführen und daraus resultiert die fehlerhafte Interpolation.

Verblüffend ist die Detektion der Würfelkanten, die im dichten Verschiebungsvektorfeld nicht nachzuvollziehen ist. Durch die 3D-Visualisierung der Szene (Abb. 7.14 d) aus einer anderen Betrachterposition wird deutlich, daß auch die Winkel zwischen den Seitenflächen des Würfels erhalten bleiben.

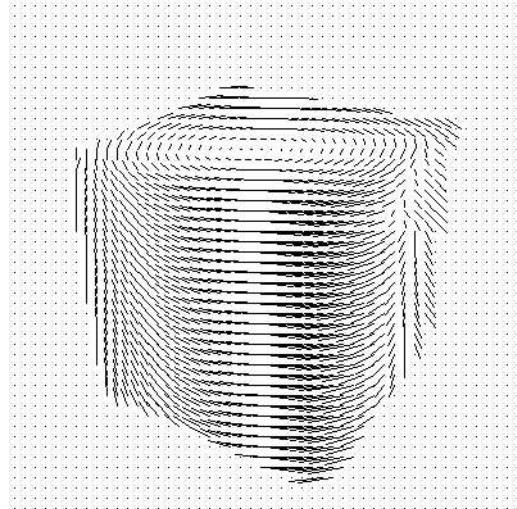
Reale Drehtellerszene

Etwas anders sieht die Situation bei der Rekonstruktion einer Realszene aus, welche aus zwei Bildern mit jeweils 10 Grad Drehung berechnet wurde (vgl. Abb. 7.15).

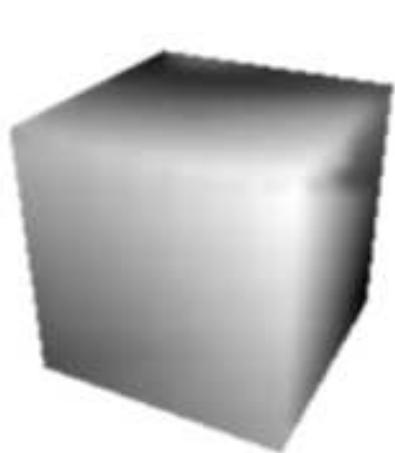
Hier zeigen sich im dichten Verschiebungsvektorfeld Inkonsistenzen, die auch durch Interpolation nicht behoben werden konnten. Typische Probleme entstehen durch Verdeckungen, Texturen (hier ein Schriftzug) und fehlende Kanten, die dann zu falschen Verschiebungsvektoren führen. Aus der Tiefenkarte ist zumindest die korrekte Lage der Objekte zueinander erkennbar. Für eine Weiterverarbeitung sind die Ergebnisse der Gestaltrekonstruktion dieser Szene jedoch zu fehlerhaft.



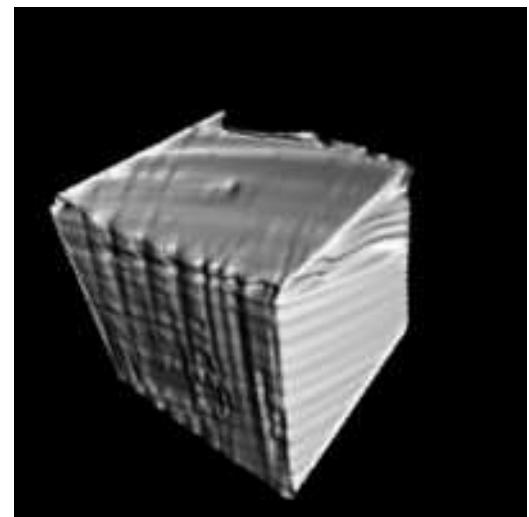
a) Grauwertbild



b) Interpolierte Verschiebungsvektoren

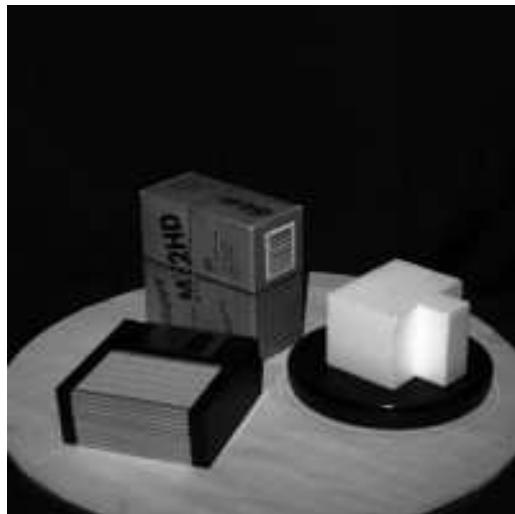


c) Tiefenkarte

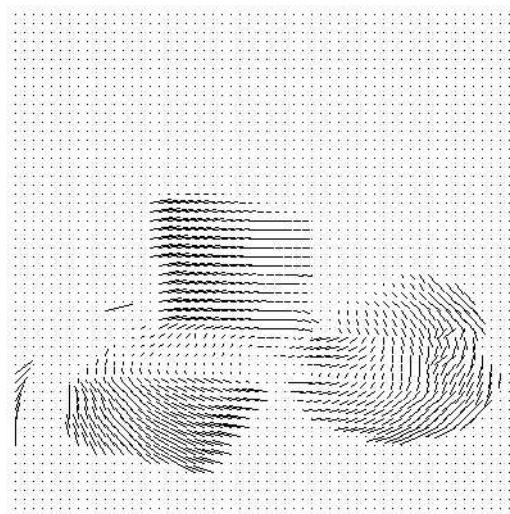


d) 3D-Rekonstruktion der Tiefenkarte

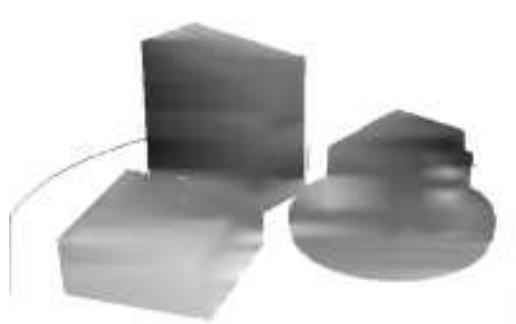
Abbildung 7.14: Gestaltsrekonstruktion einer Würfelszene: a) Das künstlich Würfelbild b) Aus Kanten-Verschiebungsvektoren interpoliertes Vektorfeld unter Zuhilfenahme eines segmentierten Ausgangsbildes c) Rekonstruierte Tiefenkarte der Interpolation d) 3D-Visualisierung der Tiefenkarten der Szenen aus einem anderen Sichtwinkel.



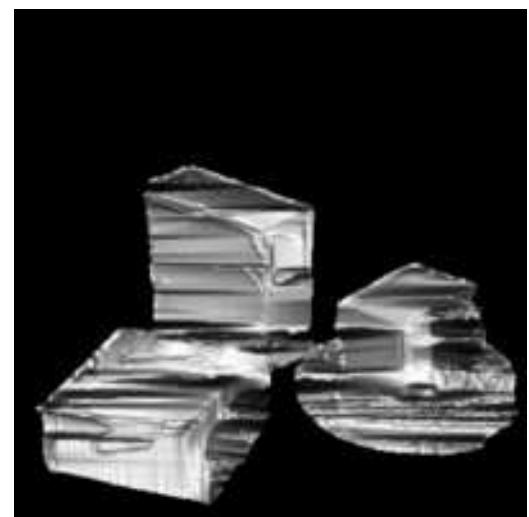
a) Grauwertbild



b) Interpolierte Verschiebungsvektoren



c) Tiefenkarte



d) 3D-Rekonstruktion der Tiefenkarte

Abbildung 7.15: Gestaltsrekonstruktion einer Realszene: a) Das im Labor aufgenommene Bild b) Aus Kanten-Verschiebungsvektoren interpoliertes Vektorfeld unter Zuhilfenahme eines interaktiv segmentierten Ausgangsbildes c) Rekonstruierte Tiefenkarte der Interpolation d) 3D-Visualisierung der Tiefenkarten der Szenen.

7.4 Bewertung

Das implementierte Verfahren zur Kamerakalibrierung kann aufgrund seiner Genauigkeit und Stabilität zur Benutzung uneingeschränkt weiterempfohlen werden. Die vorgestellten Kantendetektoren liefern bei komplexen Realszenen nicht genügend Merkmale, was die nachfolgenden Verfahren stark beeinflußt. Sie sollten im Zusammenhang mit Segmentierungsprozessen weiter verbessert werden.

Die vorgestellten Beispiele der präsentierten Verfahren zeigen, daß bei korrekten und vollständigen Verschiebungsvektoren die Rekonstruktion der Gestalt und Tiefe erfolgreich durchgeführt werden kann. Das Verfahren mit Kenntnis des Drehwinkels ist für eine exakte und stabile Berechnung der Tiefe aus Verschiebungsvektoren den anderen präsentierten Techniken vorzuziehen. Restümierend ist zu bemerken, daß der erfolgsversprechende Ansatz, die Gestalt aus Bildfolgen über eine Drehtellergometrie zu rekonstruieren, weiterverfolgt werden sollte.

7.4.1 Ausblick und Fortsetzung

Für die Fortsetzung bzw. Erweiterung dieser Arbeit bieten sich vielfältige Möglichkeiten an. Die Kalibrierung könnte um eine Bestimmung der Lichtquellenrichtung erweitert werden, um entsprechende Verfahren zu unterstützen. Die aus den Verschiebungsvektoren für ausgezeichnete Merkmale gewonnenen absoluten Tiefen ließen sich im Fourierraum mit den relativen Tiefenwerten aus einem *shape from shading* Verfahren zu einer komplexen Gestaltrekonstruktion kombinieren. Hierbei wären die hochfrequenten Anteile aus den Shading- und die niederfrequenten aus den partiellen Tiefenkarten der vorgestellten Verfahren zu vereinigen.

Auch wäre eine Verbesserung der Merkmalsgewinnung durch sogenannte *featurebased energie detectors* oder eine Ergänzung der Merkmale um Flächen möglich. Das Motionkonstraint könnte durch Methoden der Computer Grafik (*Constructive solid geometry CSG*) wie folgt erweitert werden:

Jeder Umriß einer Szene aus Betrachterrichtung stellt eine Eingrenzung der räumlichen Ausdehnung der Objekte dar. Er definiert in jeder Aufnahme neue kegelförmige Polyeder. Der Schnitt dieser Polyeder aus verschiedenen Ansichten ergibt dann eine äußere Hülle der Szene. Schließlich bietet sich im Bereich der Visualisierung als Fortsetzung eine Kombination der $2\frac{1}{2}$ D Daten zu einem echten 3D Modell (Voxel oder polyedrische Repräsentation) an.

Hinweis

Die Angaben und Programme wurden mit großer Sorgfalt erarbeitet bzw. zusammengestellt. Trotzdem sind Fehler nicht ganz auszuschließen. Wir möchten deshalb darauf hinweisen, daß weder eine Garantie noch irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler wären wir jederzeit dankbar.

Kapitel 8

Programmbeschreibung

8.1 Benutzeranleitung

8.1.1 Programmsteuerung

Die Steuerung der Programme erfolgt durch die Angabe von *Optionen*, welche mit einem Spiegelstrich - eingeleitet werden. Der direkt nachfolgende Buchstabe ist entscheidend, egal ob er groß oder klein geschrieben wird. Ebenso kann dieser beliebig erweitert werden, d.h. die Option -h ist identisch mit -help oder dem Kommando -HILFE.

Erwartet eine Option mehrere Parameter, so sind sie durch Leerzeichen voneinander zu trennen. Die Reihenfolge ist unwesentlich und bei sich ausschließenden Optionen wird nur die letzte Angabe berücksichtigt. Gibt man beim Programmaufruf ein ? oder -h an, so erscheint ein *Benutzungshinweis* mit der Auflistung der unterstützten Optionen.

8.1.2 Kamera-Kalibrierung

Das Kalibrierungsobjekt

Für die Kamerakalibrierung ist es notwendig, daß ein Objekt möglichst exakt vermessen wird, d.h für mindestens 5 coplanare (bzw. 7 nicht-coplanare) Punkte müssen die dreidimensionalen Weltkoordinaten bekannt sein. Der Ursprung und die Orientierung können prinzipiell frei gewählt werden. Die optische Achse der Kamera darf allerdings nicht genau auf den Ursprung des Weltkoordinatensystems ausgerichtet werden, denn aufgrund der verwendeten Kalibrierungstechnik darf die vertikale Translation T_y nicht 0 sein.

Die Kalibriermarken

Die für die Kamerakalibrierung notwendige subpixelgenaue Punktbestimmung aus der Aufnahme des exakt vermessenen Kalibrierungsobjekts wird interaktiv durch den Benutzer unter Zuhilfenahme von calpnt durchgeführt.

```
> calpnt -vh
(Cal)ibriation (p)oi(nt)s, $Revision: 1.1 $$State: Exp $$Date: 1994/03/25 21:54:57 $

Usage: calpnt [-q] [-v] [-h] ... <pgmfile>
-t <color> : Schwellwert fuer Binarisierung (Default 127)
Wertebereich: 0 <= <color> <= 255
-l : Modus mit lokalen Bildoperationen (Default)
-g : Modus mit globalen Bildoperationen
-m : Modus mit manueller Bestimmung
-k : Start des Koordinatensystems Pixelmitte
-o <file> : speicher die Daten in der Datei <file>
              (Default Bildschirmausgabe)
-G <gamma> : Gammakorrektur des Bildes
-d <name> : oeffnet das Display <name> (Default :0)
-q : Keine Textausgabe
-v : Version
-h : Hilfe
```

Das Programm ermöglicht die Festlegung der Koordinaten auf drei verschiedene Arten: manuell, global und lokal. Im *manuellen* Modus werden die Kalibrierungspunkte in der Vergrößerung eines zuvor angewählten Bildausschnittes vom Benutzer durch Festlegen des Mittelpunktes bestimmt. Im Gegensatz dazu werden sowohl im *lokalen* als auch im *globalen* Modus die Punktkoordinaten automatisch berechnet. Es muß in beiden Modi lediglich die Auswahl und Reihenfolge der ermittelten Kalibrierungspunkte interaktiv durchgeführt werden.

Lokal erfolgt die Segmentierung und Subpixelberechnung innerhalb eines gewählten Bildausschnittes, in dem der Benutzer die Auswahl bestätigt. Im Verfahren mit globalen Bildoperationen wird durch die interaktive Eingabe eines Polygons, das alle interessanten Bildbereiche enthalten sollte, eine grobe Vorauswahl getroffen. Dieser Ausschnitt wird jetzt vom Benutzer und dem Programm binarisiert, bis es möglich ist alle Kalibrierungspunkte zu identifizieren, um dann abschließend ausgewählt zu werden. Die Modifikation des intern verwendeten Schwellwerts (siehe Kapitel 4) ist über den Parameter **-t <color>** realisiert.

Die subpixelgenauen Koordinaten werden auf dem Bildschirm oder optional in eine ASCII-Datei ausgegeben. Sie beziehen sich auf den *links unten* liegenden Ursprung des Bildes, eines linkshändigen Koordinatensystems, d.h. es zeigen die positive X-Achse nach rechts und die Y-Achse nach oben. Für den Ursprung wird angenommen, daß er am linken unteren Rand des Pixels liegt. Mit der Option **-k** kann er jedoch auch in die Pixelmitte verschoben werden.

Lineare Kalibrierung LINCAL

Das Programm **lincal** führt eine lineare Kamera-Kalibrierung durch und berechnet die affine Transformation zwischen Kamera- und Weltkoordinatensystem sowie eine Approximation der Brennweite. Sofern keine Ausgabedatei angegeben wird, werden die Ergebnisse auf dem Bildschirm angezeigt.

```
> lincal
LINCAL - Lineare Kalibrierung V1.0

AUFRUF   : lincal [?] [OPTIONEN] -w <datei> -i <datei> -s <datei>

-w[orld]  <datei.wrld> Weltkoordinaten der Kalibrierungspunkte.
-i[image] <datei.img> Bildkoordinaten der Kalibrierungspunkte.
-s[sensor] <datei.cam> Herstellerangaben ueber den Sensor.

OPTIONEN : -o[ut]      <datei.cal> Name der Ausgabedatei.
           -c[oplanar]          Kalibrierungspunkte liegen in XY-Ebene.
           -q[uiet]              Keine Bildschirmausgaben erzeugen.
           -h[elp]               Ausgabe dieses Benutzungshinweises.
```

Kalibrierungs Optimierung OPTCAL

Eine mit `lincal` erzeugte Kalibrierungsdatei kann noch optimiert werden. Bei der Verbesserung der Kalibrierungsdaten mit dem Programm `optcal` wird die radiale Verzerrung berechnet und die Brennweite angepaßt.

```
> optcal
OPTCAL - Kalibrierungs Optimierung V1.0

AUFRUF   : optcal [?] [OPTIONEN] -w <datei> -i <datei> -c <datei>

-w[orld]  <datei.wrld> Weltkoordinaten der Kalibrierungspunkte.
-i[image] <datei.img> Bildkoordinaten der Kalibrierungspunkte.
-c[al]     <datei.cal> Lineare Kalibrierungsdaten.

OPTIONEN : -o[ut]      <datei.opt> Name der Ausgabedatei.
           -f[ocus]    n      Brennweite 'f' in [mm] explizit definieren.
           -a[spect]   n      Bildskalierungsfaktor sx definieren.
           -1          n      Verzerrungskoeffizient 'k1' definieren.
           -2          n      Verzerrungskoeffizient 'k2' definieren.
           -q[uiet]    Keine Bildschirmausgaben erzeugen.
           -h[elp]     Ausgabe dieses Benutzungshinweises.
```

Radiale Entzerrung CALDIS

Nicht alle Verfahren benutzen das verwendete Kameramodell mit einer radialen Linsenverzerrung. Damit bei den Berechnungen nicht immer die Verzerrung eingebaut werden muß, versucht das Programm `caldis` das Grauwertbild entsprechend der Kalibrierung zu entzerren. Bei der Sony CCD Videokamera sind die Abweichungen aber nicht gravierend, so daß man sich diese zeitintensive Prozedur ersparen kann.

```
> caldis
CALDIS - Radiale Entzerrung V1.0

AUFRUF   : caldis [?] [OPTIONEN] -c <datei> -p <datei> -o <datei>

-c[al]      <datei.opt> Optimierte Kalibrierungsdaten.
-p[ic]      <datei.pgm> Kamerabild der Kalibrierungspunkte.
-o[ut]      <datei.pgm> Name der Ausgabedatei.

OPTIONEN : -f[ocus]    n          Brennweite 'f' in [mm] explizit definieren.
          -a[spect]   n          Bildskalierungsfaktor 'sx' definieren.
          -1           n          Verzerrungskoeffizient 'k1' definieren.
          -2           n          Verzerrungskoeffizient 'k2' definieren.
          -q[uiet]     Keine Bildschirmausgaben erzeugen.
          -h[elp]      Ausgabe dieses Benutzungshinweises.
          -n[oscreen]  WorkBench Fenster oeffnen (nur AMIGA).
```

Kalibrierungs Visualisierung CALVIS

Das Ergebnis der Kamera-Kalibrierung kann schließlich mit dem Programm `calvis` betrachtet werden. Wird nur die Kalibrierungsdatei als Parameter angegeben, so wird ihr Inhalt auf dem Bildschirm angezeigt. Um fehlerhaft eingegebene Koordinaten besser ausfindig machen zu können, ist es in Verbindung mit den Welt- und Bildkoordinaten möglich, den (Rück-)Projektionsfehler für jeden Kalibrierungspunkt zu speichern.

Die Qualität der Kalibrierung kann aber auch visuell begutachtet werden. Durch die zusätzliche Übergabe des Kamerabildes erscheint das Grauertbild in einem Fenster, wozu allerdings ein X-Windows Display mit 8bit-Pseudocolor und eingeschaltetem Backing-Store erforderlich ist. Die Weltkoordinaten werden dann gemäß der kalibrierten Werte in das Bild projiziert und mit einem Kreuzchen markiert. Es besteht ebenfalls die Möglichkeit das Drehtellerraster einzublenden.

```
> calvis
CALVIS - Kalibrierungs Visualisierung V1.0

AUFRUF   : calvis [?] [OPTIONEN] <datei.cal|opt>

OPTIONEN : -w[orld]    <datei.wrld> Weltkoordinaten der Kalibrierungspunkte.
          -i[mage]     <datei.img> Bildkoordinaten der Kalibrierungspunkte.
          -p[ic]        <datei.pgm> Grauwertbild der Kalibrierungspunkte.
          -o[ut]        <datei.pgm> Name der Ausgabedatei.
          -error]       <datei.err> Projektionsfehler der Kalibrierungspunkte.
          -f[ocus]      n          Brennweite 'f' in [mm] explizit definieren.
          -a[spect]    n          Bildskalierungsfaktor 'sx' definieren.
          -1            n          Verzerrungskoeffizient 'k1' definieren.
          -2            n          Verzerrungskoeffizient 'k2' definieren.
          -h[elp]      Ausgabe dieses Benutzungshinweises.
          -q[uiet]     Keine Bildschirmausgaben erzeugen.
          -m[esh]       Dreh scheibengitter zeichnen.
          -n[oscreen]  WorkBench Fenster oeffnen (nur AMIGA).
```

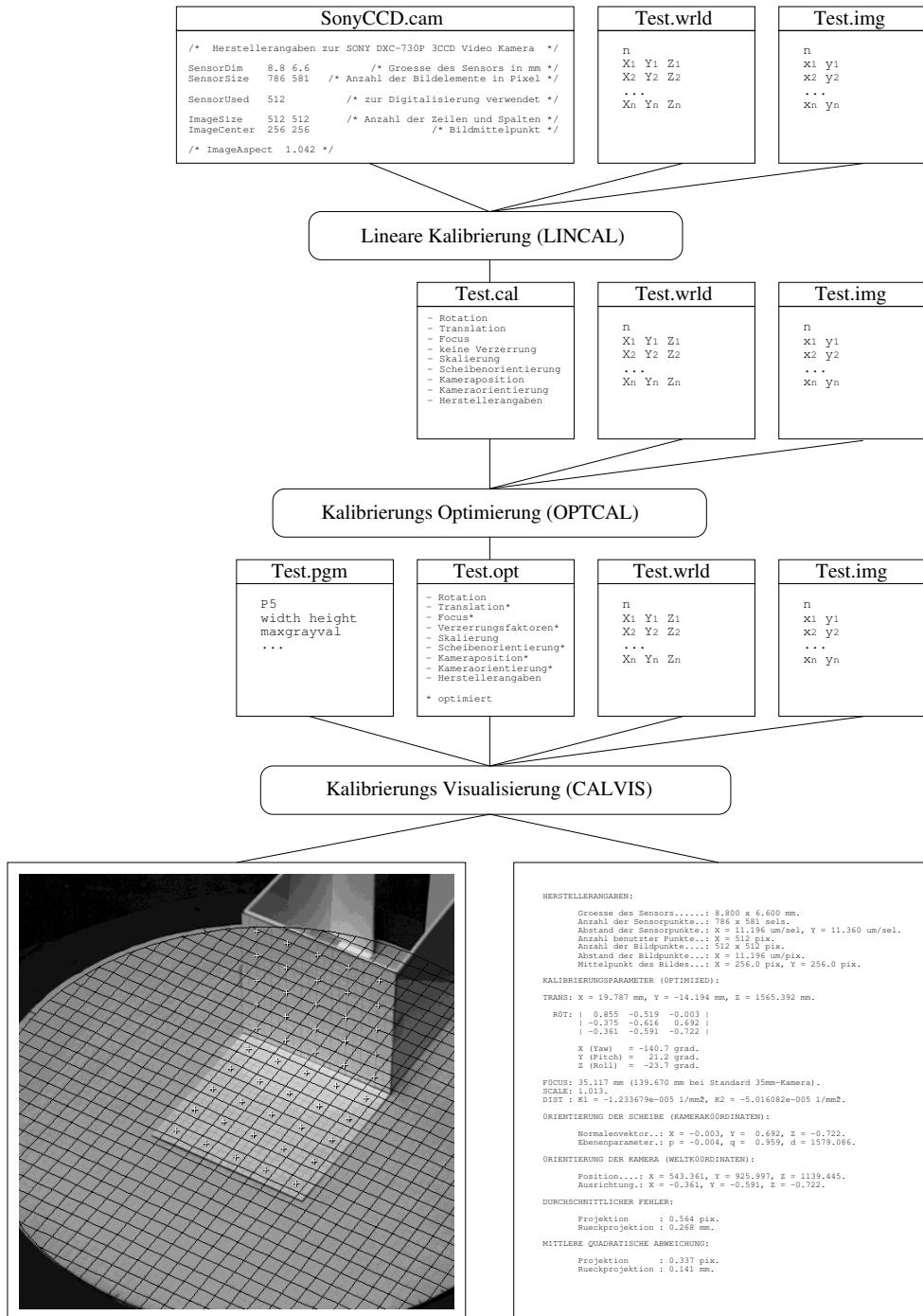


Abbildung 8.1: Datenfluß bei der Kamera-Kalibrierung.

Kalibrierungsdaten Generator SYNTH

Für die synthetische Generierung von Kalibrierungsdaten ist das Programm *synth* entwickelt worden. Beim Start wird zunächst ein voreingestellter Datensatz von *Welt- und Bildkoordinaten* erzeugt. Dadurch ist gewährleistet, daß sinnvolle Parameter verwendet werden und alle Kalibrierungspunkte im Bild sichtbar sind. Durch die Angabe von zusätzlichen Optionen kann schließlich der Einfluß einer Änderung von jedem einzelnen Kalibrierungsparameter ausprobiert werden.

Zur optischen Kontrolle der synthetisch generierten Datensätze wird zusätzlich ein *Kamerabild* erzeugt. Durch die Verbindung der Kalibrierungspunkte zu Gittern soll die räumliche Darstellung verbessert und die radiale Verzerrung hervorgehoben werden. Die bei der Generierung verwendeten *idealen Parameter* werden ebenfalls gespeichert, so daß die Qualität der anschließenden Kamera-Kalibrierung einfacher verifiziert werden kann.

```
> synth
SYNTH - Kalibrierungsdaten Generator V1.0

AUFRUF : synth [?] [OPTIONEN] -s <datei.cam> -o <datei.cal>

      -s [ens]   <datei.cam> Herstellerangaben.
      -o [ut]    <datei.cal> Basisname der Testdaten.

OPTIONEN : -h[elp]           Ausgabe dieses Benutzungshinweises.
          -f[ocus] n        Brennweite in [mm] explizit definieren.
          -a[spect] n       Bildskalierungsfaktor 'sx' definieren.
          -1 n              Verzerrungskoeffizient 'k1' definieren.
          -2 n              Verzerrungskoeffizient 'k2' definieren.
          -e[rror] w i     Standardabweichung fuer Welt- und Bildpunkte.
          -r[ot] y p r     Rotationswinkel (Yaw, Pitch, Roll) [grad].
          -t[rans] x y z   Translationsvektor [mm].
          -d[ist] x y z   Abstand der Kalibrierungspunkte [mm].
          -c[ount] x y z   Anzahl der Kalibrierungspunkte.
          -q[uiet]          Keine Bildschirmausgaben erzeugen.
          -n[oscreen]        WorkBench Fenster oeffnen (nur AMIGA).
```

8.1.3 Vorverarbeitung

Die Algorithmen zur Bildverbesserung wurden in dem Programm *hist* zusammengefaßt und erlauben eine Vorverarbeitung der Grauwertbilder durch Histogramm-Modifikationen. Durch die globale *Grauwertanpassung* wird der existierende Intensitätsbereich zur Kontrastverstärkung auf den gesamten zur Verfügung stehenden Bereich gestreckt.

Bei der *dynamischen Schwellwertbildung* kann die Binarisierung durch das Verhältnis von Maxima zu Minima im Histogramm beeinflußt werden. Mit der *dynamischen Histogramm-Egalisierung* kann der lokale Kontrast maximal angehoben werden und die Aufteilung des Bildes bei den adaptiven Verfahren erfolgt durch die Angabe einer Fenstergröße.

```
> hist
HISTOGRAMM - Bildverbesserung V1.0

AUFRUF   : hist [?] [OPTIONEN] {-a|-t n|-e} <datei.pgm>

-a[dapt]           Grauwertanpassung.
-t[hresh] n       Dynamische Schwellwertbildung [0..1].
-e[qual]          Dynamische Histogramm-Egalisierung.

OPTIONEN : -o[ut]      <datei.pgm> Ergebnisbild speichern.
          -w[indow] n Fenstergroesse [16..64].
          -d[ynl]        Schwellwert statt Binarbild anzeigen.
          -h[elp]         Ausgabe dieses Benutzungshinweises.
          -q[uiet]        Keine Bildschirmausgaben erzeugen.
          -n[oscreen]     WorkBench Fenster oeffnen (nur AMIGA).
```

8.1.4 Merkmalsextraktion

Der Marr-Hildreth-Operator MARR

Für die pixelgenaue Bestimmung von Grauwertkanten kann der Marr-Hildreth-Operator `marr` eingesetzt werden. Das Programm wird hauptsächlich durch die Angabe der Standardabweichung σ gesteuert, jedoch kann der Radius r der Filtermaske auch manuell variiert werden. Um das Hintergrundrauschen von den echten Kanten trennen zu können, muß ein Vergleich mit dem binarisierten Gradientenbetrag des Ausgangsbildes durchgeführt werden.

Der verwendete Schwellwert g bezieht sich auf die Steigung der ersten Ableitung und bestimmt den Bereich, der aus dem verrauschten Resultat des Marr-Hildreth-Operators in das Kantenbild übernommen wird. Um ein unerwartetes Ergebnis im resultierenden Kantenbild besser verifizieren zu können, ist es möglich den Bearbeitungsprozeß durch das Anzeigen von Zwischenergebnissen zu verfolgen.

```
> marr
MARR HILDRETH - LoG Operator V1.0

AUFRUF   : marr [?] [Optionen] -s n <datei.pgm>

-s[igma] n       Standardabweichung [0..38].

OPTIONEN : -o[ut]      <datei.pgm> Binarisiertes Kantenbild speichern.
          -r[adius] n   Radius der Filtermaske [1..24].
          -g[rad] n      Schwellwert fuer Gradientenvergleich.
          -h[elp]         Ausgabe dieses Benutzungshinweises.
          -t[race]        Zwischenergebnisse anzeigen.
          -q[uiet]        Keine Bildschirmausgaben erzeugen.
          -n[oscreen]     WorkBench Fenster oeffnen (nur AMIGA).
```

Der Kantendetektor CANNY

Mit dem Programm `canny` können pixelgenaue Grauwertkanten detektiert werden. Die Steuerung erfolgt wie bei `marr` durch die Standardabweichung σ und optional durch den Radius r der Filtermaske. Die vielen Zwischenergebnisse können zur Kontrolle ebenfalls angezeigt werden.

```
> canny
CANNY EDGE - Kantendetektor V1.0

AUFRUF : canny [?] [OPTIONEN] -s n <datei.pgm>

-s[igma] n Standardabweichung.

OPTIONEN : -o[ut] <datei.pgm> Binarisiertes Kantenbild speichern.
           -r[adius] n Radius der Filtermaske [1..24].
           -h[elp] Ausgabe dieses Benutzungshinweises.
           -t[race] Zwischenergebnisse anzeigen.
           -q[uiet] Keine Bildschirmausgaben erzeugen.
           -n[oscreen] WorkBench Fenster oeffnen (nur AMIGA).
```

Merkmalsbestimmung CCM

Die im Kapitel 4 beschriebenen Verfahren zur Merkmalsextraktion sowie Teile der im Kapitel 3 vorgestellten Vorverarbeitungsschritte sind im Programm `ccm` zusammengefaßt worden.

```
> ccm -vh
(C)anny(C)ornerResponse(M)arrHildreth, $Revision: 1.1 $$Date: 1994/03/25 21:52:05 $

Usage: ccm [-q] [-v] [-h] ... <pgmfile>
       -o <file> : speicher das(die) Ergebnisbild(er) in <file>
       -m <mode> : <mode> = 1: Canny Edge (Default)
                    = 2: Corner Response
                    = 3: Marr Hildreth
       -G <gamma> : Gammakorrektur des Bildes (Default 1.0)
       -M          : Median (3x3) Filterung
       -n          : CE: kein 'thinning'
       -T <num>   : MH: Gradienten 'threshhold' (Default 0.0)
       -S <sigma> : CE & MH (Default 1.5)
       -R <radius> : CE & MH (Default 4.0)
       -K <kappa> : CR: Grauwert Nullpunkt (Default 1/0.04)
       -d <name>  : oeffne Display <name> (Default kein Display)
       -q          : Keine Textausgabe
       -v          : Version
       -h          : Hilfe
```

Die Funktionsweise der verschiedenen Verfahren ist in den oben erwähnten Abschnitten ausreichend erläutert worden. Vor der Merkmalsberechnung kann das Eingabebild optional mit einem 3×3 Median gefiltert und eine Gammakorrektur der Grauwerte durchgeführt werden. Es werden nur Bilder im PBM P2- oder P5-Format eingelesen. Die Option `-o <file>` bestimmt den Namen des Ergebnisbildes. Der *Canny Edge Operator* liefert ein Binärbild, das im P5-Format abgelegt wird. Die beiden anderen Verfahren extrahieren noch zusätzliche Information in Form weiterer Bilder.

Das Resultat der *Corner Response Funktion* wird in drei Teilen gespeichert: Die *Kantenmerkmale* als P5-Grauwertbild mit der Endung '.e', die *punktförmigen Merkmale* mit der Endung '.p', sowie das Ergebnis der *Corner Response Funktion* als File im I8-Format (siehe 8.2.4). Zusätzlich zum P5-Binärbild der Nulldurchgänge des *Marr-Hildreth-Operators* werden die im Abschnitt 4.1 beschriebenen Subpixelinformationen in zwei weiteren Dateien im F8-Format abgelegt. Diese Abweichung in Zeilenrichtung erhält die Endung '.x' und in Spaltenrichtung '.y'.

8.1.5 Verschiebungsvektorbestimmung

Der Simulator SIMUL

Die synthetische Generierung von Verschiebungsvektoren einer Bildfolge kann mit dem Simulator **simul** durchgeführt werden. Die Ausgabe von idealen Tiefenkarten und der verwendeten Kalibrierungsdaten ist ebenfalls möglich. Aufgrund der Komplexität wird das Programm nicht durch Optionen sondern mit einer umfangreichen Konfigurationsdatei gesteuert. Die Funktionsvielfalt und der konkrete Dateiaufbau werden im Abschnitt (8.2.1) erläutert.

```
> simul
SIMUL - Simulator fuer Optischen Fluss V2.0

AUFRUF   :  simul [?] [OPTIONEN] <datei.cfg>

OPTIONEN : -h[elp]           Ausgabe dieses Benutzungshinweises.
          -n[oscreen]        WorkBench Fenster oeffnen (nur AMIGA).
```

Ausgezeichnete Verschiebungsvektorbestimmung VV

Zur automatischen Berechnung von ausgezeichneten Verschiebungsvektoren wird das Programm **vv** bereitgestellt. Es ermöglicht die Bestimmung von Vektoren über eine Sequenz von Kantenbildern durch Auffinden der Korrespondenzen und der Kombination mehrerer berechneter Bilder.

```
> vv -vh
(V)erschiebungs(v)ektoren, $Revision: 1.1 $$State: Exp $ $Date: 1994/03/25 21:52:08 $

Usage: vv [-q] [-v] [-h] ... <seq_name> <seq_number>
      -o <v-file>  : Verschiebungsvektorkarte in <v-file> speichern
      -c <c-file>  : lese Kalibrierungsdaten aus <c-file>
      -P <phi>     : Schrittweite <phi> des Drehwinkels (in Grad)
      -R <radius>  : Drehteller <radius> in mm
      -Z <height>  : Drehteller Hoehe bzg. WKS in mm
      -L <level>   : Kombinationslevel der VV-Karten (Default 1)
      -% <percent> : Erlaubte max. Abweichung von mittleren bzw.
                     vorhergesagten VVs in Prozent (Default 10%)
      -q            : keine Textausgabe
      -v            : Version
      -h            : Hilfe
```

Benötigt werden die Kalibrierungsdaten in der entsprechenden Datei der Kamerakalibrierung, binäre Kantenbilder im Unterverzeichnis ‘./ca/’ sowie Bilder mit punktförmigen Merkmalen in ‘./cr/’. Die Dateien müssen alle im PBM P2- bzw. P5-Format vorliegen (mit Ausnahme der Kalibrierungsdaten). An **vv** wird der Familienname aller Bilder z.B. *cube.pgm* und deren Anzahl innerhalb der Sequenz übergeben. Sie unterscheiden sich durch eine Laufnummer am Ende ihres Names und ihre Lage in den Unterverzeichnissen. Die fortlaufende Nummerierung muß immer bei Null beginnen. Die Bilder der punktförmigen Merkmale benötigen zusätzlich noch die Kennung ‘.p’ (siehe 8.1.4).

Beispiele: *./ca/cube.pgm0*, *./ca/cube.pgm1*, ..., *./cr/cube.pgm0.p*, ...

Die berechneten Verschiebungsvektorkarten werden im Unterverzeichnis ‘./dpm/‘ abgelegt. Ihr Name setzt sich aus der durch `-o <vv-file>` übergebenen Bezeichnung und der Kombination der Nummer der Ausgangsbilder abhängig vom Kombinationslevel zusammen (siehe Abb. 5.5 in Abschnitt 5.3). Der übergebene Winkel ϕ der Drehtellerbewegung muß für alle Winkel zwischen zwei Aufnahmen einer Sequenz gleich sein.

Die Verschiebungsvektor-Interpolation IP

Um dichte Verschiebungsvektorkarten zu erhalten kann das Programm ip benutzt werden. Es beruht auf den im Abschnitt 5.3 beschriebenen Interpolationsprozessen und benötigt Verschiebungsvektoren von Kanten in Form eines Bildes (z.B. durch vv gewonnen) sowie ein segmentiertes Ausgangsbild, um die Berechnung bei eventuell fehlenden Kantenvektoren auf eine Fläche einzuschränken.

```
> ip -vh
(i)nter(p)olieren von Verschiebungsvektoren, $Revision: 1.1 $$Date: 1994/03/25 ...

Usage: ip [-q] [-v] [-h] ... <vvfile> <segfile>
      -o <vv-file> : Verschiebungsvektorkarte in <vv-file> speichern
      -% <percent> : Erlaubte max. proz. Abweichung der Interpolation
                      innerhalb des Segments (Default 10%)
      -k <size>    : Konsistenz des VV-Feld innerhalb einer
                      <size> x <size> Umgebung prüfen (Default 0)
      -g <size>    : VV-Feld mit <size> x <size> Maske glätten (Default 0)
      -q            : keine Textausgabe
      -v            : Version
      -h            : Hilfe
```

Die Option `-%` für die maximale prozentuale Abweichung bezieht sich auf den aktuell berechneten Vektor während der Interpolation.

8.1.6 Visualisierung

Anzeige von Verschiebungsvektorkarten SHOWDPM

Die Visualisierung von Verschiebungsvektorfeldern erfolgt mit dem Programm showdpm. In einem Mindestabstand von 4 pix werden die Vektoren (u, v) im Bild aufgetragen. Die Anzahl der darzustellenden Verschiebungsvektoren wird automatisch reduziert, wenn das Fenster für die Anzeige zu klein gewählt wurde.

Für besonders kleine oder extrem große Drehwinkel können die Vektoren bei der Betrachtung mit einem Faktor skaliert werden. Durch Drücken der Maustaste kann ein Ausschnitt der Karte in einem neuen Fenster vergrößert dargestellt werden (Zoom). So kann auch bei großen Karten jeder Verschiebungsvektor genau betrachtet werden.

```
> showdpm
SHOWDPM - Anzeigen von Verschiebungsvektoren V1.0

AUFRUF : showdpm [?] [OPTIONEN] <datei.dpm>

OPTIONEN : -o[ut]      <datei.pgm> Grauwertbild speichern.
           -r[educe]   n       Reduzieren der Punkte um den Faktor n.
           -d[imension] n       Groesse des Fensters.
           -s[cale]     n       Skalierungsfaktor fuer die Vektoren.
           -b[ckgrnd]   n       Grauwert fuer den Hintergrund.
           -h[elp]      Ausgabe dieses Benutzungshinweises.
           -q[uiet]     Keine Statusmeldung ausgeben.
           -n[oscreen]  WorkBench Fenster oeffnen (nur AMIGA).
```

Anzeige von Grauwertbildern SHOWPGM

Das kleine Programm `showpgm` dient lediglich dazu die verwendeten Grauwertbilder auf dem Bildschirm anzuzeigen.

```
> showpgm
SHOWPGM - Anzeigen einer Portable GrayMap V1.0

AUFRUF : showpgm [?] [OPTIONEN] <datei.pgm>

OPTIONEN : -h[elp]      Ausgabe dieses Benutzungshinweises.
           -n[oscreen]  WorkBench Fenster oeffnen (nur AMIGA).
```

Die Tiefenkarten Visualisierung VISUAL

Um das Ergebnis der Gestaltsrekonstruktion betrachten zu können wird der Tiefenkarten Visualisierer `visual` eingesetzt. Neben einer automatisch skalierten Grauwertdarstellung der Karte kann auch ein Polyeder erzeugt und dreidimensional begutachtet werden. In Abhängigkeit von dem Skalierungsfaktor des Sensors muß der XY-Gitterabstand des Polyeders angepaßt werden. Er ist für den Simulator mit 1.0 voreingestellt und ergibt sich bei einer realen Kamera, wenn man durch den Abstand benachbarter Bildpunkte teilt. Bei der verwendeten SONY CCD Videokamera entspricht das einem Gitterabstand von etwa 89.3.

Für die Objektoberfläche können unterschiedliche Darstellungen vom Drahtgittermodell bis zur interpolierten Schattierung gewählt werden. Bei starkem Rauschen kann die Tiefenkarte auch geglättet werden und durch Einschalten von Antialias können Grauwertübergänge optisch verbessert werden. Die Betrachtungsparameter sind entsprechend der Beschreibung von `simul` zu wählen und für eine erste Vorschau ist es meistens sinnvoll, die Anzahl der verwendeten Tiefenwerte zu reduzieren.

```
> visual
VISUAL - Tiefenkarten Visualisierer V2.0

AUFRUF   :  visual [?] [OPTIONEN] <datei.dep>

OPTIONEN : -d[imension] n          Groesse des quadratischen Fensters.
           -f[ocus]      n          Brennweite bzgl. 35mm Kamera in [mm].
           -r[educe]     n          Reduzieren der Punkte um den Faktor n.
           -b[ckgrnd]    n          Grauwert fuer den Hintergrund.
           -m[esh]       n          Skalierungsfaktor fuer XY-Gitterabstand.
           -l[ight]      x y z    Richtungsvektor zur Punktlichtquelle.
           -c[amera]     x y z    Position der Kamera.
           -t[target]    x y z    Ausrichtung der Kamera.
           -o[ut]        <datei.pgm> Grauwertbild speichern.
           -v[ertex]      3D-Gitternetz mit verdeckten Flaechen.
           -i[nterpol]    Schattierte 3D-Darstellung mit Glaettung.
           -s[hading]    Schattierte 3D-Flaechen mit Lambertmodell.
           -w[ireframe]  3D-Drahtgitterliniennetz.
           -g[auss]       Tiefenkarte mit Gauss glaetten.
           -a[ntialias]  Kantenglaettung einschalten.
           -q[uiet]       Keine Statusmeldung ausgeben.
           -h[elp]        Ausgabe dieses Benutzungshinweises.
           -n[oscreen]    WorkBench Fenster oeffnen (nur AMIGA).
```

8.2 Dateiformate

8.2.1 Aufbau der Konfigurationsdatei .cfg

Die Konfigurationsdatei besteht aus ASCII-Zeichen, die mit einem herkömmlichen Editor bearbeitet werden können. Bei der Erstellung einer neuen Datei ist zur Kennzeichnung das Anhängen des Suffixes '.cfg' empfehlenswert.

Reservierte Schlüsselwörter

Die folgenden Schlüsselwörter sind in beliebiger Reihenfolge zu verwenden, wobei zwischen Groß- und Kleinschreibung nicht unterschieden wird:

Albedo	AntiAlias	Background	CalbFile	DispFile
DepthFile	Focus	ImageFile	Light	Noise
Observer	Points	Polygons	Quiet	Scale
Smooth	Target	Texture	Transform	

Ergänzende Kommentare, die in der C-Syntax mit `/* ... */` gekennzeichnet sind, können an beliebiger Stelle im Text zur Erklärung verwendet werden. In der folgenden Befehlsbeschreibung werden Schlüsselwörter in **Schreibmaschinenschrift** und Terminalsymbole, wie z.B. Konstanten oder Zeichenketten für Dateinamen, in **Schrägschrift** gedruckt. Optionale Parameter werden durch Klammerung [...] angegeben.

Bildschirm

Um die Größe der quadratischen Bilder und erzeugten Karten verändern zu können ist der Parameter

`Scale pixel`

zu verwenden.

Objektdefinition

Es kann ein *Polyeder* erzeugt werden, der durch Angabe von Punkten und Polygone definiert wird. Zu Beginn der *Punktliste* wird die Anzahl *n* der nachfolgenden 3D-Weltkoordinaten benötigt:

`Points n x1y1z1 x2y2z2 ... xnynzn`

Der Begriff Polygon wird in diesem Fall sehr speziell verwendet, denn es handelt sich im Prinzip jeweils nur um Dreiecke. In der *Polygonliste*, die durch die Polygonanzahl *m* angeführt wird, werden die Punktindizes (1, 2, ..., *n*) von den drei Begrenzungspunkten definiert:

`Polygons m p1,1p2,1p3,1 p1,2p2,2p3,2 ... p1,mp2,mp3,m`

Für eine korrekte Oberlächennormale ist die Reihenfolge der Punkte *im Uhrzeigersinn* zu konstruieren. Um eine Konvertierung von existierenden Polyedern zu vereinfachen, ist für das Weltkoordinatensystem des Simulators nicht die spezielle Drehscheibengeometrie erforderlich. Für die Objektkonstruktion zeigt die X-Achse nach rechts, die Y-Achse nach oben und die Z-Achse nach vorne. Im Simulator werden dann die Weltkoordinaten der Geometrie des Drehtellers angepaßt.

Betrachtungsparameter

Die *Kameraposition* wird mit

Observer *xpos ypos zpos*

festgelegt und die *Ausrichtung* kann mit

Target *xpos ypos zpos*

beeinflußt werden. Ein durch diese beiden Weltpunkte beschriebener Vektor entspricht dann der optischen Achse. Normalerweise wird diese in positiver Z-Richtung gewählt, d.h. die Kameraposition sollte sich im negativen Halbraum befinden. Die *Brennweite* der Kamera

Focus *dist*

wird bezüglich einer 35 mm Kamera angegeben. So können z.B. Normal- (50 mm), Weitwinkel- (28 mm) und Teleobjektive (135 mm) eingesetzt werden. Da eine ideale Perspektivprojektion verwendet wird, sind Unschärfen durch falsche Fokussierung und optische Verzerrungen bei der Simulation nicht möglich.

Beleuchtung

Die *Beleuchtungsrichtung* einer unendlich entfernten Punktlichtquelle kann mit

Light *xpos ypos zpos*

definiert werden. Die Koordinaten brauchen nicht normiert zu sein, so daß auch die 3D-Weltkoordinaten der Lichtquelle angegeben werden können.

Transformationen

Eine Folge von *n* Bildern wird durch die *Rotation des Polyeders* um die Drehtellerachse generiert. Der erste Listeneintrag *n* legt die Anzahl der Transformationen fest, gefolgt von den Rotationswinkeln *w_i*:

Transform *n w₁w₂...w_n*

Die Angabe der Drehung erfolgt in *grd* und definiert den *relativen Winkel* zum vorangegangenen Bild der Folge. Ein positiver Winkel entspricht dabei einer Drehung im Uhrzeigersinn, wenn man von oben auf die Drehscheibe schaut.

Weitere Einstellungen

Neben den bisher notwendigen Anweisungen sind auch diverse optionale Befehle vorhanden. Der Grauwert für die *Hintergrundfarbe* lässt sich mit

Background *intensity*

individuell auf die persönlichen Bedürfnisse einstellen. Für eine Ausgabe auf einem Drucker ist es manchmal sinnvoll, den normalerweise schwarzen Hintergrund auf weiß zu ändern. Der Wertebereich der einzelnen Graustufen variiert von 0 bis 255.

Um erweiterte *Statusmeldungen* bei der Berechnung zu unterdrücken, kann das Schlüsselwort

Quiet

verwendet werden.

Objektoberfläche

Bei der Berechnung eines Polyeders kann eine optische *Glättung* der Polygonübergänge mit

Smooth

durch lineare Interpolation der Oberflächennormalen erzielt werden. Die Entstehung von Treppenstufen an idealen Kanten kann durch die Verwendung von

AntiAlias

verhindert werden. Die in der Lambert-Reflektion verwendete *Reflektanzkonstante* kann durch

Albedo const

verändert werden und ist mit 1.0 voreingestellt. Der Polyeder kann auch mit einer *Oberfächentextur* überzogen werden:

Texture size name.pgm

Bei dem verwendeten Algorithmus wird das angegebene Grauwertbild der Größe *size* entsprechend vervielfältigt und als Würfel um das Objekt gelegt. In Abhängigkeit von der Oberflächennormalen wird dann eine Fläche des Würfels ausgewählt, deren Grauwerte auf die Oberfläche projiziert werden. Diese Vorgehensweise kann an Rundungen zu Diskontinuitäten führen, die jedoch bei ungerichteten natürlichen Texturen kaum auffallen.

Dateierzeugung

Zur Kommunikation zwischen den Programmen sind verschiedene Dateien eingeführt worden, die jeweils optional angelegt werden können.

```
Imagefile name.pgm
Depthfile name.dep
Dispfile name.dpm
Calbfile name.cal
```

Der Simulator kann alle *Grauwertbilder*, die *Tiefenkarte* des ersten Bildes, alle *Verschiebungsvektorfelder* und die *idealen Kalibrierungsdaten* speichern. Eine Unterscheidung in der Bildfolge wird durch Anhängen einer fortlaufenden Nummer gewährleistet.

Beispiel: Polyeder

Das folgende Beispiel soll die Definition von polyedrischen Objekten im Simulator veranschaulichen:

```

Background 255          /* Hintergrundfarbe weiss */
Scale      512           /* Groesse des Bildes 512x512 */

Observer   100.0 145.0 -200.0    /* Position der Kamera */
Target     0.0  45.0  0.0        /* Blickrichtung */
Light      2.0  1.0  -4.0       /* Lichtquellenposition */

Albedo     1.0            /* Reflektanzkonstante */
AntiAlias   1.0            /* Kantenglaettung einschalten */
Texture    90.0 stone.pgm    /* Oberflaechentextur */

Transform 3             /* 3 Bilder generieren */
5.0          /* Rotation um 5 Grad Y-Achse */
5.0          /* Rotation um weitere 5 Grad */
10.0         /* Insgesamt um 20 Grad rotiert */

Imagefile cube.pgm       /* Grauwertbild des Polyeders */
Depthfile cube.dep        /* Tiefenkarte des Polyeders */
Dispfile  cube.dpm        /* Verschiebungsvektorkarte */
Calbfile   cube.cal       /* Kalibrierungsdaten fuer das */
                          /* Scheibenkoordinatensystems */

Points 8
/* 1 */   -45  90  45      /* Y      1-----2 */
/* 2 */    45  90  45      /* |      /|      /| */
/* 3 */    45  0   45      /* |      5-----6 | */
/* 4 */   -45  0   45      /* |      ||      || */
/* 5 */   -45  90  -45     /* |      Z | 4-----+3 */
/* 6 */    45  90  -45     /* |      /|      /| */
/* 7 */    45  0   -45     /* |      8-----7 */
/* 8 */   -45  0   -45     /* +----- X */

Polygons 12
/* 1 */    6  1  2          /* Flaeche oben */
/* 2 */    1  6  5
/* 3 */    4  7  3          /* Flaeche unten */
/* 4 */    7  4  8
/* 5 */    1  8  4          /* Flaeche links */
/* 6 */    8  1  5
/* 7 */    3  6  2          /* Flaeche rechts */
/* 8 */    6  3  7
/* 9 */    4  2  1          /* Flaeche hinten */
/* 10 */   2  4  3
/* 11 */   6  8  5          /* Flaeche vorne */
/* 12 */   8  6  7

```

8.2.2 Aufbau der Kalibrierungsdateien

Die Kameradatei .cam

Für die Durchführung einer Kalibrierung sind diverse Herstellerangaben zu der Kamera erforderlich. Diese sind z.B. für die SONY D XC-730P 3CCD Videokamera bereits in der Datei *sonyccd.cam* vorhanden:

```
SensorDim  8.8 6.6          /* Groesse des Sensors in mm */
SensorSize 786 581          /* Anzahl der Bildelemente in Pixel */

SensorUsed 512              /* zur Digitalisierung verwendet */

ImageSize 512 512          /* Anzahl der Bildzeilen und -spalten */
ImageCenter 256 256         /* Bildmittelpunkt */
```

Bei den nachfolgenden ASCII-Text-Dateien ist auf eine korrekte Eingabe zu achten, da kein spezieller Syntaxcheck erfolgt. Außer Leerzeichen, Zeilenumbrüchen und Tabulatoren sind keine weiteren Kommentare, Buchstaben oder Sonderzeichen erlaubt.

Die Weltkoordinaten .wrld

Die Koordinaten im 3D-Weltkoordinatensystem werden in der Einheit *mm* angegeben. Nach der Anzahl der vorhandenen Kalibrierungspunkte *n* folgt die Liste mit den 3D-Punkten:

$$\begin{matrix} n \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{matrix}$$

Die Bildkoordinaten .img

Die korrespondierende Zeilen- und Spaltenposition eines Kalibrierungspunktes im Kamerabild wird in (Sub-)Pixel angegeben, wobei der Ursprung in der linken unteren Ecke plaziert ist.

$$\begin{matrix} n \\ x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{matrix}$$

Die Punktanzahl *n* und die Reihenfolge der Kalibrierungspunkte müssen in beiden Dateien übereinstimmen. Die Koordinaten können dabei als ganze Zahlen oder auch als Fließkommazahlen eingegeben werden.

Die Kalibrierungsdatei .cal bzw. .opt

Das binäre Dateiformat für die lineare Kalibrierung und für die optimierten Parameter ist identisch. Durch unterschiedliche Namensgebung im Suffix kann jedoch der aktuelle Bearbeitungszustand hervorgehoben

werden. Da die Ergebnisse in der Kalibrierungsdatei nicht menschenlesbar sind, wurde für die Weiterverarbeitung in fremden Programmen ein eigenständiges Beispielprogramm namens *readcal* geschrieben, welches den Umgang mit den kalibrierten Parametern verdeutlichen soll.

Es liest eine Kalibrierungsdatei ein und demonstriert den Zugriff auf die Strukturen, die anschließend auf dem Bildschirm ausgegeben werden. Die *Koordinatentransformation* zwischen den Weltkoordinaten und den Bildpositionen wird an mehreren Beispielen vorgeführt. Neben der Projektion eines Weltpunktes in das Bild wird auch die Rücktransformation eines Bildpunktes bei bekannter Tiefe im 3D-Kamerakoordinatensystem sowie der Schnitt eines Sichtstahls mit einer bekannten Ebene behandelt.

8.2.3 Format der Gauwertbilder .pgm

Bei den Bildern haben wir uns für das sehr verbreitete *Portable BitMap (PBM)* Format entschieden. Die Programme lesen und schreiben Grauwertbilder in dem speziellen *PBM P5-Format* (d.h 256 Graustufen in binärer Kodierung), wobei entgegen der Spezifikation keine Kommentare im Header erlaubt sind. Der Kopf eines $w \times h$ großen Bildes lautet im ASCII-Format:

```
P5
w   h   255
...
```

Nach dem maximalen Grauwert von 255 folgt ein Zeichen für den Zeilenumbruch und anschließend sind die Grauwerte mit einem Byte pro Bildpunkt binär abgelegt.

8.2.4 Format der Bilddaten I8 und F8

Die Formate I8 und F8 sind in ihrem strukturellen Aufbau identisch mit dem oben beschriebenen PBM P5-Format. Die Unterschiede liegen in der Kennung (I8 bzw. F8 statt P5) und der Speichergröße eines Bildpunktes. Im I8-Format wird anstelle eines Bytes ein *Integer* und im F8-Format ein *Float* pro Punkt abgelegt. Die Größen der beiden Typen ist systemabhängig.

8.2.5 Format der Verschiebungsvektorkarten .dpm

Für die Verschiebungsvektoren des Optischen Flußes wird das Dateiformat der Implementation von *Barron und Fleet* [BAR92] verwendet. Alle Daten werden dabei als binäre Fließkommazahlen gespeichert. Da der durch Filterung entstehende Rand mit der Breite *off* weggelassen wird, beschreibt der Eintrag *tot* die ursprüngliche Dimension der Karte und *num* die Anzahl der effektiv berechneten Nadeln. Nach den sechs Einträgen für den Kopf folgt eine $num_x \times num_y$ lange Liste mit den Verschiebungsvektoren (*u*, *v*):

<i>tot_x</i>	<i>tot_y</i>
<i>num_x</i>	<i>num_y</i>
<i>off_x</i>	<i>off_y</i>
<i>u₁</i>	<i>v₁</i>
<i>u₂</i>	<i>v₂</i>
\vdots	\vdots
<i>u_n</i>	<i>v_n</i>

So beschreibt z.B der Verschiebungsvektor (1, 1) eine diagonale Bewegung im Bild um $\sqrt{2}$ pix nach rechts oben. Als Markierung für Positionen, an denen keine Bewegung festgestellt werden konnte, wird der Wert 100.0 eingesetzt.

8.2.6 Format der Tiefenkarten .dep

Die berechneten Tiefenkarten enthalten nicht die absolute Entfernung im Kamerakoordinatensystem. Für eine bessere Visualisierung werden alle Tiefenwerte in den Ursprung des Koordinatensystems verschoben. Dazu wird die Z-Komponente des kalibrierten Translationsvektors T_z konstant von allen Entfernungen abgezogen. Der lesbare Kopf einer $w \times h$ großen Tiefenkarte lautet

$\#DEP$	
w	h
d_1	
d_2	
⋮	
d_n	

wobei die binären Tiefenwerte d_i in Fließkommadarstellung mit doppelter Präzision gespeichert werden.

Abbildungsverzeichnis

1.1 Einordnung des Verfahren	5
2.1 Kamerageometrie	9
2.2 Radiale Linsenverzerrung	11
2.3 Abbildungsprozeß des verwendeten Kameramodells.	12
2.4 Beispiele der synthetischen Generierung von Kalibrierungspunkten.	20
2.5 Fehler der Kalibrierung bei variabler Punktanzahl pro Ebene.	23
2.6 Kalibrierungsfehler bei unterschiedlich vielen Ebenen.	23
2.7 Fehler durch Verrauschen der Kalibrierungsdaten.	24
2.8 Kalibrierungsfehler bei unterschiedlichen Brennweiten.	24
2.9 Fehler in Abhängigkeit der radialen Verzerrung.	25
2.10 Beispiele realer Kalibrierungsbilder mit eingeblendetem Drehteller-Raster.	25
3.1 Kontrastanhebung durch globale Grauwertanpassung.	30
3.2 Interpolation der lokalen Schwellwerte.	31
3.3 Anwendung des dynamischen Schwellwertes.	31
3.4 Histogrammegalisierung mit verschiedenen Fenstergrößen	32
4.1 Digitalisierungseffekt am Beispiel einer Kante	34
4.2 Subpixelgenaue Punktebestimmung	36
4.3 Perspektivische Verzerrung von Ellipsen	36
4.4 Gaußfunktion mit Standardabweichung $\sigma = 2$	37
4.5 LoG-Operator mit Standardabweichung $\sigma = 1.5$	38
4.6 Merkmalsextraktion mit dem modifizierten Marr-Hildreth-Operator.	40
4.7 subpixelgenaue Nullstellen	41
4.8 Maximumssuche anhand der Gradientenrichtung.	43
4.9 Masken für indirekte 8-Nachbarschaftsverbindung.	44
4.10 Masken zur Überprüfung der Skeletteigenschaft.	45
4.11 Masken für die Bestimmung von Endpunkten.	45
4.12 Ergebnis der Merkmalsextraktion mit dem Canny-Kantendetektor.	47
4.13 Darstellung der Rauschempfindlichkeit des Monotonieoperators	50
4.14 Die Ergebnisse der Corner Response Funktion	51
4.15 Punktformige Merkmale	51
5.1 Epipolarkonstraint	57
5.2 Zylinderkonstraint	58
5.3 Motionkonstraint	59
5.4 Ausgezeichnete Verschiebungsvektoren	61

5.5	Verknüpfungstrategie von Verschiebungsvektorkarten	62
5.6	Verschiebungsvektoren berechnet aus Kantenbildern	62
5.7	Bilinlear interpolierte Verschiebungsvektorfelder	64
6.1	Kartesisches- und Zylinderkoordinatensystem	70
6.2	Tiefenberechnung ohne Kenntnis der Rotation	72
7.1	Grauwertbilder von synthetisch generierten Objekten.	74
7.2	Durch Simulation bestimmte ideale Verschiebungsvektoren.	74
7.3	Aus idealen Verschiebungsvektoren rekonstruierte Tiefenkarte.	75
7.4	3D-Visualisierung der rekonstruierten Tiefenkarte.	75
7.5	Das Ergebnis der Gestaltserkennung bei idealen Verschiebungsvektoren.	76
7.6	Störung der Verschiebungsvektoren mit Standardabweichung $\sigma = 0.5$	76
7.7	Rekonstruktion bei verrauschten Verschiebungsvektoren mit $\sigma = 0.15$	77
7.8	Rekonstruktion mit gestörten Verschiebungsvektoren bei $\sigma = 0.5$	77
7.9	Ergebnis nach Gaußglättung der Tiefenkarte mit einer 7×7 Maske.	78
7.10	Anwendung von Optical-Flow Verfahren auf synthetischen Würfel nach Rotation um 1grd.	79
7.11	Einsatz von Texturen zur Verbesserung der Verschiebungsvektorbestimmung.	79
7.12	Anwendung der Optical-Flow Verfahren bei texturierten Oberflächen.	80
7.13	Rekonstruktion einer realen Szene unter Verwendung des Optical-Flow Verfahrens von <i>Anandan</i> nach einer 5 <i>grd</i> Drehung.	81
7.14	Gestaltsrekonstruktion einer Würfelszene	84
7.15	Gestaltsrekonstruktion einer Realsszene	85
8.1	Datenfluß der Kalibrierung	91

Literaturverzeichnis

- [BRO87] **Bronstein, I.N.** und **Semendjajew, K.A.**: *Taschenbuch der Mathematik*, Verlag Harri Deutsch, 1987
- [ENG88] **Engeln-Müllges, G.** und **Reuter, F.**: *Formelsammlung zur Numerischen Mathematik mit MODULA 2-Programmen*, BI-Wiss.-Verl., 1988, S.36-40.
- [FEL88] **Fellner W.D.**: *Computer Grafik*, BI-Wiss.-Verl., 1988
- [FOL90] **Foley J.D., van Dam A. u.a.**: *Computer Graphics Principles and Practice*, Addison-Wesley, 1990
- [HAB89] **Haberäcker, P.**: *Digitale Bildverarbeitung: Grundlagen und Anwendungen*, Carl Hanser Studienbücher, 3. Auflage 1989.
- [HOR86] **Horn, B.K.P.**: *Robot Vision*, McGraw-Hill Book Company, New York 1986.
- [BJA93] **Jähne, B.**: *Digitale Bildverarbeitung*, Springer Verlag, 3. Auflage 1993.
- [KLE92] **Klette, R.**: Skripte zur Vorlesung *Computer Vision III* (1991) und *Computer Vision II* (1992)
- [STE93] **Steinbrecher, R.**: *Bildverarbeitung in der Praxis*, Oldenbourg Verlag, 1993.
- [NRC92] **W.H.Press, S.A.Teukolsky, W.T.Vetterling, B.P.Flannery**: *Numerical Recipies in C*, Cambridge university press, Second Edition 1992.

Aufgabenteilung

Aufgrund der ausgezeichneten Zusammenarbeit ist eine strikte Aufteilung von Teilaufgaben auf die Autoren nicht möglich. Alle theoretischen Grundlagen und auftretenden Probleme wurden gemeinsam erarbeitet bzw. gelöst. Daher richtet sich die geforderte Gliederung im folgenden hauptsächlich nach der schriftlichen Ausarbeitung.

Volker Rodehorst ist für das Kapitel 2 der Kamera-Kalibrierung und die ersten drei Abschnitte der Vorverarbeitung in Kapitel 3 verantwortlich. Die Beschreibung der Operatoren von Marr-Hildreth und Canny in Kapitel 4 sowie der Optical-Flow-Techniken wurden ebenfalls von ihm durchgeführt. Schließlich ist er noch für den Simulator zur synthetischen Verschiebungsvektorbestimmung und für die Visualisierung der Daten zuständig.

Dirk Mehren hat sich neben der Filterung in dem Kapitel 3 der Vorverarbeitung besonders mit dem Problem der Subpixelgenauigkeit, dem Monotonie-Operator und der Corner Response Funktion bei der Merkmalsextraktion in Kapitel 4 beschäftigt. Er ist ebenfalls zuständig für die Bestimmung und Interpolation ausgezeichneter Verschiebungsvektoren in Kapitel 5 sowie für das in Kapitel 6 beschriebene Verfahren zur Tiefenbestimmung.

Danksagung

An dieser Stelle möchten wir noch Herrn Prof. Reinhard Klette unseren Dank für die fachliche Anleitung und ständige Hilfsbereitschaft aussprechen. Der Dank gilt ebenso allen Mitarbeitern des Fachbereichs Computer Vision an der Technischen Universität Berlin sowie allen Freunden, die mit konstruktiver Kritik und Korrekturlesen an der Gestaltung dieser Arbeit mitgewirkt haben. Insbesondere möchten wir unseren Eltern danken, die uns dieses Studium überhaupt erst ermöglicht haben.