Qualitative Lokalisation und Rekonstruktion flexibler Szenenobjekte zum Zwecke des Greifens

Diplomarbeit

am Lehrstuhl für Kognitive Systeme
des Instituts für Informatik und Praktische Mathematik
der Christian–Albrechts–Universität zu Kiel
Prof. Dr. G. Sommer

vorgelegt von

Marco Matthias Päschke

Betreuer: Dr. Josef Pauli

Kiel, im Oktober 1999

Zusammenfassung

Ausgehend von der Motivation in komplexen Szenen (z.B. Elektronikschrott) Kabelleitungen zu detektieren und demontieren, beschreibt diese Arbeit ein System zur Lokalisation und Rekonstruktion von flexiblen Objekten mit der Zielsetzung, diese mit einem Roboterarm greifen zu können. Hierzu wird ein *generisches Modell* (Splines) zur Beschreibung der Objekte zugrunde gelegt. Ein kalibriertes Stereokamerasystem nimmt die Szene auf. Die Stereobilder werden parallel verarbeitet, die 2D-Rekonstruktionsergebnisse im Bild durch ein Stereomatchingverfahren in 3D-Koordinaten überführt, und die Ergebnisse aus verschiedenen Sichten in einem 3D-Besetztheitsgitter akkumuliert. Die Erkennung im Bild erfolgt in zwei Stufen. Zuerst werden farbbasiert Segmente aus der Szene extrahiert. Die Farbinformation bereits bekannter Objekte wird aus einer Histogramm-Datenbank genommen, während ein Segmentierungsverfahren neue Objekte im Bild detektiert. Im zweiten Schritt werden die Segmente formbasiert analysiert und im Sinne des generischen Modells auf Basis von Splines, soweit möglich, zu zusammenhängenden Objektverläufen gruppiert. Aus den in beiden Bilder detektierten Verläufen wird dann mittels der epipolaren Geometrie eine 3D-Rekonstruktion erstellt. Hierbei werden die Ergebnisse mehrerer Sichten in einem 3D-Besetztheitsgitter integriert, so daß eine Zugriffsposition für den Greifer bestimmt werden kann. Ein Anwendungsbeispiel des Systems ist die Demontage von Kabelleitungen an einem Monitor.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Motivation und Zielsetzung

Moderne, flexible industrielle *Montage* oder - zukünftig unvermeidlich - auch *Demontage* von Gütern durch Robotersysteme bietet ein breites Einsatzgebiet für *visuellbasierte Robotiksysteme*. Im konstruktiven Fall (Montage) z.B. als Überwachungssystem zur Fertigungsablauf- und auch Qualitätskontrolle. Im "destruktiven Fall" (Demontage) liegen die Aufgaben dieser Systeme sowohl in der Erkennung der zu demontierenden Objekte, als

Abbildung 1.1: Elektronikschrott

auch der Rekonstruktion derselben, so daß ein Greifen durch das Robotersystem möglich ist. Sie besitzen gegenüber den herkömmlichen industriellen Robotersystemen *ohne* Sensoren den Vorteil, flexibel auf Veränderungen der (Arbeits=)Umwelt reagieren zu können. Speziell im Falle der Demontage ist der Einsatz von Sensoren unumgänglich, da die genaue Lage der zu demontierenden Objekte vorab in der Regel nicht bekannt ist.

Idee und Motivation dieser Arbeit lieferte die in Abbildung 1.1 dargestellte Szene. Zielsetzung ist die Extraktion der Kabel und ihre 3-dimensionale Rekonstruktion zum Zwecke des Greifens, der Demontage. Der folgende Abschnitt schildert die daraus resultierenden Problemstellungen.

1.2 Problemfelder und Objektdefinition

Die Zielsetzung lautet: Erkennung und Rekonstruktion der Objekte vom Typ Kabel. Es ist also eine *Objektbeschreibung* gesucht, welche die Kabel (oder ähnlich geartete Objekte) charakterisiert. Hier ergibt sich nun das Problem, daß eine einfache geometrische Beschreibung durch Figuren wie Polyeder, Polygonzüge oder dergleichen nicht möglich ist. Es existiert *keine fixe a-priori Information* über die Form eines Kabels. Auftreten in der Szene unterscheiden sich erheblich in Größe, Ausdehnung, Lage und Orientierung. Wir werden Kabel deshalb im Rahmen dieser Arbeit durch eine generisch-geometrische Beschreibung charakterisieren und als *im Verlauf glatte, annähernd 1-dimensionale Objekte mit einer bzgl. des Farbtones homogenen Erscheinung* auffassen.

Zweites Problem, wie man Abbildung 1.1 entnehmen kann, ist die *Komplexität der Szene*. Die Vielzahl verschiedener Objekte und das Fehlen eines eindeutigen Hintergrundes zur Separierung der Objekte erschwert die Erkennungsaufgabe erheblich. Eng mit der Komplexität der Szene verbunden ist auch das Problem der *Verdeckungen*. Die Kabel verlaufen vor und hinter anderen Objekten, so daß ihr vollständiger Verlauf nicht unmittelbar 2-dimensional im Bild zu rekonstruieren ist.

Ein weiteres, nicht zu vernachlässigendes Problemfeld ist die *Bildqualität*. Der Einsatz von Farbkameras erhöht die Anfälligkeit gegenüber Signalausfällen, da jedes der 3 Farbbänder (drei Typen von Farbsensoren) von Hardwareseite individuellen Störungen ausgesetzt ist. Die dabei entstehenden Bildstörungen werfen ähnliche Probleme auf wie die bereits genannten Verdeckungen.

Abbildung 1.2 zeigt einen Ausschnitt aus einer typischen Szene, in der die oben genannten Probleme noch einmal hervorgehoben sind.

Als Endergebnis soll eine Rekonstruktion des Kabels stehen, derart, daß ein Greifen mit einem Roboterarm möglich ist. Dieses setzt den Einsatz eines *kalibrierten Stereokamerasystems* voraus.

Welche Verfahren man auch immer zur Objekterkennung wählt, sie liefern aus den oben

Abbildung 1.2: Probleme der Verdeckungen und Bildqualität bei der Objekterkennung

genannten Gründen keine *exakten*, sondern nur *ungefähre, angenäherte* Ergebnisse. Da das 3D-Rekonstruktionsverfahren auf diesen Daten basiert, ist auch die Rekonstruktion - verstärkt noch durch Kalibrierungsfehler - fehlerbehaftet. Ein exaktes Erkennen und Rekonstruieren der Kabel ist somit nicht möglich.

Man findet in der Literatur eine Reihe von Ansätzen zur Behandlung von gebogenen (nicht-polygonen) Formen. Kurvige Bildelemente (*curved image features*) als Bildprimitiva aufgefaßt, findet man bei [Blaszka94]. Dort werden parametrische Modelle zur Beschreibung solcher Formen verwendet. Der Großteil der Literatur propagiert den Einsatz der generalisierten (hoch dimensionalen) Hough-Transformationen zur Lokalisation von Freiformobjekten [Samal97], [Ser95]. Einen spezielleren Ansatz aus diesem Bereich findet man bei [Kender91]. Hier werden mittels eines 7-dimensionalen Hough-Raumes torodiale Körper beschrieben und lokalisiert. Als dritte Klasse von Lösungsansätzen sind die neuronalen zu nennen. In [Spirkovska94] finden sich Ansätze zur 2D- und 3D-Erkennung mit high-order neuronalen Netzen (HONN‘s), während [Seibert91] einen allgemeinen Ansatz zur Erkennung von 3D-Objekten auf Basis neuronaler Netze vorstellt.

All diesen Ansätzen ist gemeinsam, daß sie die reine geometrische Form der Objekte beschreiben und lokalieren. Es sind im Sinne einer fixen geometrischen Beschreibung exakte Verfahren. In unserem Fall ist die Beschreibung durch eine starre geometrische Form, wie bereits erläutert, nicht möglich. Es ist also ein vollständig anderes Vorgehen gefordert.

Die Überlegungen im folgenden Abschnitt sollen aufzeigen, daß der “Verzicht” auf die Exaktheit, auch ein Weg zum Ziel, Greifen des Objektes, sein kann.

1.3 Qualitatives Lokalisieren und Rekonstruieren

"It's vain to do with more what can be done with less"
William of Occam (13.Jh.) - *Occam's-Razor*

Dieses ökonomische Grundprinzip ist aus der modernen Sichtweise des *purposive, qualitive vision* ([Aloimonos93]) nicht mehr wegzudenken. Methoden und Werkzeuge so zu wählen, daß das Ziel erreicht wird, und kein darüber hinausgehender Aufwand betrieben wird, ist auch Grundidee dieser Arbeit. Die Anwendung dieses Grundsatzes ist dabei wie folgt motiviert:
Im vorangegangen Abschnitt wurde bereits dargelegt, daß -und vor allem warum- der Ansatz einer exakten Rekonstruktion fehlschlägt, gesucht ist also eine angenäherte Lösung. Unter Hinzunahme der Zielsetzung - das Gesamtsystem soll Kabel erkennen und rekonstruieren, um sie *Greifen* zu können - folgt, daß eine Rekonstruktionsgenauigkeit im Rahmen der Ausmaße des Greifers ausreichend ist. Da die zum Greifen verwendeten Endeffektoren eine relativ weiten Zugriffsbereich bieten (z.B. ca. 5 cm), kann eine Abweichung von ca. 2 cm toleriert werden (siehe dazu Abbildung 1.3). Trotz der verschiedenen Rekonstruktionsergebnisse ist das Greifen des Kabels jeweils möglich.

Statt der Exaktheit im Erfolgsfalle, wie bei geometrischen Verfahren (z.B. Hough- Transformation), steht bei den im Rahmen dieser Arbeit verwendeten Verfahren die stabile Approximation in allen (oder zumindestens möglichst vielen) Fällen im Vordergrund.

Wir gehen daher bei der Erkennung und Rekonstruktion in zwei großen Schritten vor. Anfangs wird das reine Bild*signal*, die *photometrische Information*, untersucht. Ist dieses geschehen, so werden die so erhaltenen Ergebnisse *geometrisch* analysiert. Sämtliche

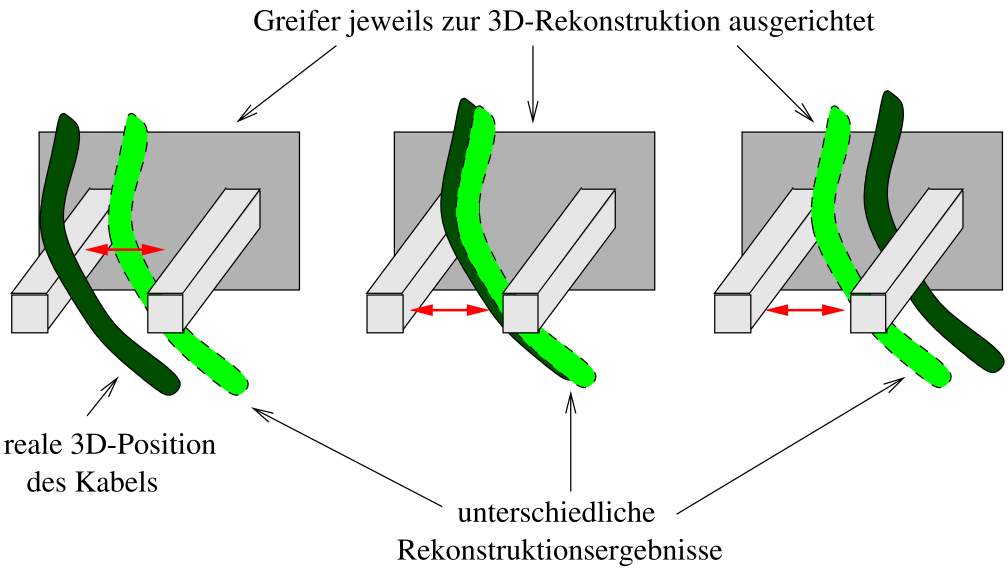


Abbildung 1.3: Zugriff bei unterschiedlichen Rekonstruktionsergebnissen

vorgestellte Verfahren agieren defensiv [1], so daß sichergestellt ist, das nur photometrische Information, die geometrisch bestätigt ist, also ein Konsens zwischen beiden vorliegt, als Ergebnis akzeptiert wird; Ziel ist es dabei die Fehleranfälligkeit zu mindern. Daß dabei möglicherweise auf (riskante/unsichere) Information verzichtet werden muß, wird bewußt in Kauf genommen.

Ein solches, defensives Vorgehen bedeutet auch, daß die Rekonstruktion im allgemeinen nicht aus einer Sicht der Szene erfolgen kann, sondern eine Reihe von Ansichten benötigt wird. Idee ist natürlich, daß die geeignete Summe/Aggregation relativ sicherer Information eine stabile, möglichst fehlerarme Rekonstruktion liefert.

Im nächsten Abschnitt wird der strukturelle Aufbau des Gesamtsystems einleitend erläutert, ohne daß dabei genauer auf die einzelnen Verfahren eingegangen wird. Dieses erfolgt später.

1.4 Eine kurze Gesamtübersicht über das System

An dieser Stelle soll eine kurze thematische Übersicht über den Gesamtaufbau des vorgestellten Systems gegeben werden. Abbildung 1.4 zeigt den Aufbau graphisch.

Ein kalibriertes Stereokamerasystem liefert Bilder einer Szene. In diesen sollen jeweils die Kabel erkannt und markiert werden. Hierzu wird zuerst farb- und dann formbasiert vorge-

[1] defensiv soll hier im Sinne von vorsichtig, zurückhaltend verstanden werden

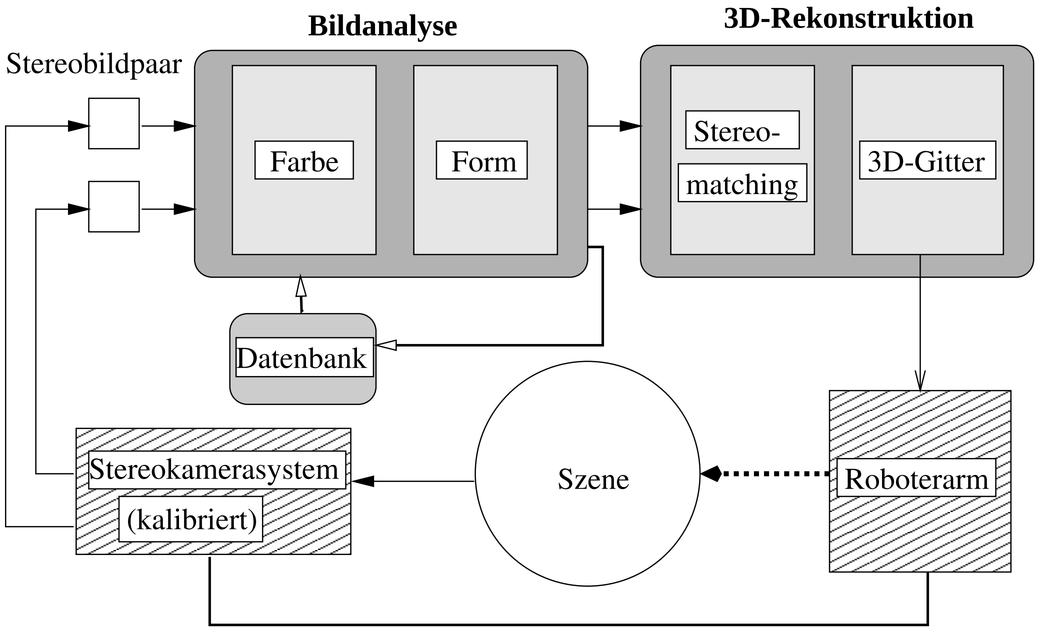


Abbildung 1.4: Thematisches Grobschema des Gesamtsystems

gangen. Die Bilder werden dabei farbbasiert parallel auf zwei Arten analysiert; sie werden segmentiert, und gleichzeitig wird bereits vorhandene, vorab initialisierte oder in einem vorherigen Durchlauf gewonnene Farbinformation, zur Lokalisation bestimmter Kabel im Bild benutzt. Die beiden Ergebnisse werden geeignet kombiniert, so daß das Resultat Flächensegmente sind, die im zweiten Teil auf ihre Form hin untersucht werden. Dabei wird versucht, die Segmente so zu verbinden, daß möglichst zu einem Kabel gehörende Segmente gefunden werden.

Danach werden die in beiden Stereobildern gefundenen Kabel mittels epipolarer Geometrie verglichen und zu den korrespondierenden Bildpunkten die 3D-Position rekonstruiert. Die entstehenden Weltpunkte werden in ein 3D-Besetztheitsgitter eingetragen.

So entsteht bei Aggregation mehrere Sichten in diesen Gitter eine (möglicherweise fragmentierte) Rekonstruktion der Kabel. Anhand dieser kann eine Zugriffsposition für den Greifer ermittelt werden.

Der abschließende Abschnitt der Einleitung liefert einen Überblick über die Gliederung der weiteren Arbeit und die wichtigsten Literaturquellen.

1.5 Aufbau der Arbeit und Literatur

In der gesichteten Literatur wurde kein dem hier entwickelten System in seiner Gesamtheit vergleichbarer Ansatz gefunden. Es werden daher parallel zur Schilderung des Aufbaus die jeweils wichtigsten, relevanten Literaturstellen aufgeführt.

Das folgende Kapitel 2 *Roboter-Vision-System und Hardware* beschreibt die Grundkonzeption eines Roboter-Vision-Systems und erläutert Aufgaben und Interaktion der einzelnen Komponenten. Es wird die verwendete Hardware vorgestellt und abschließend der reale Versuchsaufbau beleuchtet.

In Kapitel 3 *Farbbasierte Objekterkennung* werden die Grundlagen der *Farbräume* HSV und RGB kurz erläutert, und darauf aufbauend eine *Farbhistogramm-Datenbank* (basierend auf dem *Fuzzy-Art-Algorithmus* nach [CaGroRo91]) vorgestellt. Im Anschluß werden die Verfahren zur Lokalisation der Objekte (*Graphenbasierte Bildsegmentierung* nach [CoFish83] und [Niemann83] und *Histogramm-Rückprojektion* [SwainBallard91]) und die Kombination derer Ergebnisse dargestellt.

Abschnitt 3.4 ist der erste Teil dreier Beispiele zum Bereich Objekterkennung. Es wird die Analyse dreier Szenen nach den vorgestellten, farbbasierten Methoden veranschaulicht.

Die *Formbasierte Objekterkennung* ist Inhalt des Kapitels 4. Hier werden zuerst *einfache Formmerkmale* (Seitenverhältnis der *Bounding-Box*, Ergebnis der Anwendung des *Thinning-Operators*) als Auswahlkriterien für Flächensegmente eingeführt. Zur Approximation der Segmente werden *Kohonen-Netze* basierend auf [Kohonen82] verwendet, deren Verlauf durch eine einfache Anwendung des *Kalman Filters* [Zhang95], [Kalman60] geglättet wird. Abschließend wird *das* generische Modell zur Approximation von Kabelverläufen, *Splines*, als Basiskriterium für einen Kabeltest gewählt.

Abschnitt 4.5 ist der zweite Teil der Beispiele zum Bereich Objekterkennung. Die Segmentierungsergebnisse aus Teil 1 werden hier mittels der Verfahren aus Kapitel 4 exemplarisch analysiert.

Nach der Objekterkennung im Bild folgt in Kapitel 5 die Beschreibung der *3D-Rekonstruktion mittels der epipolaren Geometrie* [Faugeras93]. Hierbei wird die praktische Kamerakalibrierung, das zur Bildentzerrung verwandte Verfahren nach [Lenz88] und dazu ein Beispiel beschrieben. Im zweiten Teil werden dann die Grundlagen der epipolaren Geometrie und die Vorgehensweise zum Auffinden der Stereokorrespondenzen erläutert. Kapitel 5.3 zeigt beispielhaft die ermittelten Stereokorrespondenzen für drei Sichten einer Monitorszene.

In Kapitel 6 *3D Besetztheitsgitter zur Objektrekonstruktion* werden 3D-Gitter nach einer Idee von [Moravec96] als Unterabtastung des Raumes vorgestellt, deren Zellen additiv über die Zeit die Besetztheitsinformation sammeln, und so eine 3D-Rekonstruktion der Kabelobjekte liefern. Die Ergebnisse werden in Abschnitt 6.3 anhand der Fortführung des Beispiels aus Abschnitt 5.3 illustriert.

Als Brückenschlag zur Motivation der Arbeit wird in Kapitel 7 ein prototypisches System zur Demontage von Kabelleitungen an Monitoren vorgestellt.

Zusammenfassung und Ausblick in Kapitel 8 beschließen die Arbeit.

Kapitel 2

Roboter-Vision-System und Hardware

Im ersten Abschnitt wird der grundlegende konzeptionelle Aufbau eines Roboter-Vision-Systems zum Rekonstruieren und Greifen von Kabelobjekten eingeführt. Danach folgt die Beschreibung der Hardware und die Darstellung des realen Versuchsaufbaus.

2.1 Aufbau des Roboter-Vision-Systems

Ausgangspunkt der Arbeit ist die Zielsetzung, ein System zu entwickeln, daß in komplexen Szenen, genauer Ansammlungen von Elektroschrott oder -geräten (z.B. Monitor), Kabelleitungen erkennen und greifen kann. Dieser Abschnitte stellt das System aus konzeptioneller Sicht vor. Abbildung 2.1 zeigt die dazu nötigen Teilmodule und ihre Interaktion.

Fokus des Systems ist die *Szene*. *Kamerasystem* und *Manipulator* bilden einerseits eine Einheit, da die Kameras am Manipulator befestigt sind, und deren Bewegung durch den Manipulator erfolgt, andererseits besitzen beide Einheiten auch separat ihre ureigensten Aufgaben der Bildaufnahme und des Greifens.

Aus der Szene soll mittels des beweglichen Kamerasystems Information gewonnen werden, dafür ist es notwendig, Positionen festzulegen, bzw. zu ermitteln, aus denen die Szene beobachtet wird. Diese Aufgabe übernimmt das Modul *Sichtengenerierung*. Im Rahmen dieser Arbeit wurden für die einzelnen Tests jeweils feste Positionen vorgegeben. Das Modul steuert den Roboterarm geeignet, löst die Bildaufnahme aus und leitet die Signale an das *Analyse*-Modul weiter.

Hier erfolgt die Bildverarbeitung, bestehend aus Objekterkennung und -lokalisation, gegebenenfalls unter Verwendung von a-priori- oder zyklisch erworbenem Wissen (Objektdefinition), im Bild, und die 3D-Rekonstruktion der Stereobilddaten unter Verwendung

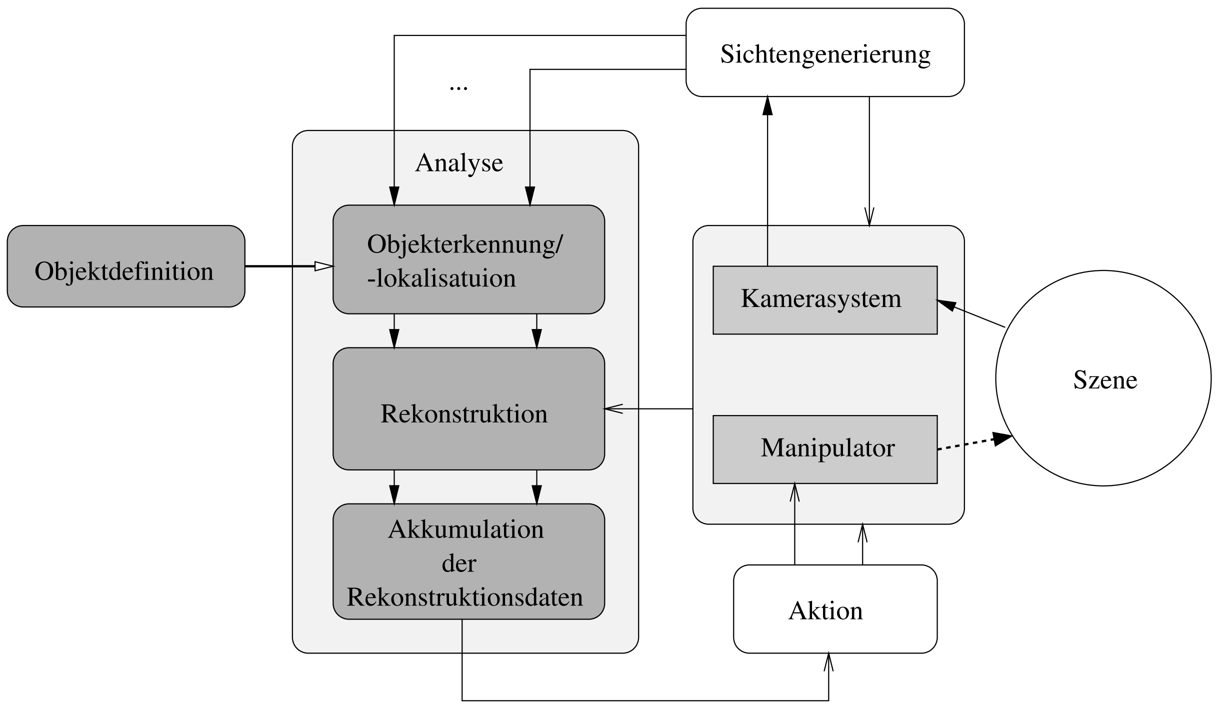


Abbildung 2.1: Konzeptioneller Aufbau des Roboter-Vision-Systems

der Kalibrierungsdaten der Kameras, sowie der Position des Roboterarms und damit des Kamerasystems. Die Möglichkeit, die Rekonstruktionsergebnisse über die Zeit/Sichten zu sammeln, ist hier zum Analyse-Modul gezählt; eine Ausgliederung als eigenständiges Modul wäre genauso zu motivieren.

Ist eine im Sinne der Aufgabe ausreichende Rekonstruktion des Objektes erfolgt, so kann die Szene wie gewünscht (in unserem Falle: Greifen des Objektes) verändert werden. Hierzu bestimmt das Modul *Aktion* eine Zugriffsposition und bewegt den Roboter zum Objekt hin. Dieser Bereich ist weniger Bestandteil dieser Arbeit, und zu Testzwecken nur rudimentär und exemplarisch unter Vernachlässigung von Aspekten der Hindernisvermeidung implementiert.

Betrachtet man den ganzen Ablauf, so findet man den Aufbau eines *Wahrnehmungs-Aktions-Zyklus*. Die Module Kamerasystem, Sichtengenerierung und Analyse gehören zum Bereich Wahrnehmung, während Aktion und Manipulator den Aktionsteil darstellen.

Nach dem konzeptionellen Aufbau werden im folgenden Abschnitt die einzelnen Hardware-Komponenten des realen Versuchsaufbaus vorgestellt.

2.2 Die Hardware

Die wesentlichen Hardwareeinheiten sind : Roboterarm, Recheneinheit (Sun Enterprise 4000), Stereokamerasystem und Drehteller. Im Folgenden werden sie einzeln und in Kürze vorgestellt.

2.2.1 Der Roboterarm RX90

In dieser Arbeit wird der Roboterarm RX90 der Firma Stäubli verwendet. Er besteht aus sechs Gliedmaßen, die miteinander durch Gelenke verbunden sind. Jedes Gelenk stellt eine Achse dar, um die sich jeweils zwei Gliedmaßen drehen. Die Greifhand ist an der sechsten Gliedmaße befestigt. Sie besitzt 2 Finger, die sich öffnen und schließen lassen.

Die vorhandene Peripherie (im Steuerschrank CS7) löst die Probleme der inversen Kinematik, so daß man nicht explizit die Gelenkwinkel des Roboters angeben muß, um die Greifhand zu positionieren. Es genügt, einen Raumpunkt anzugeben. Dessen Positionierung unterliegt dabei 2 Koordinatensystemen, dem Basis- und dem Werkzeugkoordinatensystem, deren Lage in Abbildung 2.2 dargestellt ist.

Ein Raumpunkt wird hierbei immer durch sechs Parameter (X, Y, Z, *yaw*, *pitch*, *roll*) beschrieben. Hierbei ist (X, Y, Z) die *Entfernung* des Ursprungs des Werkzeugsystems zum Roboterbasiskoordinatensystem in Millimeter (*mm*), und (*yaw*, *pitch*, *roll*) die *Ori-*

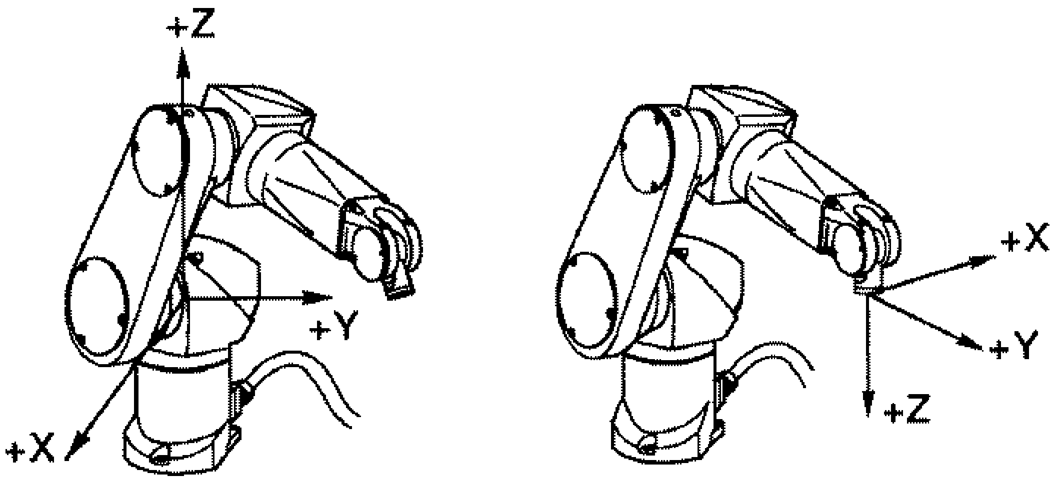


Abbildung 2.2: Die Koordinatensysteme BASE (Basissystem) und TOOL (Werkzeugsystem) des Manipulators

entierung des Werzeugsystems relativ zum Roboterbasiskoordinatensystem in Grad. Diese Orientierung wird in der Eulernotation angegeben. Die Orientierung beträgt genau dann Null Grad in jeder Richtung, wenn die Hand senkrecht nach oben zeigt, und die Handfläche parallel zur y-z-Ebene ist. Der Ursprung des Werkzeugkoordinatensystems befindet sich üblicherweise in der Mitte der Werkzeugflansch, an der die Greifhand befestigt ist.

2.2.2 Die Recheneinheit

Die Berechnungen für Sichtengenerierung, Aktion und insbesondere das Analyse-Modul erfolgen auf einem Mehrprozessorrechner des Herstellers SUN, einer Enterprise 4000, ausgestattet mit vier ULTRASPARC 167 MHz Prozessoren. Die sich daraus ergebende Möglichkeit, die Rechnungen zu parallelisieren, wurde hauptsächlich für die Bildverarbeitungsoperationen im Rahmen des Analyse-Moduls genutzt. Hier kann im Vergleich zu einer Einprozessor-Maschine eine Reduzierung der Berechnungszeiten um den Faktor ≈3,6 erzielt werden.

2.2.3 Das Stereokamerasystem

Zur Bildaufnahme werden 1-CCD-Chip-Farbkameras ST-M1000 der Firma Stemmer verwendet. Sie haben eine Auflösung von 752 × 582 Pixeln. Die in dieser Arbeit genutzten quadratischen Bilder mit 512 × 512 Pixeln werden mittig (Abbildung 2.3(a)) aus dem Gesamtraster gelesen. Die Objektive weisen eine Brennweite von 5.6 mm auf. Abbildung 2.3(b) zeigt die Stereoanordnung der Kameras. Sie sind zwecks Entstörung auf einer Kunst-

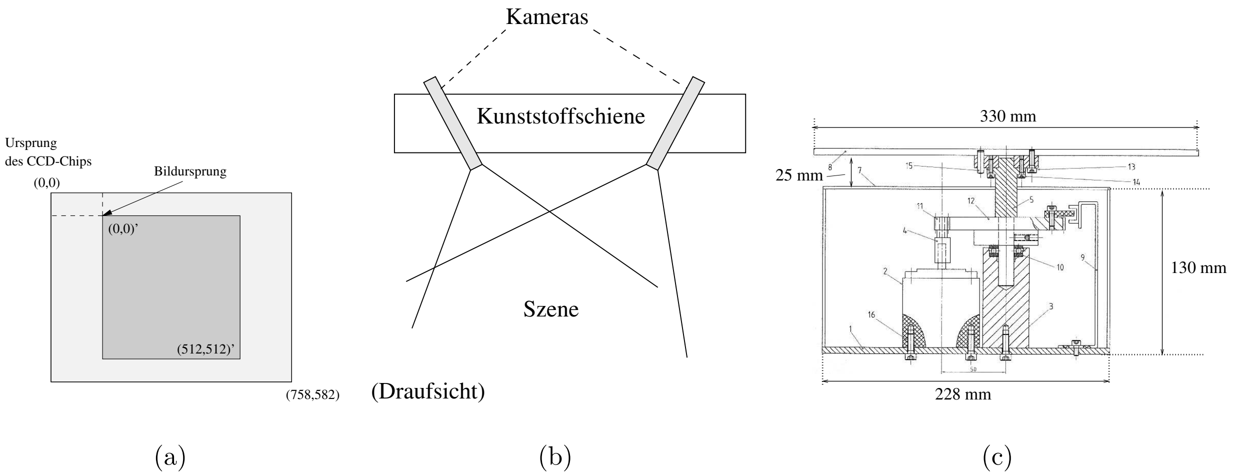


Abbildung 2.3: Bildausschnitt aus CCD-Kamera MT-M1000 (a), Stereoanordnung der Kameras (b) und technische Zeichnung des Drehtellers (c)

stoffschiene montiert.

2.2.4 Der Drehteller

Um einen weiteren Freiheitsgrad für die Sichtengenerierung zu erlangen, werden die betrachteten Objekte auf einem Drehteller positioniert. Abbildung 2.3(c) zeigt eine technische Zeichnung des lehrstuhlinternen Eigenbaus und seine Aussenmaße.

2.3 Der Versuchsaufbau

In Abbildung 2.4 ist der vollständige Versuchsaufbau dargestellt. Die Sun Enterprise 4000 steuert sowohl den Roboterarm als auch den Drehteller. Die direkte Ansteuerung des Roboterarms übernimmt der Steuerschrank, der wiederum die Position des Roboters an die Sun-Schnittstelle liefert. Auch die Ansteuerung des Drehtellers erfolgt nach demselben

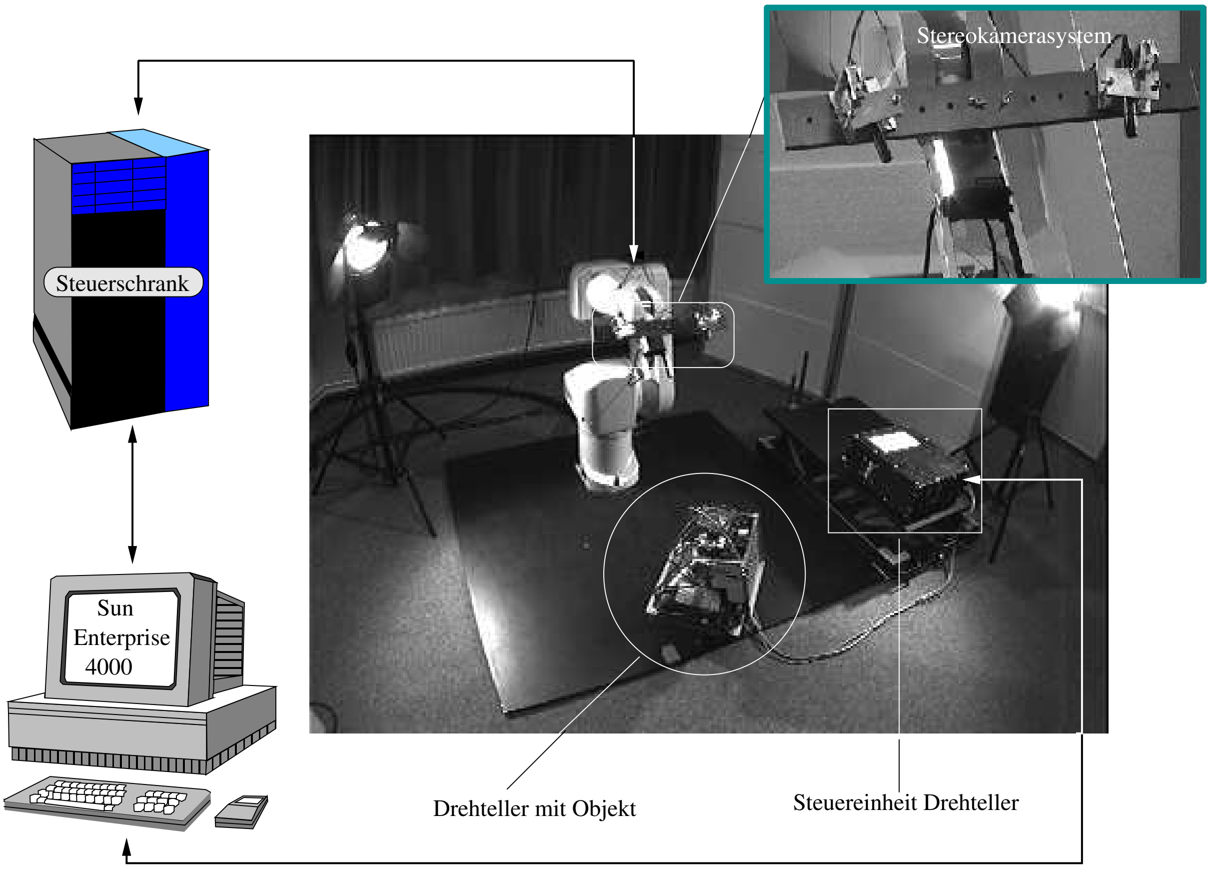


Abbildung 2.4: Der Versuchsaufbau des Gesamtsystems

Prinzip über eine Steuereinheit. Das Stereokamerasystem ist auf einer Kunstoffschiene am Roboterarm montiert, und kann somit durch diesen bewegt werden.

Kapitel 3

Farbbasierte Objekterkennung

3.1 Objekterkennung mittels Farbe und Form

Versucht man eine Klasse von Objekten zu beschreiben, so bedient man sich im Bereich der Bildverarbeitung typischer Merkmale wie Größe, Form oder auch Grauwerten. Hier-

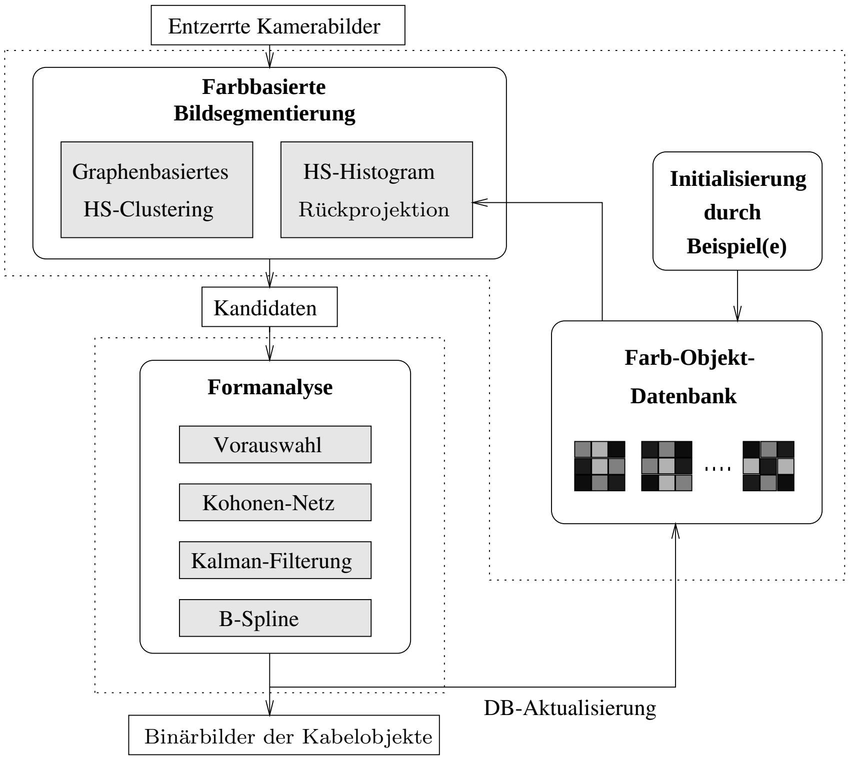


Abbildung 3.1: Die einzelnen Module der farb- bzw. formbasierten Objekterkennung und ihr Zusammenwirken

zu werden häufig mittels verschiedener Kantenextraktionsverfahren Repräsentationen der Objektumrandungen aus dem Bild gewonnen, diese dann anhand von geometrischen Merkmalen klassifiziert. Als Alternative sind auch *erscheinungsbasierte* Ansätze ([Murase93]) zu nennen, welche beispielhaft Ansichten der Objekte vorgeben und diese dann gegen das Bildsignal matchen und so das Objekt lokalisieren. Alle diese Ansätze benötigen eine genaue Objektbeschreibung, bzw. setzen voraus, das die Objekte sich in irgend einer Form von Hintergrund und anderen Objekten unterscheidet.

In unserem Fall werden Objekte (Kabel, Leitungen) gesucht, die weder in Form noch Größe bzw. Ausdehnung exakt faßbar sind. Der Hintergrund ist beliebig komplex und unterschiedlich ausgeleuchtet. Ein weiteres Problem sind Verdeckungen. Gesucht sind einfache Kriterien ein Objekt zu lokalisieren und zu erkennen. Hierbei soll genau soviel Aufwand betrieben werden, daß auf Basis dieser Ergebnisse eine zum Greifen ausreichend genaue Rekonstruktion möglich ist. Wir werden hierzu in zwei Schritten vorgehen:

1. Lokalisation der (möglichen) Kabel im Bild durch Farbinformation/-segmentierung
2. Erkennung der Objekte durch Analyse der Form.

Im ersten Schritt wird ein einfaches Farbsegmentierungsverfahren verwendet, um ohne Vorwissen in einem Bild zusammenhängende Regionen zu markieren; ist bereits Farbinformation vorhanden (zyklisch gewonnen oder durch Initialisierung), so wird diese mittels einer Histogramm-Rückprojektion zur Lokalisation im Bild genutzt. Die so gewonnenen Kandidaten, werden durch eine mehrstufige Formanalyse (Kapitel 4) als Kabel (oder gegebenenfalls nicht Kabel) klassifiziert. Im positiven Fall wird die Farbinformation dieses klassifizierten Objektes in eine Farb-Datenbank (bestehend aus Histogrammen) integriert. Aufbau und Update-Algorithmus der Datenbank sind in Abschnitt 3.2 beschrieben. Die Verfahren zur Lokalisation/Segmentierung sind in Abschnitt 3.3 erläutert.

Bestandteil dieses Kapitels sind auch drei kleinen Beispiele, die jeweils am Ende des ersten Teiles (Farbanalyse) in Abschnitt 3.4 und am Ende des Kapitels (Formanalyse) in Abschnitt 4.5 Anwendung und Ergebnisse der Methoden belegen.

3.2 Farb-Objektdatenbank

Die Objekte Kabel werden durch ihre Farbe bzw. eine geeignete Charakterisierung derselben beschrieben. Dem Aufbau des Systems entsprechend soll sowohl die Möglichkeit bestehen, Objektbeschreibungen vorzugeben (Initialisierung), als auch zur Laufzeit neue Objekte hinzuzufügen bzw. vorhandene Farbinformation zu adaptieren. Realisiert werden diese Fähigkeiten durch eine *Farb-Objekt-Datenbank* und einen der adaptiven Resonanztheorie nach [CaGro87] entliehenen Algorithmus zur Aktualisierung derselben. Eine Übersicht über das Vorgehen liefert Abbildung 3.2. Zur Initialisierung werden Beispielbilder

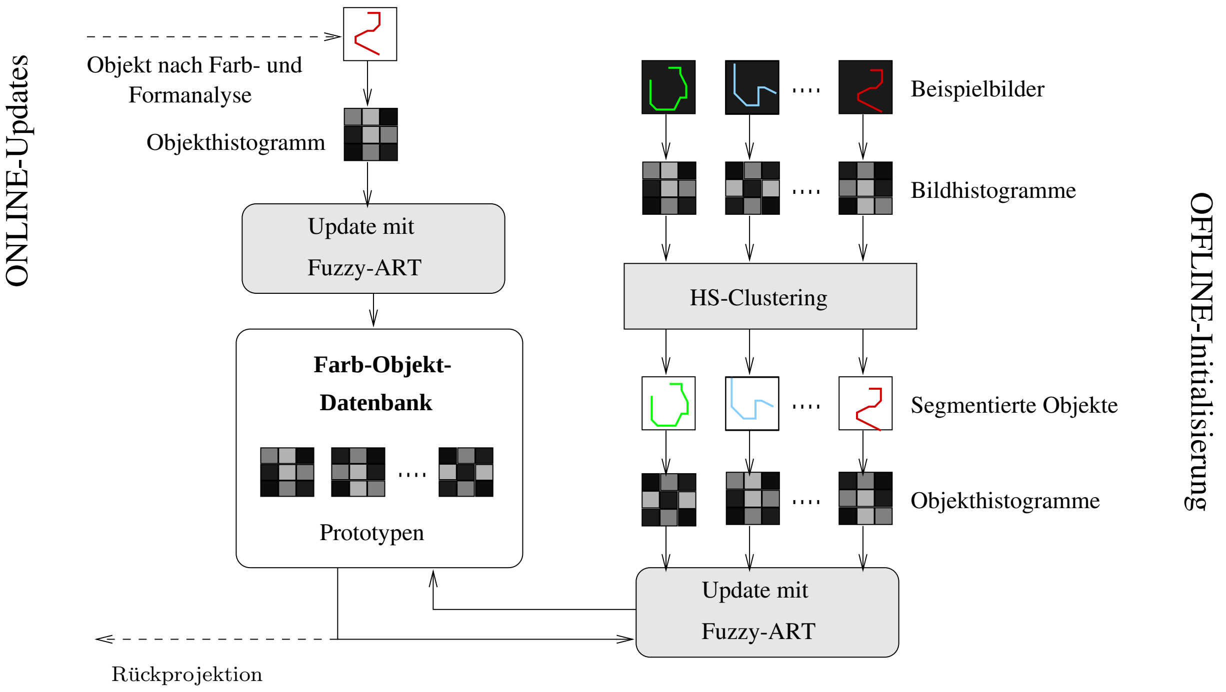


Abbildung 3.2: Farb-Objekt-Datenbank : Initialisierung und Aktualisierungen zur Laufzeit

von Kabelobjekten vorgegeben. Der Hintergrund ist hierbei homogen gewählt. Durch Segmentierung im Histogramm (siehe später Abschnitt 3.3.1) werden die Objekthistogramme gewonnen, und die Datenbank mit diesen aktualisiert. Zur Laufzeit werden neu gewonnene Farbobjekte auf dieselbe Art in die Datenbank integriert.

In den nächsten Abschnitten werden die verwendeten Farbmodelle (RGB, HSV), die Objektbeschreibung als zweidimensionales Histogramm und der Algorithmus zum Aktualisieren der Datenbank vorgestellt.

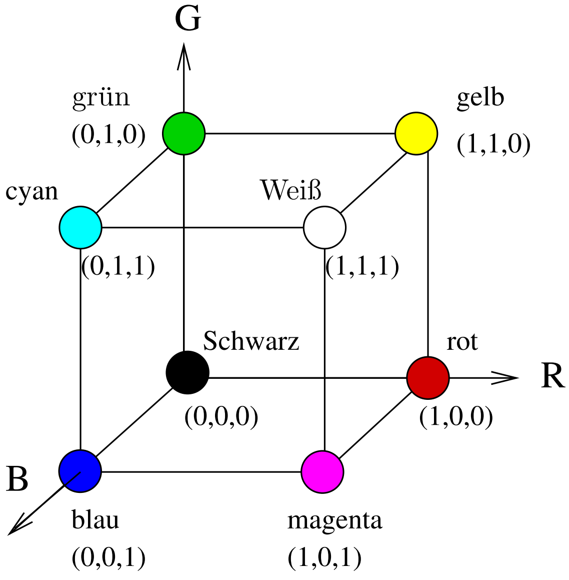
3.2.1 RGB- und HSV-Farbraum

Eine grundlegende Einleitung zum Thema Farbräume und Farbbildsegmentierung liefern [Rehrmann95] oder auch [Haberäcker95]. Im Folgenden werden das RGB- und das HSV-Farbmodell vorgestellt.

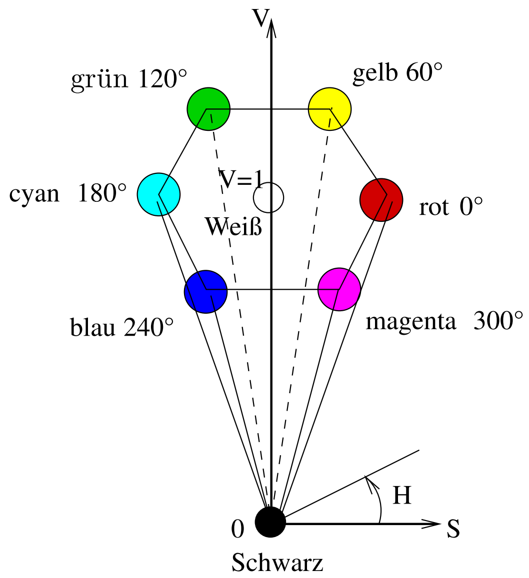
Das RGB-Farbmodell

Da Farbe eine drei-dimensionale Größe ist, handelt es sich bei Farbmodellen um 3D-Räume zur eindeutigen Beschreibung von Farben. Geläufige hardware-orientierte Farbmodelle sind das RGB-Modell (z.B. Farbmonitore), das YIQ-Modell (amerikanisches TV-Farbsystem), oder auch das CMY-Modell für Farbdrucker.

Die von uns verwendeten Kameras liefern ein RGB-Signal. Es handelt sich hier um ein additives Modell, da die Farbwahrnehmung bzw. -beschreibung durch additive Kombination der Primärfarben Rot, Grün und Blau charakterisiert ist. Übliche Darstellung des RGB-Modells ist ein kartesisches Koordinatensystem als Einheitswürfel (Abbildung 3.3(a)).



(a)



(b)

Abbildung 3.3: Das RGB- (a) bzw. HSV-Farbmodell (b)

Der RGB-Farbraum ist linear. Mischungen zwischen Farben ergeben sich durch Mittelung der Primärfarben. Eine ausgezeichnete Achse ist die Hauptdiagonale des Würfels, da hier die Primärfarben zu gleichen Anteilen auftreten findet man entlang dieser die Grauwerte, von $(0,0,0)$ schwarz bis $(1,1,1)$ weiß.

Das HSV-Farbmodell

Das RGB-Farbmodell ist stark hardware-orientiert und spiegelt weniger die menschliche Farbvorstellung (basierend auf den Attributen Farbton, Sättigung und Helligkeit) wieder. Das von Smith [Smith78] vorgeschlagene HSV-Farbmodell (Hue-Saturation-Value) tut dieses, wobei zu betonen ist, daß es sich hier nur um eine Approximation der menschlichen Farbwahrnehmung handelt. In unserem Fall ist weniger die psychologische Plausibilität des Modells von Bedeutung, als vielmehr die Möglichkeit, die Dimensionalität des Farbraumes zu reduzieren, möglichst ohne auf Farbinformation zu verzichten, und, damit verbunden, eine bezüglich der Beleuchtung invariante Farbdarstellung zu erlangen. In den folgenden

Algorithmus 1 Konvertierungsalgorithmen RGB-HSV und HSV-RGB

```
procedure RGB_TO_HSV(r,g,b:real; var h,s,v:real)
{gegeben : r,g,b jeweils in [0,1],
  gesucht : h in [0,360], s und v in [0,1]}
 begin
  max := Maximum(r,g,b);
  min := Minimum(r,g,b);
  v := max;
  if(v<>0) then s := (max - min) / max;
           else s := 0;
  if(s=0) then h := UNDEFINED;
          else begin
           delta := max - min;
           if(r=max) then      h := (g-b) / delta;
           else if(g=max) then h := 2 + (b-r) / delta;
else if(b=max) then h := 4 + (r-g) / delta;
           h := h * 60;
           if(h<0) then h := h + 360;
          end
 end
```

```
procedure HSV_TO_RGB(var r,g,b:real; h,s,v:real)
{gegeben : h in [0,360] or UNDEFINED, s und v in [0,1],
 gesucht : r,g,b jeweils in [0,1]}
 begin
  if(s=0) then (r,g,b) = (v,v,v);
          else begin
           if(h=360) h:=0;
           h := h\ 60;
           i := trunc(h);
           f := h - i;
           p := v * (1-s);
           q := v * (1- (s*f));
           t := v * (1- (s*(1-f));
           case i of
            0: (r,g,b) := (v,t,p);
            1: (r,g,b) := (q,v,p);
            2: (r,g,b) := (p,v,t);
            3: (r,g,b) := (p,q,v);
            4: (r,g,b) := (t,p,v);
            5: (r,g,b) := (v,p,q);
           end
          end
 end
```

Abschnitten werden wir nur auf die H(ue)- und S(aturation)-Komponenten der Farbinformation zurückgreifen und auf den beleuchtungsabhängigen V(alue)-Anteil verzichten.

Abbildung 3.3(b) zeigt den HSV-Raum als eine sechsseitige Pyramide. Die Grundfläche der Pyramide entspricht $V = 1$, also den relativ hellen Farben, während die Spitze $V = 0$ die Farbe Schwarz repräsentiert. Der Farbton H ist durch den Winkel um die vertikale Achse gegeben, wobei rot bei 0 beginnt, grün bei 120 und blau bei 240 liegen. Komplementärfarben liegen sich somit 180 Grad voneinander entfernt gegenüber. Der Wert der Sättigung S reicht von 0 an der V-Achse bis 1 an den Rändern des Sechseck und bedeutet das Verhältnis der Reinheit einer Farbe zu ihrer maximalen Reinheit (bei S=1). Die Grauwerte liegen folglich auf der V-Achse (bei S=0). Für sie ist der Farbwinkel undefiniert. Die voll gesättigten und hellsten Farben liegen am Rand des obersten Sechsecks. Eine Verringerung von S entspricht einer Beimischung von weiß, während eine Reduktion von V, eine Erhöhung des Schwarzanteiles bedeutet.

Zum intuitiven Verständnis der Beziehung zwischen RGB- und HSV-Raum sei gesagt, daß man die sechseckige Basisschicht (V=1) durch Projektion der Oberfläche des RGB-Würfels entlang der Diagonalen (von Schwarz nach Weiß) erhält. Die Hauptdiagonale wird zur V-Achse des HSV-Raumes.

Ein entscheidender Vorteil des HSV-Modells ist die schnelle Umrechnung von RGB- in HSV-Modell. Im Gegensatz zu anderen Modellen (z.B. HSI [Haberäcker95]) kann hierbei auf die Verwendung trigonometrischer Funktionen verzichtet werden, welches die Berechnung relativ beschleunigt.

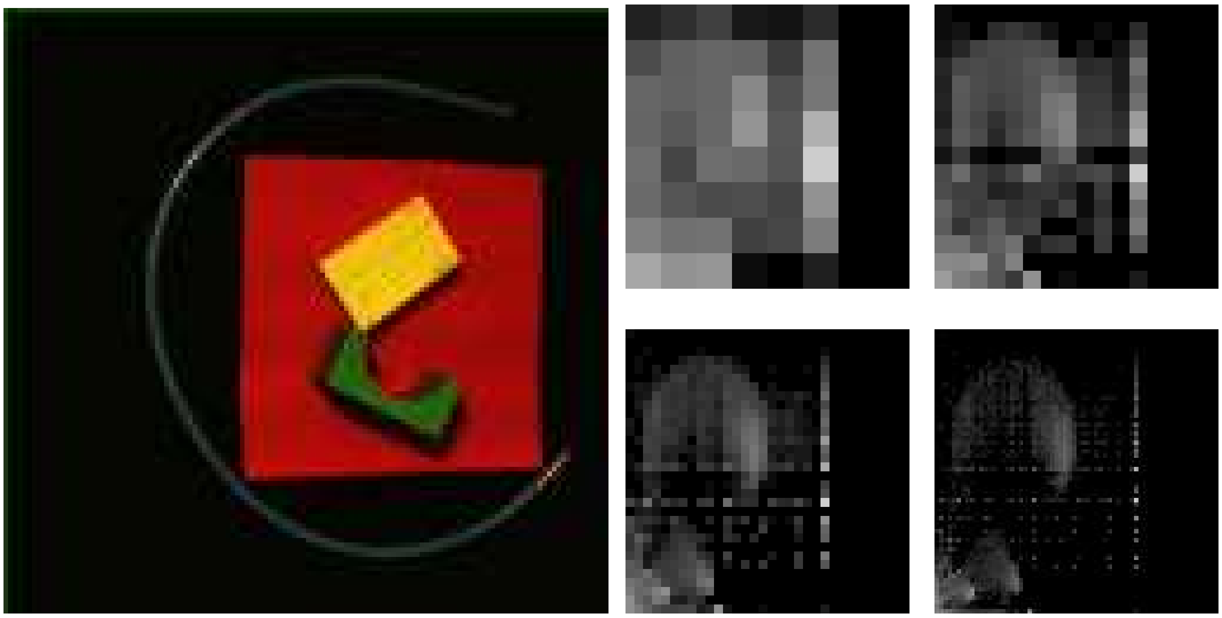
Algorithmus 1 beschreibt die numerische Konvertierung der beiden Farbräume in Pseudocode Notation (nach [Foley90]). In unserem Fall ist die Konvertierung von RGB nach HSV von Bedeutung. Unsere Farbobjekte werden durch die im folgenden Abschnitt beschriebenen HS-Histogramme definiert.

3.2.2 HS-Farbhistogramme

Unter einem Histogram H soll von nun an immer ein zweidimensionales Histogramm eines HSV-Bildes verstanden werden, das nur die beiden Komponenten Farbwert(Hue) und Sättigung(Saturation) betrachtet. Abbildung 3.4 zeigt ein Beispiel für eine typische Szene und dazugehörige HS-Histogramme mit unterschiedlicher Anzahl von Bins. Hier und im weiteren Verlauf der Arbeit gilt der in Abbildung 3.5 dargestellte Aufbau der Histogramme, d.h. die Farbwerte befinden sich entlang der horizontalen Achse, während man die Sättigung in vertikaler Richtung ablesen kann. Die Grauwerte der einzelnen Felder bilden die Anzahl der Einträge ab, wobei zur Darstellung logarithmiert wurde.

3.2.3 Update-Operationen mit dem Fuzzy-ART-Algorithmus

Die einzelnen Elemente der Datenbank sind HS-Histogramme. Aufgabe ist nun die Integration neuer Daten in eine bestehende Datenbank. Erster, naiver Ansatz wäre, die Datenbank als Menge aufzufassen und neue Elemente einfach im Sinne einer Vereinigung hinzu-



(a) RGB-Bild (b) HS-Histogramme mit 8, 16, 32 und 64 Bins

Abbildung 3.4: RGB-Farbbild und zugehörige HS-Histogramme

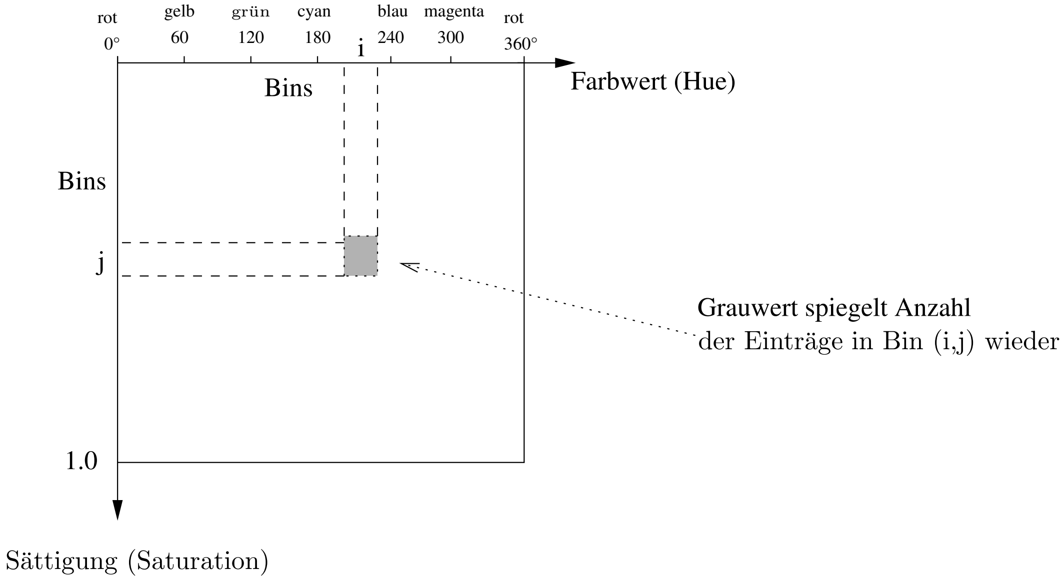


Abbildung 3.5: Aufbau der Darstellung von HS-Histogrammen als Grauwertbild

zufügen. In unserem Falle ist elementares Ziel aber auch die Klassifizierung -im Vergleich zur schlichten Summierung- von Information. Verschiedene Farbinformationen eines Farb-Objektes sollen integriert werden, so daß der resultierende Datenbankeintrag Repräsentant für eine Klasse von Objekten ist. Einen Typ inkrementellen Algorithmus, mit deren Hilfe sich dieses Ziel realisieren läßt, stellt die ART (*Adaptive Resonance Theorie*) nach Carpenter und Grossberg zur Verfügung ([CaGro87],[Moore89],[CaGroRo91]). ART-Klassifizierer sind Vektorklassifizierer. Die Vektoren, die eine Klasse repräsentieren, werden *Prototypen* genannt. Im folgenden Abschnitt wird der Fuzzy-Art-Algorithmus - als der hier verwendete Algorithmus - vorgestellt.

Fuzzy-ART-Algorithmus

Die ART-Architektur löst das *Plastizitäts-Stabilitäts-Dilemma* [Bishop95], d.h. findet einen Ausgleich zwischen der Anpassung des Systems an neue Information (*Plastizität*) und der Speicherung bereits vorhandener Daten (*Stabilität*). Die ART-Algorithmen erfüllen die folgenden *Stabilitätskriterien*:

1. Kein Prototypvektor kann nach Veränderung den vorherigen Wert annehmen.
2. Eine unendlich oft präsentierte Menge von Eingabevektoren erzeugt eine endliche Anzahl von Prototypen.

Seien P, P' Mengen von Prototyp-Vektoren aus $V \subseteq \{0,1\}^n \backslash \{0\}^n$. Der ART1-Algorithmus, als Grundlage für Fuzzy-ART, besitzt folgende Grundstruktur :

1. Starte mit leerer Menge von Prototypen $P = \emptyset$.

2. Sei I Eingabevektor. Setze $P' = P$.

3. Falls $P' \neq \emptyset$, finde dasjenige $J \in P'$, so daß J und I maximal *ähnlich.*

4. Ist $P' = \emptyset$ dann setze $P := P \cup \{I\}$. Gehe zu 2.

5. *Paßt* J nicht zu I, dann setze $P' := P' \setminus \{J\}$. Gehe zu 3.

6. *Paßt* J zu I dann *adaptiere* J durch I. Gehe zu 2.

Automatisch stellt sich die Frage nach der Definition der Begriffe *Ähnlichhkeit, zu-einander-passen, und Adaption*; sie werden im Folgenden in Bezug auf Fuzzy-Art erläutert. Der ART1-Algorithmus ist auf binären Eingabevektoren definiert. Die vorliegenden Eingabe*vektoren* (HS-Histogramme) sind aber reellwertig. Eine reellwertige Erweiterung des ART1-Algorithmus ist Fuzzy-ART. Hier werden Fuzzy-Operatoren in ART1 integriert, und ein (auf das Intervall $[0,1]$) beschränkter, reelwertiger Eingaberaum zugelassen.

Die Verwendung des *Fuzzy-Und-Operators* erfordert eine *Komplementärkodierung* der Vektoren, um die Stabilitätseigenschaften zu wahren. Vorweg die benötigten Definitionen. Seien $p, q \in I\!R^n$. Die Norm $\| \cdot \|$ sei definiert durch

$$\|p\| =_{def} \sum_{i=1}^{n} |p_i| \ . \tag{3.1}$$

Der Fuzzy-Und- bzw. Oder-Operator seien komponentenweise definiert durch

$$(p \wedge q)_i =_{def} \min(p_i, q_i) \tag{3.2}$$

und

$$(p \vee q)_i =_{def} \max(p_i, q_i) \tag{3.3}$$

für alle $i \in \{1, ..., n\}$. Sei E Eingabevektor der Form

$$E = (e_1, ..., e_n) \ \in [0,1]^n, \tag{3.4}$$

dann erhält man den zugehörigen *komplementär-kodierten* Vektor durch

$$I =_{def} (E, E^c) = (e_1, ..., e_n, (1-e_1), ..., (1-e_n)) \ . \tag{3.5}$$

Seien P, P' Mengen von Prototyp-Vektoren aus $V \subseteq [0,1]^n \setminus \{0\}^n$. Ferner sei $0 \leq \rho \leq 1$ der sogenannte *Wachsamkeitsparameter*. Der Fuzzy-Art-Algorithmus:

1. Initial sei $P = \emptyset$.
2. Sei E Eingabevektor.
   - Setze $I = (E, E^c)$.
   - Setze $P' := P$.
3. Falls $P' \neq \emptyset$, finde $J \in P'$, daß
   $$\frac{\|J \wedge I\|}{\|J\|}$$
   maximal ist.
4. Ist $P' = \emptyset$,
   - erzeuge neuen Prototypen $J := \beta I$.
   - Setze $P := P \cup \{J\}$.
   - Gehe zu 2.
5. *Paßt J nicht zu I*, gilt
   $$\frac{\|J \wedge I\|}{\|I\|} < \rho \; ,$$
   - Setze $P' = P' \setminus \{J\}$
   - Gehe zu 3.
6. *Paßt J zu I*,
   - so setze $J := \beta(I \wedge J) + (1 - \beta)J$
   - und $P := P \cup \{J\}$.
   - Gehe zu 2.

Der Parameter β ist die *Lernrate*. Ein Wert $\beta < 1$ bedeutet ein langsames Lernen, während $\beta = 1$ einen Modus des schnellen Lernens darstellt.

Der hier beschriebene Algorithmus läßt sich auch in Form eines Neuronalen Netzes (*fuzzy-art-net*) beschreiben. Die algorithmische Form erscheint übersichtlicher und soll hier genügen; für die neuronale Sicht sei auf die Original-Artikel (s.o.) verwiesen.

Anwendung auf HS-Histogram-Datenbank

Die in dieser Arbeit betrachteten Prototypvektoren sind 2D-Histogramme, also Matrizen. Da die Anordnung als Matrix für den Update-Algorithmus keinerlei Bedeutung hat, läßt sich ein $n \times n$ HS-Histogramm H leicht durch zeilenweises Auslesen als Vektor interpretieren. Mit H_v sei die Vektorrepräsentation von H bezeichnet. Es gilt dann

$$H(i,j) = H_v(i + j \cdot n) \tag{3.6}$$

für alle $i, j \in \{1, ..., n\}$. Die Komplementärkodierung H_v^c erhält man als

$$H_v^c(k) = \begin{cases} H_v(k) \quad , & \text{für} \quad k \leq n^2 \\ 1 - H_v(k) \quad , & \text{für} \quad k > n^2 \end{cases} \tag{3.7}$$

für $k \in \{1, ..., n^2, n^2+1, ..., 2 \cdot n^2\}$. In der praktischen Anwendung ist ein Prototyp Repräsentant für eine Klasse von HS-Histogrammen. Wird ein neuer Vektor I präsentiert und einem Prototypen P zugeordnet, so würde durch den Minimum-Operator bei Wahl von $\beta = 1$ ein Histogrammeintrag auf Null gesetzt, sofern eines der Histogramme an dieser Stelle einen Nulleintrag hat. Folglich wird ein $\beta < 1$ gewählt. In den Tests wurden Werte von $\approx 0.2 - 0.4$ verwendet. Somit ist gesichert, daß der die Klasse repräsentierende Prototyp, bzw. das zugehörige Histogramm, niemals das Auftreten eines dortigen Eintrags "vergißt".

Ein Beispiel

Zur Veranschaulichung der einzelnen Schritte zu Aufbau und Aktualisierung einer Datenbank diene das folgende Beispiel. Abbildung 3.6 zeigt ein blaues und ein rotes Kabel, die nacheinander in die Datenbank aufgenommen werden. Die darauffolgende Hinzunahme eines weiteren roten Kabels führt zur Aktualisierung des "roten" (in der Darstellung linken) Prototypen (Histogramms).

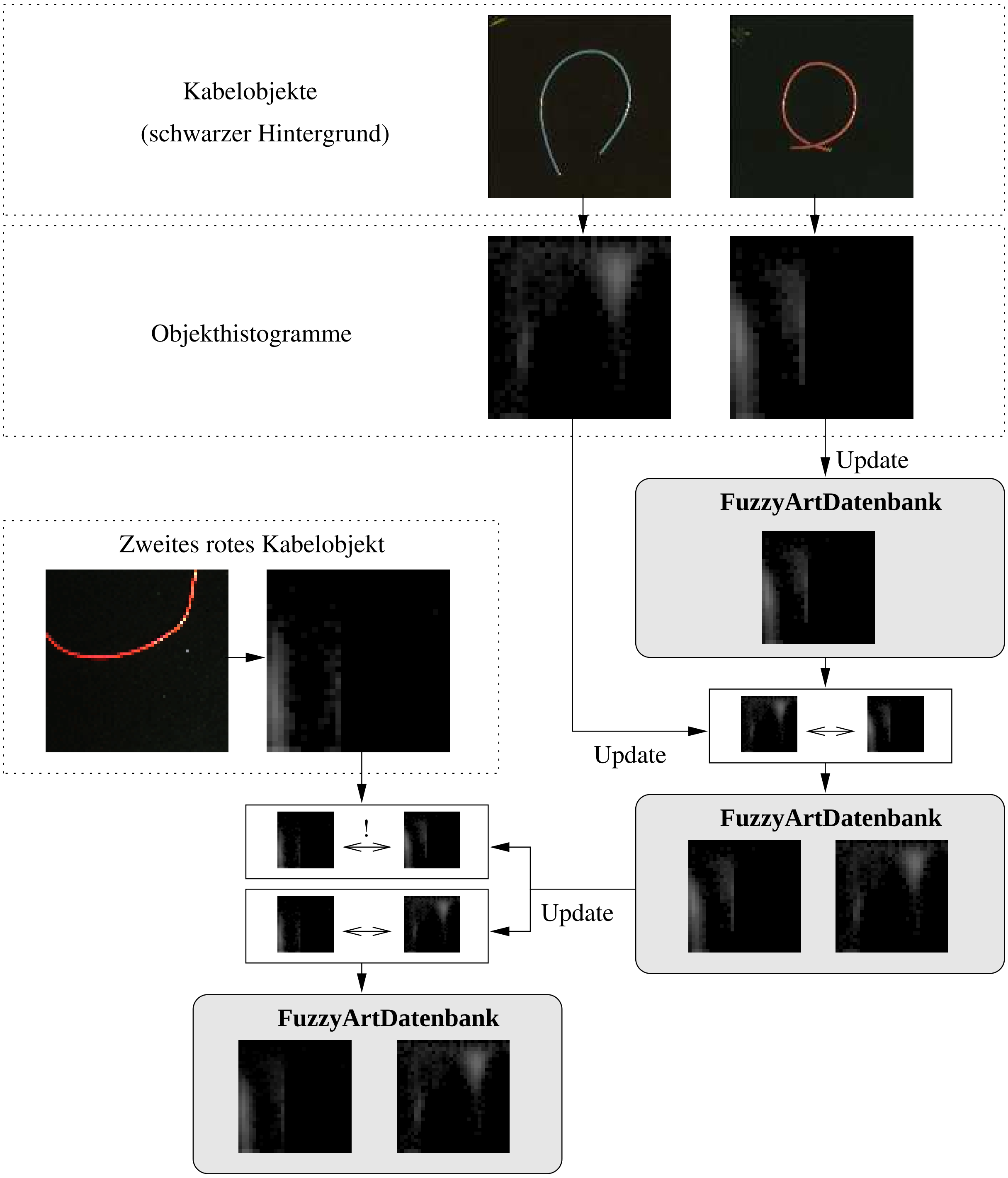


Abbildung 3.6: Ein einfaches Beispiel für Aufbau und Aktualisierung einer Datenbank

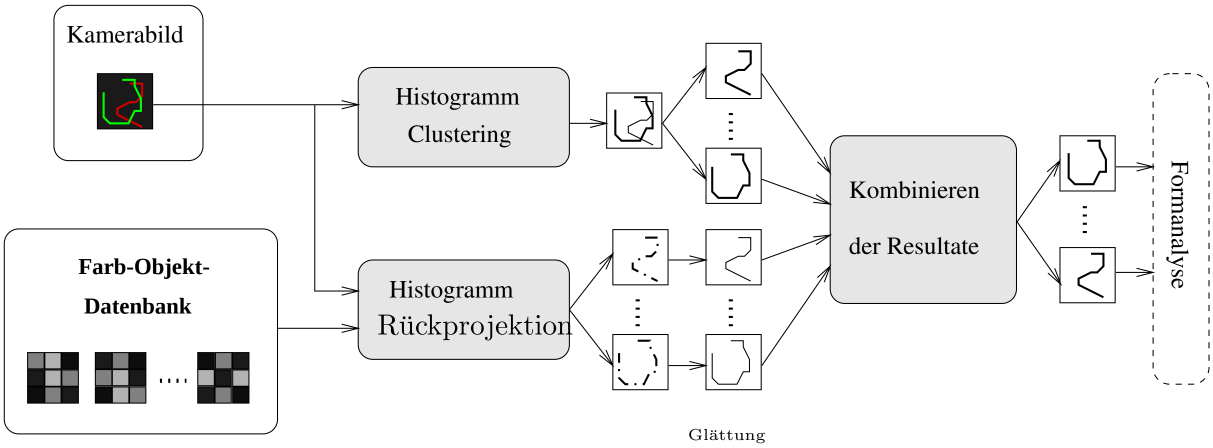


Abbildung 3.7: Objektlokalisation als Bildsegmentierung mittels Farbe

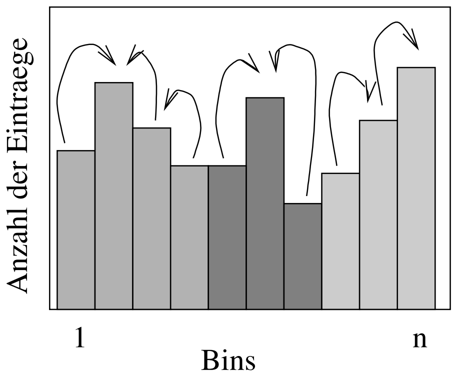
3.3 Objektlokalisation mittels Farbe

Nachdem die Datenstrukturen eingeführt sind, werden jetzt zwei Verfahren zur Lokalisation der Kabel im Bild vorgestellt (Abbildung 3.7). Das Verfahren der graphenbasierten Bildsegmentierung segmentiert beliebige Szenen, während die Rückprojektion der Lokalisation schon bekannter Farb-/Kabelobjekte im Bild dient. Die Ergebnisse beider Ansätze werden geeignet kombiniert (Abschnitt 3.3.3), so daß als Ergebnis eine Menge von möglichen Kandidaten für das Auftreten eines Kabels im Bild steht.

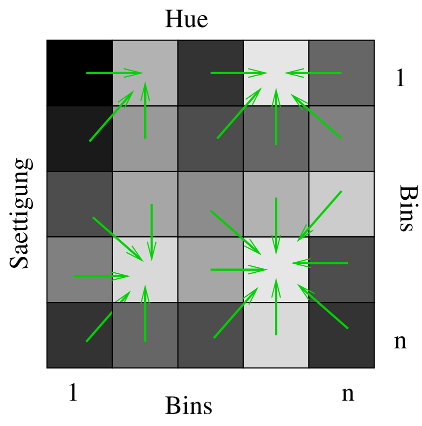
3.3.1 Graphenbasierte Bildsegmentierung

Ziel der Segmentierung ist die Unterteilung eines Bildes in zusammenhängende, disjunkte Teilbereiche. Jede dieser Regionen ist nach einem Einheitlichkeitskriterium gebildet. In der Praxis findet man verschiedene Verfahren zur Farbsegmentierung. Bei den *Bereichswachstumsverfahren* (z.B. [Wahl84]) wird ausgehend von einer Kernregion (häufig ein einzelnes Pixel) versucht, den Bereich zu vergrößern. Dazu müßte in unserem Fall ein Abstandsmaß zwischen zwei Farbwerten existieren, welches im HSV-Raum nicht so einfach vorliegt. Zudem werden Glanzlichter oder ungleichmäßig ausgeleuchtete Regionen nicht als zusammenhängend erkannt; der Unterschied liegt nur in der Helligkeit, nicht im Grundton. Letzteres Argument spricht auch gegen *kantenbasierte Verfahren* oder *Schwellwertansätze* (z.B. [Haberäcker91]).

Im Folgenden wird ein histogrammbasiertes Segmentierungsverfahren auf Farbbildern vorgestellt, basierend auf [Niemann83], [CoFish83] und [Niemann90]. Grundidee dieses Ansatzes ist es, daß Farbwerte nicht direkt miteinander verglichen werden, sondern nur deren Nachbarschaften im Histogramm und die Häufigkeiten.



(a) Grauwert-Histogramm



(b) HS-Histogramm

Abbildung 3.8: Prinzip des Graphenbasierten Clustern auf Histogrammen

Die Clusterbildung findet im Histogramm statt. Die dabei gebildeten Cluster stellen eine Partition des gesamten Histogramms dar. Sei nun ein $n \times n$ HS-Histogramm H gegeben. Mit $H^8(i,j)$ sei die Achter-Nachbarschaft einer Histogrammposition $H(i,j)$ bezeichnet. Wie in Abbildung 3.8 gezeigt wird für jeden Eintrag ein Zeiger auf den Nachbarn mit dem höchsten Eintrag gesetzt. Bei gleich hohen Nachbarwerten wird kein Zeiger gesetzt. Bei Einträgen am Rand werden entsprechend weniger Nachbarn untersucht; ebenso entfällt bei Nicht-Existenz eines höherwertigen Nachbarn der Eintrag. Die Bezeichnung als "graphenbasiertes" Verfahren leitet sich aus der graphischen Darstellung dieses Vorgehens ab.

Mittels dieser Segmentierung des Histogramms wird nun das Bild in zusammenhängende Regionen unterteilt. Einträge ohne einen Zeiger bilden lokale Maxima. Jedem Farbwert wird jetzt der am Ende der sich ergebenen Zeigerkette stehende Farbwert zugeordnet. Bei Segmentierung im HS-Raum kommt eine Besonderheit des HSV-Raumes zum Tragen: die Hue-Komponente ist zyklisch, d.h. der Farbwert H=0 liegt nahe bei H=359. Abbildung 3.9 die Segmentierungsergebnisse zu Abbildung 3.4(a) mit unterschiedlicher Anzahl von Bins des Histogramms.

3.3.2 Rückprojektion

Die Rückprojektion ist ein Lokalisationsalgorithmus nach [SwainBallard91]. Er liefert eine Menge von Koordinatenpaaren , welche die Lage des Objektes in der Szene beschreiben. Grundlage sind Histogramme, in unserem Fall HS-Histogramme. Der Algorithmus geht hierbei in 3 Schritten vor:

1. Erstellen des Verhältnishistogramms von gesuchtem Farbmodell und Bild.

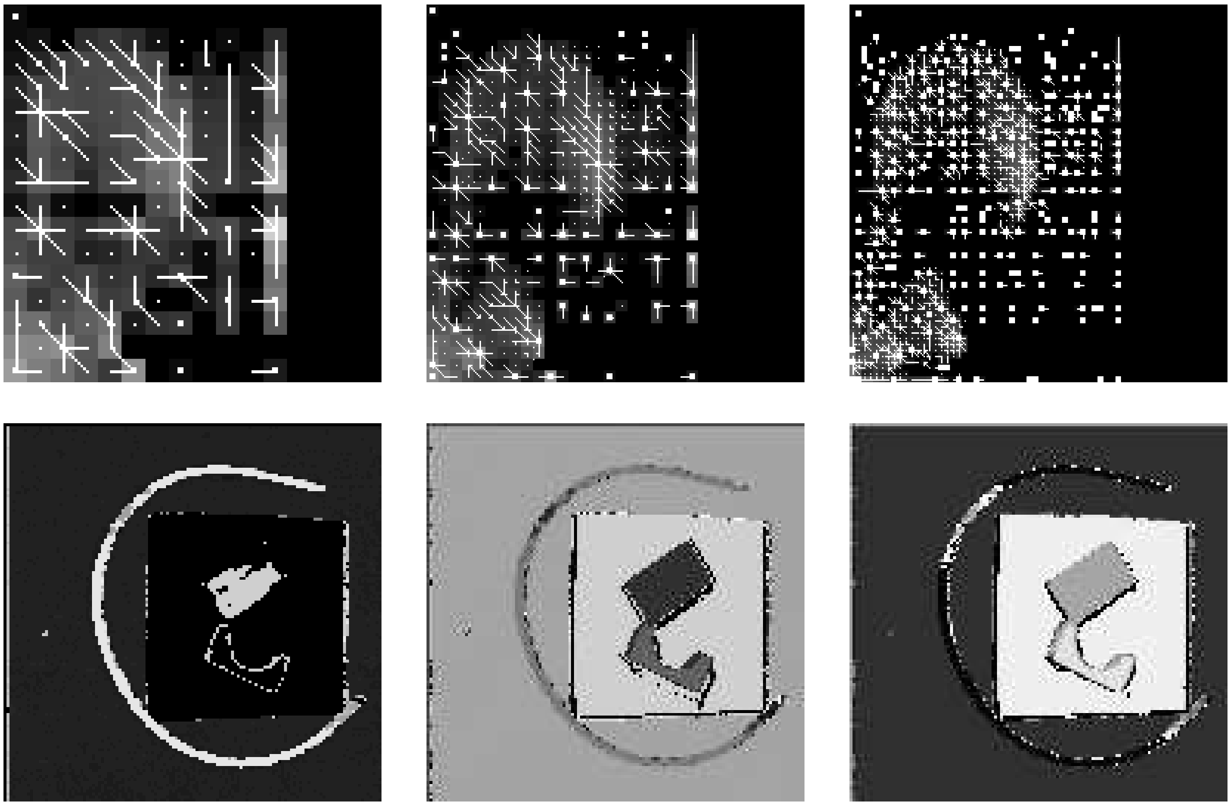


Abbildung 3.9: Segmentierungsergebnisse zu Abbildung 3.4(a) mit 16, 32 und 64 Bins

2. Rückprojektion des Verhältnishistogramms auf das Bild.

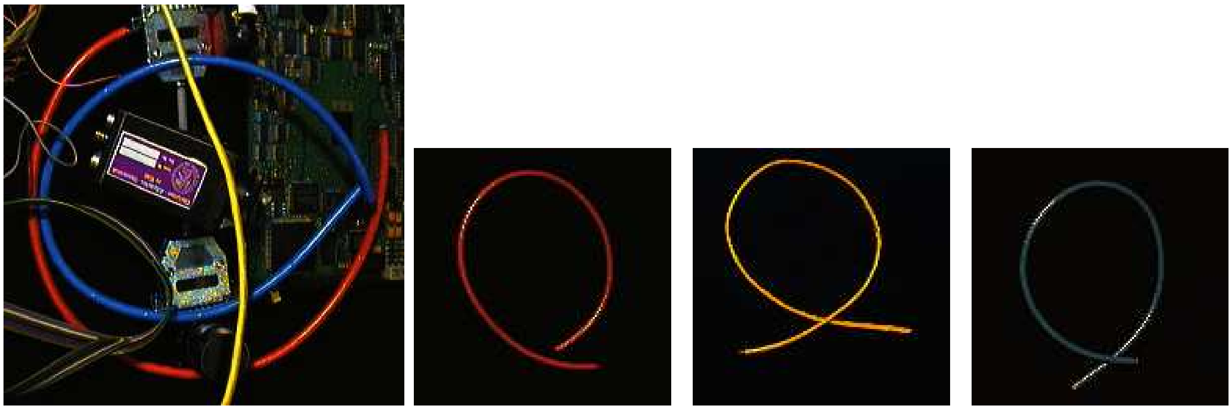
3. Glättung der Rückprojektion.

Aus den HS-Histogrammen P des Objektes und B des Bildes berechnet man das Verhältnishistogramm V als:

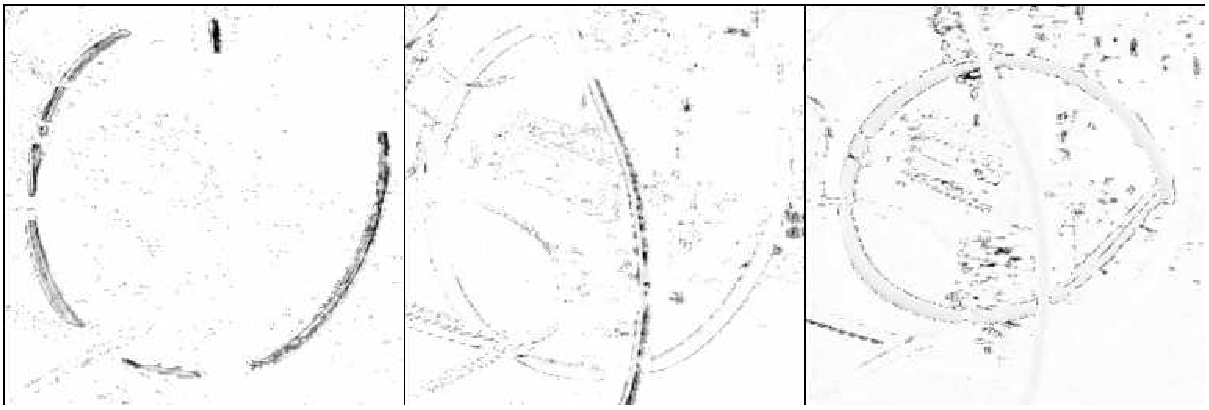
$$V(i,j) = \begin{cases} \min\left\{1, \frac{P(i,j)}{B(i,j)}\right\} & , \quad \text{falls } B(i,j) \neq 0 \\ 0 & , \quad \text{sonst} \end{cases} \qquad i,j \in \{1,...,n\}\ . \tag{3.8}$$

Man gewinnt ein Histogramm, in welchem das Verhältnis der Vorkommen von Farben des Objektes und der Szene stehen. Es ergeben sich Einträge zwischen 0 und 1, wobei ein Wert nahe Null die Farbe als in der Szene nicht charakteristisch für das Objekt bewertet, während Ergebnisse nahe bei 1 eine gute Charakterisierung bedeuten. Der rückprojizierte Wert eines Bildpunktes $B_{(x,y)}$ mit dem HS-Farbwert (i_B, j_B) ist dann $V(i_B, j_B)$.

Das so erhaltene Ergebnis wird noch einmal geglättet, um Fehler durch Störungen im Bildsignal oder numerische Ungenauigkeiten weitesgehend zu eliminieren. Abbildung 3.10 zeigt drei Kabelobjekte und die Ergebnisse der geglätteten Rückprojektion in eine komplexe Szene. Durch abschließende Schwellwertbildung werden die Ergebnisse noch in Binärbilder umgewandelt.



(a) (b)



(c) (d) (e)

Abbildung 3.10: Szene (a), Kabelobjekte (b) und ihre Rückprojektionen in (a) (c)-(f)

3.3.3 Kombination der Ergebnisse beider Ansätze

Es liegen nun zwei Listen von Objektkandidaten in Form von Binärbildern vor. Eine resultiert aus der Graphenbasierten Bildsegmentierung (Liste 1), die andere ist Ergebnis der Rückprojektion der Datenbankeinträge (Liste 2). Beide Listen müssen nun geeignet kombiniert werden, so daß eine Liste an den zweiten Teil der Analyse, das formbasierte Vorgehen, übergeben wird.

Hierzu wird schlicht aus Liste 1 ein Eintrag genommen, mit allen Einträgen aus Liste 2 derart verglichen, daß der Schnitt der markierten Gebiete berechnet wird. Ist dieser größer als ein festgelegter Prozentsatz der Gesamtobjektgröße, so werden die beiden zu einem Eintrag kombiniert, einzeln aus den Listen gestrichen und die Kombination der Liste 2 hinzugefügt. Dieses Vorgehen wird für jedes Element der Liste 1 durchgeführt.

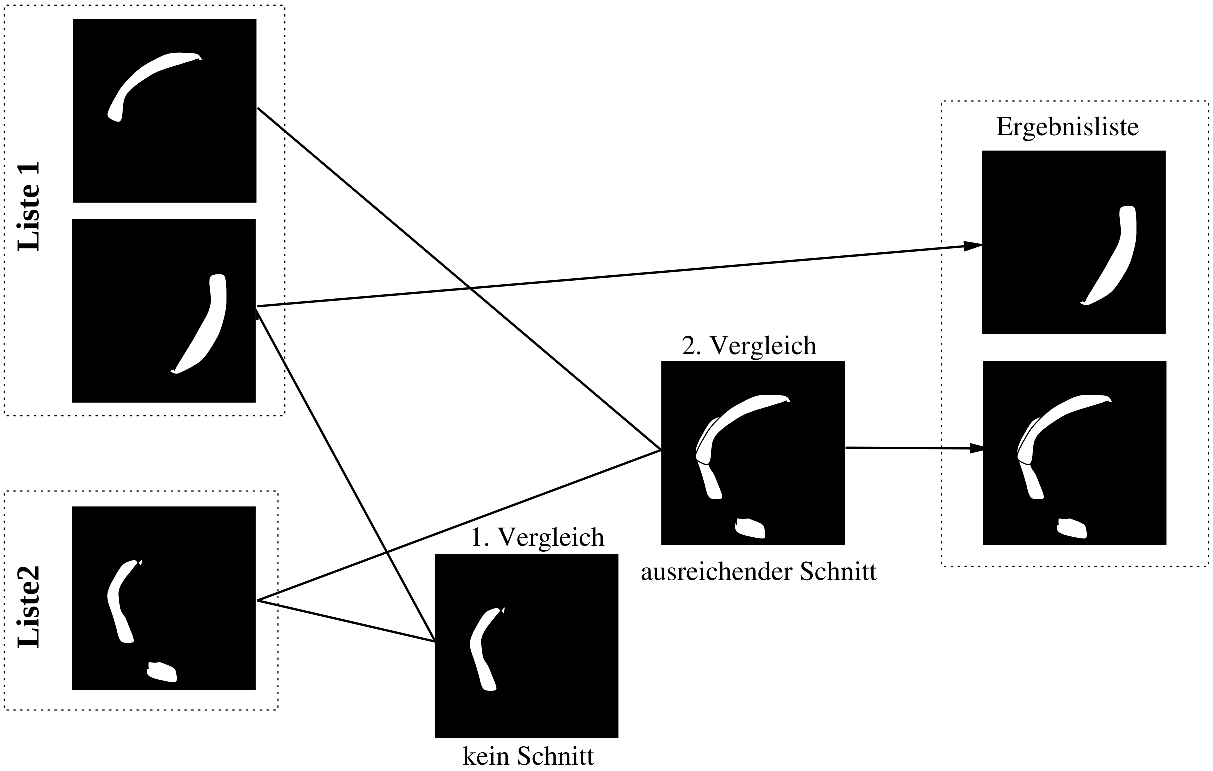


Abbildung 3.11: Ein Beipiel für die Kombination der Kandidatenlisten

Die beiden schlußendlich entstandenen Listen werden noch zusammengefügt. Sie enthalten alle kombinierten Objekte und die Einräge, für die es in der jeweils anderen Liste keine Entsprechung, Kombinationsmöglichkeit gibt. Abbildung 3.11 zeigt ein einfaches Beispiel für das Vorgehen.

3.4 Beispiele (Teil I) : Farbbasierte Objektdefinition und -lokalisation

Die folgenden drei zusammenhängenden Beispiele sollen noch einmal den Weg vom Beispiel zur Lokalisation der Kabel im Bild erläutern. Die hier gewonnen Ergebnisse (Binärbilder der "Kabelkandidaten") werden später auch als Beispiele zur Formanalyse genutzt (Abschnitt 4).

Beipiel 1 : Kabelsalat

Die Szene enthält willkürlich Kabel, Drähte und ähnliche Gegenstände. Die Datenbank ist mit einem roten und einem blauen Kabel vorinitialisiert. Die Segmentierung liefert

Teile des roten und blauen Kabels. Andere Objekte sind schlicht zu klein oder führen durch Verdeckung und Spiegelungen nicht zu geeigneten Segmenten. Die Rückprojektion der Datenbankeinträge liefert im Vergleich zur Segmentierung kein so gutes Ergebnis. Die Kombination beider Ergebnisse allerdings ist -im Sinne der Vollständigkeit der Lokalisation- besser als beide Einzelansätze für sich betrachtet (Abbildung 3.12).

Beipiel 2 : Einfache Szene - Kabel mit Gegenobjekten

In der Szene befinden sich neben dem roten Kabel weitere Objekte, die aber von der Form her kein Kabelobjekt seien können. Die Farbsegmentierung liefert eine Reihe dieser Objekte als Kandidaten. Zusammen mit der Rückprojektion des vorinitialisierten roten Kabels ergibt sich eine relativ gute Lokalisation des Kabels. Die ungeeigneten Kandidaten auszusortieren ist Aufgabe der formbasierten Analyse (Abbildung 3.13).

Beipiel 3 : Kabel im Monitor

Die Monitorszene entält zwei rot Kabel derselben Art. Die Datenbank ist ebenso mit einem "roten" Beispiel initialisiert. Zusätzlich zu dem roten Kabel wird hier durch die Segmentierung auch noch ein blaues Segment markiert, das tatsächlich auch in der Form einem Kabelobjekt entspricht (Abbildung 3.14).

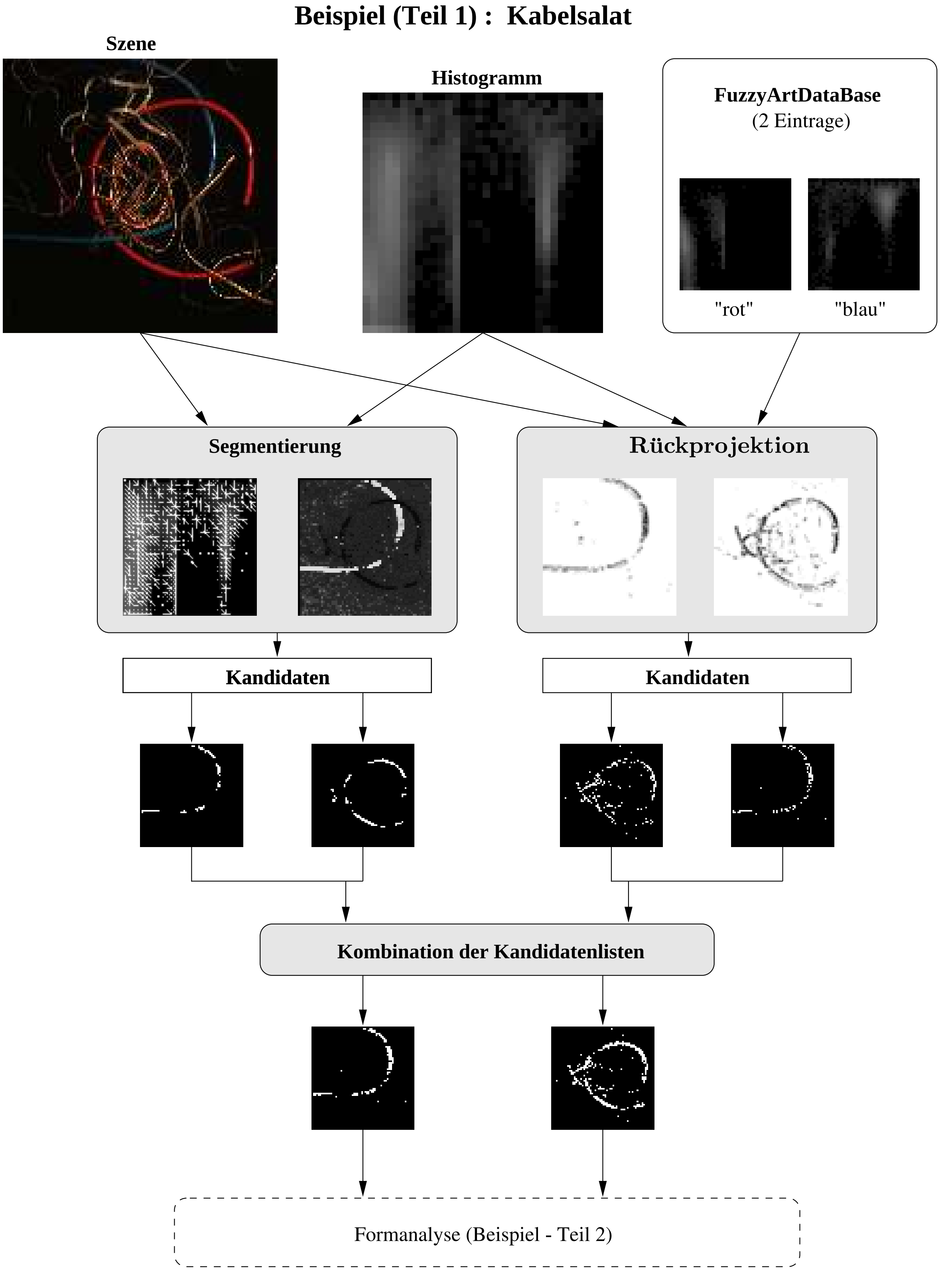


Abbildung 3.12: Erster Teil des Beispiels 1 "Kabelsalat" - Farbbasierte Lokalisation

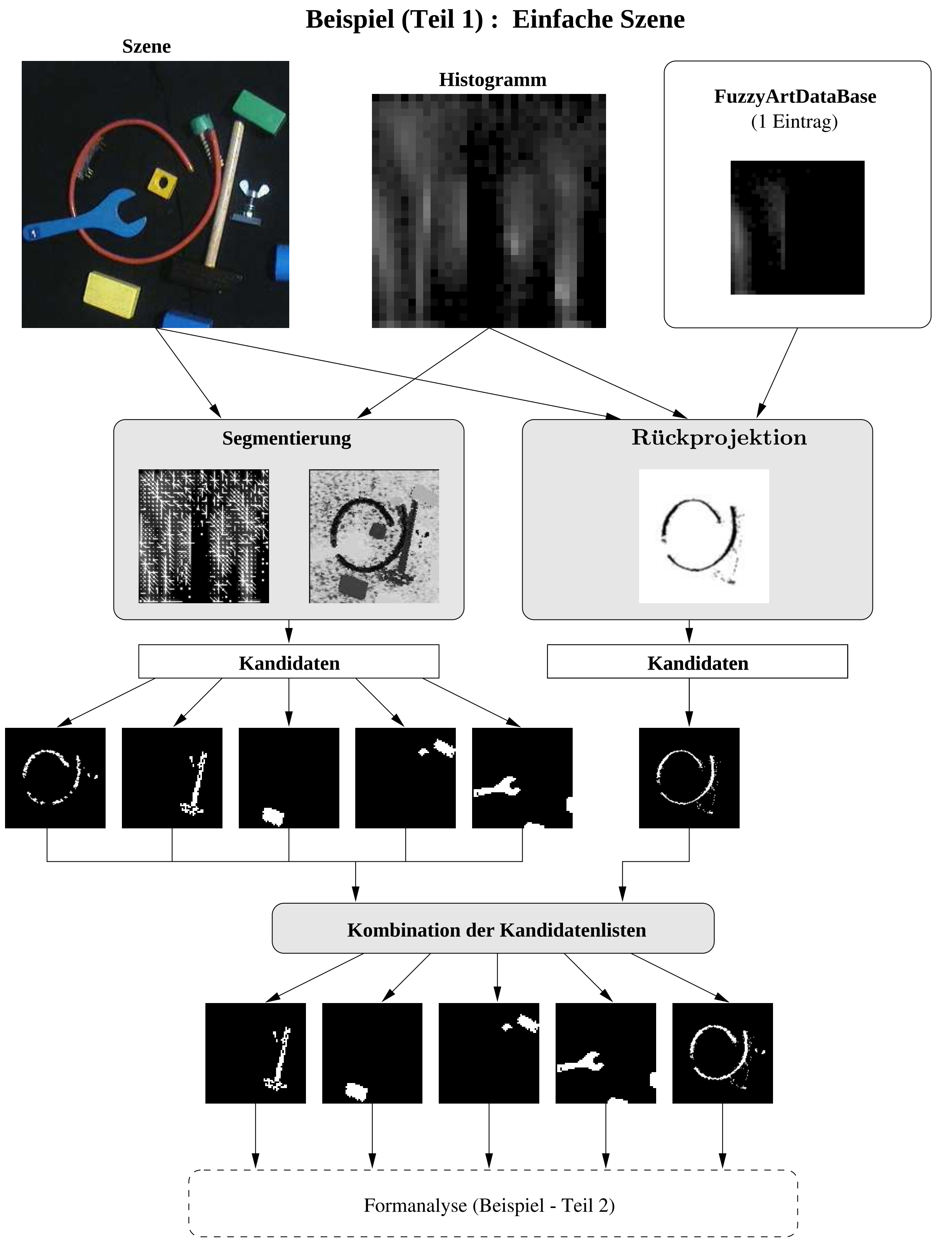


Abbildung 3.13: Erster Teil des Beispiels 2 ”Einfache Szene” - Farbbasierte Lokalisation

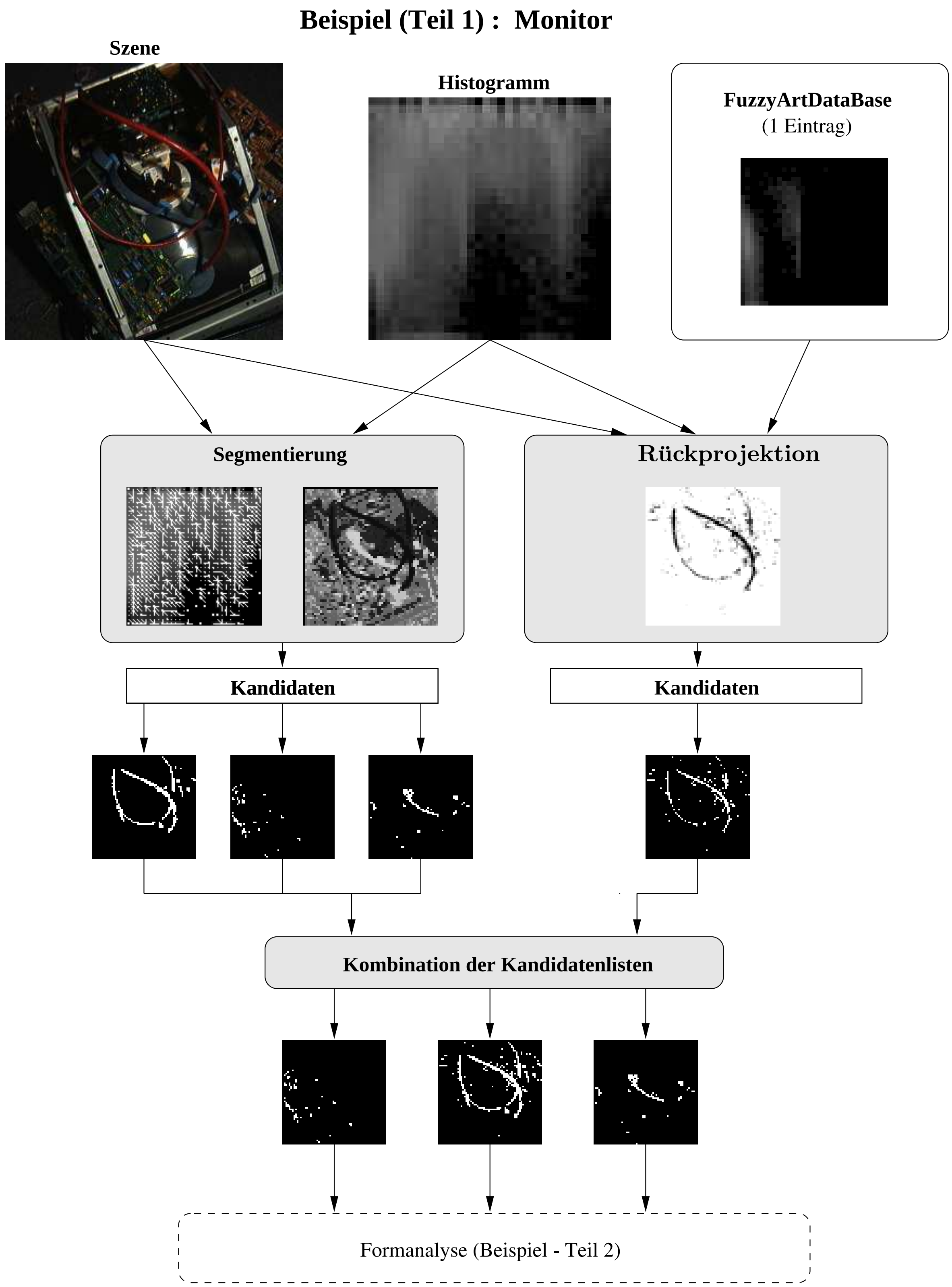


Abbildung 3.14: Erster Teil des Beispiels 3 "Monitor" - Farbbasierte Lokalisation

Kapitel 4

Formbasierte Objekterkennung

Im vorangegangenen Abschnitt 3.3 wurden mittels des Merkmales Farbe Bildregionen als potentielle Objekte (*Kandidaten*) segmentiert. In einem nächsten Schritt müssen diese Segmente nun geprüft werden, ob sie in Bezug auf Form der Objektdefinition Kabel entsprechen. Ist die Farbanalyse durch ihr pixelorientiertes Vorgehen noch auf der Stufe der Signalbearbeitung (*Signal Level* [Sarkar93]) angesiedelt, so erreicht die Analyse mit der Betrachtung der Segmente nun eine *strukturelle Stufe* (*Structural Level*).

Abbildung 4.1 zeigt das Vorgehen zur Formanalyse graphisch. In einem ersten Schritt werden - unter Verwendung eines einfachen *Hit-Miss-Operators* und der *Bounding-Box* - die von der Form unpassenden Segmente (speziell rundliche) eliminiert. Der Verlauf der verbleibenden Formen wird mit einem Kohonen-Netz approximiert, dabei entstehende, offensichtlich (im Sinne eines kontinuierlichen Verlaufs) "unsinnige" Verläufe mit einfachen heuristischen Verfahren korrigiert, und, in Erwartung einer *glatten* Form, mit einem Kalman Filter geglättet.

Nachdem der Verlauf des vermeintlichen Kabels (in Teilen) approximiert ist, stellt sich nun die Frage, ob oder inwieweit sich die Teilverläufe zu einem Objekt zusammenfügen lassen, auf welches das Prädikat Kabel zutrifft. Hierzu wird ein mathematisch gut faßbares, aber natürliches Modell für den Verlauf bzw. die Form von flexiblen Objekten verwendet: *Splines*. Das zu lösende Problem ist ein Gruppierungsproblem. Es werden hierbei zwei Gestaltprinzipien (*Gestalt laws of grouping*) herangezogen: die *Nähe* (*Proximity*) und die *kontinuierliche Fortsetzung* (*Good Continuation*) [Sarkar93].

Ergebnis der Objekterkennung ist eine Menge von Binärbildern, die im darauf folgenden Kapitel 5 Grundlage für die 3D-Rekonstruktion der Kabel sind.

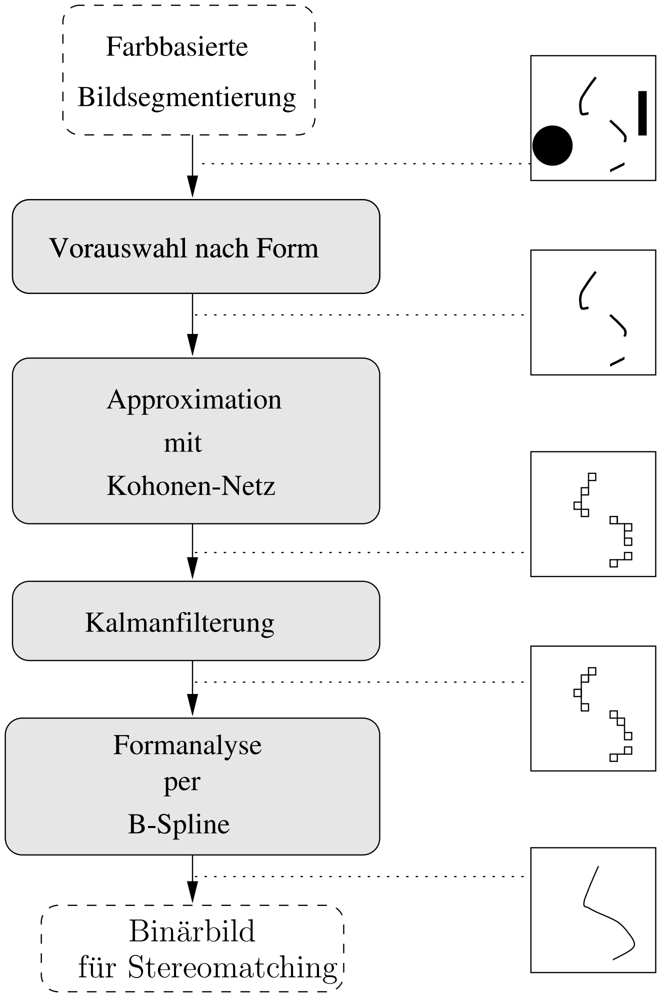


Abbildung 4.1: Vorgehen bei der Formanalyse der Objektkandidaten

4.1 Einfache Vorauswahl nach Form

In Abbildung 4.2 ist eine Reihe von typischen Flächensegmenten dargestellt, wie sie im Anwendungsfall auftreten. Idealerweise ist ein Kriterium gesucht, welches relativ gezielt nur die länglichen, schmalen und gebogenen Segmente auswählt. Da dieses aber ein kaum ("einfach") faßbares Prädikat ist, beschränkt sich diese Arbeit darauf, einen Teil der "Komplementärmenge der gesuchten Form" auszuschließen, z.B. quadratische und rundliche Objekte. Insbesondere kleine Segmente, deren Form keinerlei Information mehr birgt, werden dabei aussortiert.

Abbildung 4.2: Typische Flächensegmente, wie sie als Ergebnis der Segmentierung auftreten

4.1.1 Die orientierte Bounding-Box

Es wird das kleinste die Form umgebende Rechteck (*bounding box*) der Seitenlängen $l_k \times l_l$ genommen und dessen Seitenverhältnis geprüft. Als erstes, einfaches Auswahlkriterium könnte man folgendes wählen : Ist für o.B.d.A $l_k \leq l_l$ und einen Grenzwert $r \in [0, 1]$:

$$\frac{l_k}{l_l} > r \ , \tag{4.1}$$

so gilt das Segment als "zu rund" und wird aussortiert. Problem hierbei ist, daß die Ausrichtung (Orientierung der ersten Hauptkomponente) der Segmente nicht achsenparallel ist, und dadurch Objekte gleich dimensionierte Rechtecke ergeben, die offenbar eine vollständig andere Form besitzen. Ein besseres Ergebnis erzielt man, wenn man die Rechtecke parallel zu der ersten (und zweiten) Hauptkomponente setzt, d.h. man berechnet die Extrema der Punktemenge bzgl. dieser. Abbildung 4.3(a) zeigt hierzu ein Beispiel.

Eine Ausnahme (bzgl. Gleichung (4.1)) bilden großflächige Segmente, wie in Abbildung 4.3(b) dargestellt. Hier ist eine Entscheidung über das Seitenverhältnis offenbar unzureichend, da sowohl das Kabelsegment, als auch der Kreis akzeptiert oder abgelehnt würden, je nach Wahl von r. Um dieses Problem zu umgehen werden die Segmente vorher mittels des Hit-Miss-Operators *Thinning* ausgedünnt, skeletiert. Die gewünschte Wirkungsweise (das "Verschwinden" der flächigen Objekte) ist in Abbildung 4.3(c) zu sehen.

4.1.2 Der Hit-Miss-Operator *Thinning*

Die Idee des Hit-Miss-Operators ist es, Objekte spezieller Form zu detektieren. Für die Detektion einer bestimmten Form in Binärbildern ist die Kombination zweier morphologischer Operatoren notwendig. Ein Operator, definiert durch seine Maske M_1, detektiert die Umgebung (*miss*), ein weiterer M_2 das Objekt (*hit*). Allgemein wird der Hit-Miss-Operator

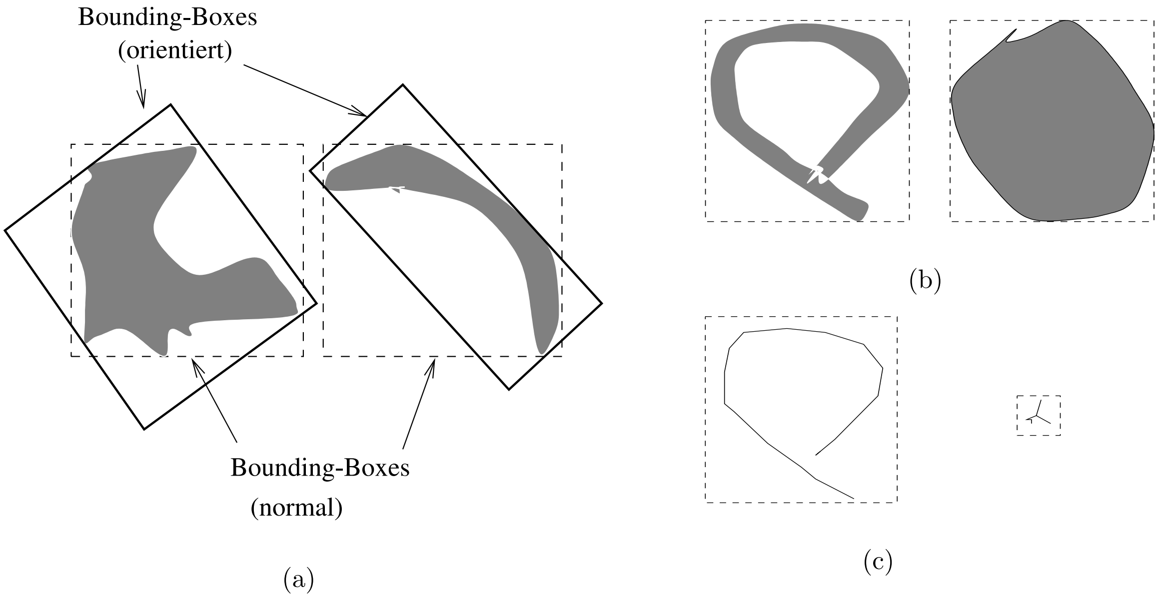


Abbildung 4.3: Die orientierte Bounding-Box (a) und ein Beispiel zum Ausdünnen großflächiger Objekte (b) und (c)

fogendermaßen definiert:

$$G \otimes (M_1, M_2) = (G \ominus M_1) \cap (\bar{G} \ominus M_2) \tag{4.2}$$
$$= (G \ominus M_1) \cap \overline{(G \oplus M_2)} \tag{4.3}$$

mit $M_1 \cap M_2 = \emptyset$ (notwendig damit $G \otimes (M_1, M_2) \neq \emptyset$).

Eine Anwendung dieses Prinzips ist der Thinning-Operator. Er erodiert ein Objekt, zerlegt es aber nicht in mehrere Teile, d.h. er ist topologieerhaltend. Unter Verwendung einer kombinierten Schreibweise (möglich, da Masken disjunkt) erhält man folgende 8 Masken (jeweils 45 rotiert), mittels derer ein Objekt symmetrisch ausgedünnt werden kann:

$$M_{t1} = \begin{bmatrix} 0 & 0 & x \\ 0 & 1 & 1 \\ x & 1 & 1 \end{bmatrix}, M_{t2} = \begin{bmatrix} 0 & 0 & 0 \\ x & 1 & x \\ 1 & 1 & 1 \end{bmatrix}, M_{t3} = \begin{bmatrix} x & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & x \end{bmatrix}, \tag{4.4}$$

$$M_{t4} = \begin{bmatrix} 0 & x & 1 \\ 0 & 1 & 1 \\ 0 & x & 1 \end{bmatrix}, \qquad M_{t5} = \begin{bmatrix} 1 & x & 0 \\ 1 & 1 & 0 \\ 1 & x & 0 \end{bmatrix}, \tag{4.5}$$

$$M_{t6} = \begin{bmatrix} x & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & x \end{bmatrix}, M_{t7} = \begin{bmatrix} 1 & 1 & 1 \\ x & 1 & x \\ 0 & 0 & 0 \end{bmatrix}, M_{t8} = \begin{bmatrix} 1 & 1 & x \\ 1 & 1 & 0 \\ x & 0 & 0 \end{bmatrix}. \tag{4.6}$$

Hierbei bedeutet 1, eine Eins in der Hit-Maske, 0 Eine eins in der Miss-Maske und x, Null in beiden Masken. Der symmetrische Thinning-Operator läßt sich somit zusammenfassend

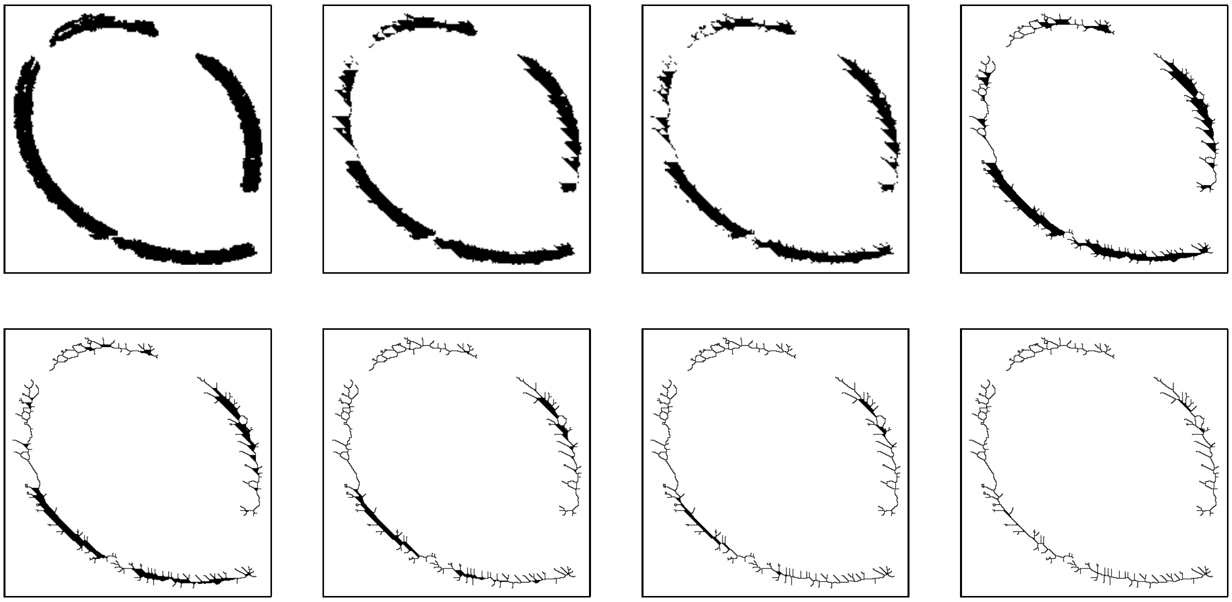


Abbildung 4.4: Anwendungsbeispiel Thinning-Operator (nach 0,1,3,5,..,13 Schritten)

als

$$G \oslash M_T = G / \bigcup_{n=1}^{8} G \oplus M_{Tn} \tag{4.7}$$

schreiben, wobei $\oslash$ die Thinning-Operation kennzeichnet. Der symmetrische Thinning-Operator ist eine iterative Prozedur, die solange wiederholt wird, bis keine Veränderungen mehr auftreten. Abbildung 4.4 zeigt die Ergebnisse der Operation, angewandt auf ein für den Rahmen dieser Arbeit typisches Flächensegment.

Zusammenfassend noch einmal das Vorgehen zur Vorauswahl für vorgegebene minimale/maximale Seitenlängen l_{min}/l_{max} und ein maximales Seitenverhältnis r :

1. Die Flächensegmente werden mit dem Thinning-Operator ausgedünnt.
2. Die orientierte Bounding-Box der Seitenlängen $l_l \times l_r$ wird berechnet.
3. Ist $l_l, l_r > l_{max}$ (Segmente vom Typ Abbildung 4.3(c)(links)),dann Segment behalten,
4. andernfalls , wenn zusätzlich $\frac{l_r}{l_l} > r$ oder $l_l, l_r < l_{min}$, aussortieren des Segments.

Die verbleibenden Segmente werden nun mit den Kohonen-Netzen approximiert.

4.2 Verlaufsapproximation durch Kohonen-Netze

Bildsegmentierung und Vorauswahl nach Form liefern eine Menge von Flächensegmenten, welche potentiell ein gesuchtes Objekt darstellen. Gesucht ist eine zusammenhängende Darstellung des Kabels. Problem ist also, die Lücken zu überbrücken, die gestreute Verteilung in irgendeiner Form in einen zusammenhängenden Kabelverlauf zu überführen. Die Idee hierzu ist, eine Folge von Positionen entlang des Kabels zu haben, somit eine topologisch geordnete Menge von Punkten (Stützstellen), die im weiteren Verlauf auch Basis für die Approximation und Erkennung durch Splines sein werden.

Zur Anwendung gelangen hier die selbstorganisierten Karten nach Kohonen (*Self organizing Maps (SOMs)*). Es handelt sich hierbei um einen selbstorganisierenden, Topologie abbildendenden, nicht-überwacht wettbewerbslernenden Typ Netz ([Kohonen82],[Kohonen84]). Im folgenden Abschnitt wird eine formelle Definition der SOMs gegeben, wie sie im Rahmen dieser Arbeit verwendet werden.

4.2.1 Formales Modell

Ein SOM definiert eine Abbildung von $I\!R^n$ auf ein ein- bzw. zwei-dimensionales Feld von m Knoten $i = 1, .., m$. Mit jedem Knoten i ist ein Referenzvektor $m_i = [\nu_{i1}, ..., \nu_{in}]^T$ assoziiert. Im klassischen Fall wird für einen Eingabevektor $x = [x_1, ..., x_n]$ der kleinste Euklidische Abstand $||x - m_i||$ zu jedem Knoten (bzw. seinem Referenzvektor) des Netzes ermittelt. Knoten c gegeben durch

$$c = argmin \{||x - m_i||\} \tag{4.8}$$

bezeichnet man als *best-matching node*. Je nach Kontext ist dann der Knoten (als Repräsentant einer Klasse) oder der zugehörige Referenzvektor Antwort des Netzes. In unserem Fall interessiert mehr die Position aller Knoten im Eingaberaum, also die Referenzvektoren und die Topologie der Knoten, ihre Lage zueinander. Es soll die Lage eines Kabels im Raum gelernt werden.

Die folgende Gleichung (4.9) beschreibt die Updatefunktion:

$$m_i(t + 1) = m_i(t) + h_{ci}(t) [x(t) - m_i(t)] , t = 0, 1, 2... . \tag{4.9}$$

Ein Update bezieht sich nicht nur auf den Knoten m_c (best match) sondern auch auf eine Nachbarschaft desselben. h_{ci} ist die*Nachbarschaftsfunktion*. Für die Konvergenz ist notwendig, daß

$$h_{ci}(t) \to 0 \qquad \text{wenn} \qquad t \to \infty . \tag{4.10}$$

In Abbildung 4.5 ist die in der Literatur gebräuchliche *r-Nachbarschaft* graphisch dargestellt. Zur Bestimmung der Nachbarschaft sei die Topologie der Knoten durch eine An-

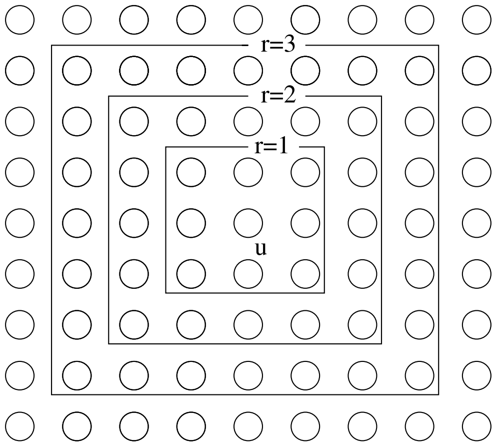


Abbildung 4.5: Die r-Nachbarschaften des Neurons u

ordnung als $n-$dimensionaler Hyperquader gegeben (siehe zweidimensionalen Fall in Abbildung 4.5), d.h. jedem Knoten ist in Form eines n-dimensionalen, ganzzahligen Vektors $p^{(u)}$ eine Position zugewiesen, welche die Position innerhalb des Gitters $\boldsymbol{Z}^n$ angibt. Die r-Nachbarschaft N_r des Knotens c ist durch einen Radius $r \geq 0$ gekennzeichnet; sie enthält die Knoten, deren Abstand zu c kleiner als r ist, also:

$$N_r(c) = \{u|dist(u,c) \leq r\} \ . \tag{4.11}$$

Hierbei ist für zwei beliebige Knoten u und v $dist(u,v)$ die Supremumsnorm, d.h. der Abstand entspricht dem Betrag der maximalen Differenz der einzelnen Koordinaten der Knoten :

$$dist(u,v) = \max_{i \in \{1,\ldots,n\}} \{|p_i^{(u)} - p_i^{(v)}|\} \ . \tag{4.12}$$

Eine Alternative bilden hier die gaussartigen Funktionen der Form

$$h_{ci} = \alpha(t) \cdot \exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right) \ , \tag{4.13}$$

welche den Grad der Nachbarschaft nicht binär, sondern kontinuierlich erfassen. Nachteil dieses Ansatzes ist die Verwendung der Exponentialfunktion, was unweigerlich zu einem Anstieg des Rechenaufwandes führt. Die hier gewählte Zwischenlösung ist eine Erweiterung der binären r-Nachbarschaft auf reelwertige Abstände. Es wird ein über die Zeit t geringer werdender Maximalabstand von $r_{max}(t)$ Knoten (Abstand im Sinne der 1-dimensionalen r-Nachbarschaft) vorgegeben. Mit

$$h(v) = \frac{1}{|index(v) - index(u)| + 1} \tag{4.14}$$

sei die im folgenden genutzte Nachbarschaftsfunktion bzgl. der Knoten u und v definiert, wobei v aus der $r_{max}(t)$-Nachbarschaft ist und $index(.)$ die Position in dem 2D-Netz darstellt.

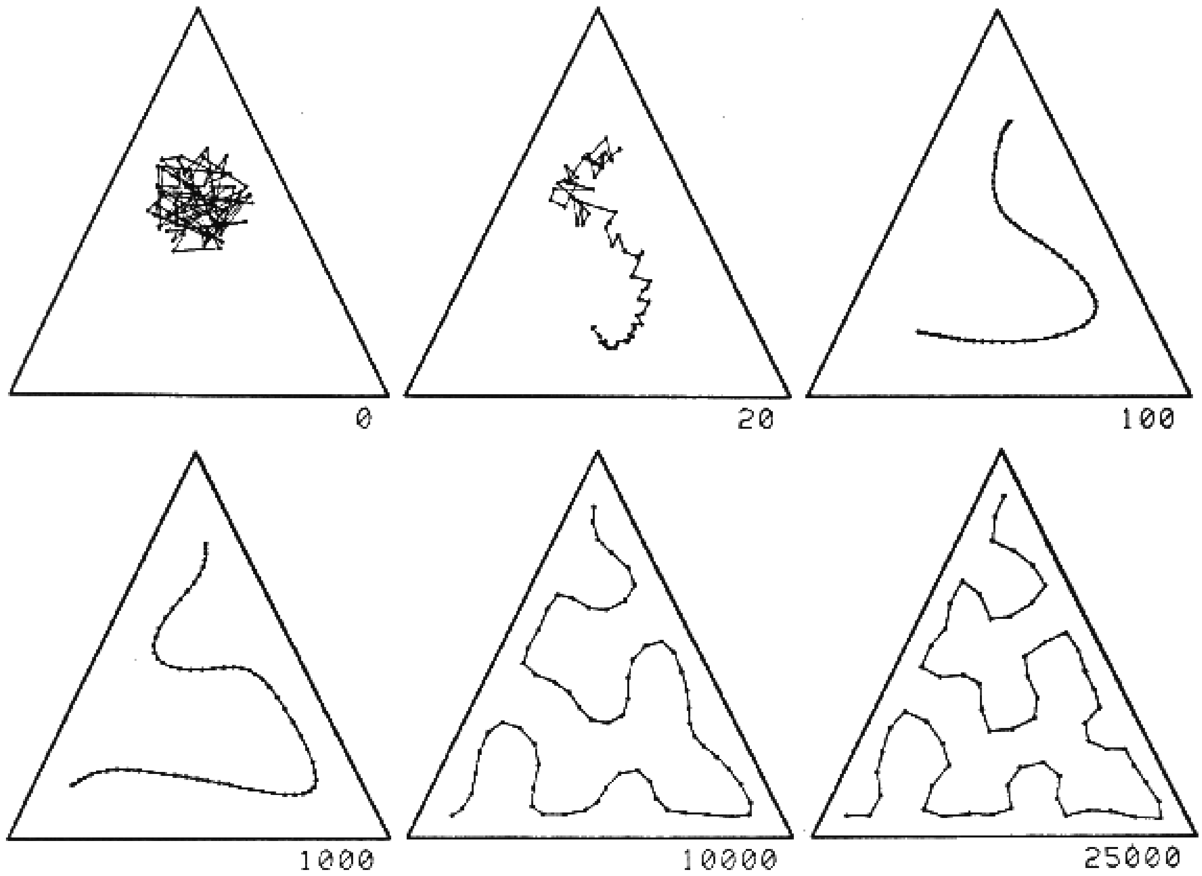


Abbildung 4.6: Klassische Anwendung eines linearen SOM's (aus [Kohonen82])

Relevant für diese Arbeit sind die Kohonen-Netze mit jeweils nur zwei Nachbarn, mit Ausnahme des ersten und letzten Knotens. Es gilt also $n = 2$. Ein Kohonen Netz ist eine geordnete Folge von m zweidimensionalen Vektoren ν_i. Diese Ketten sollen den eindimensionalen Verlauf der Kabel widerspiegeln, indem sie die einzelnen Segmente approximieren. Abbildung 4.6 zeigt ein klassisches Beispiel für die Approximationsleistung 2-dimensionaler Kohonen-Netze [Kohonen82].

4.2.2 Approximation der Segmente

Gegeben sei ein Binärbild B der Flächensegmente und die Menge der Flächensegmenten $M = \{s_1, ..., s_n\}$ mit

$$s_i = \{(x_1, y_1)_i, ..., (x_n, y_n)_i\} .$$

Jedes Segment wird durch ein 2-dimensionales Kohonen-Netz K_{s_i} mit m_{s_i} Knoten approximiert. Die Wahl der Anzahl der Knoten geschieht abhängig von der Größe der $l_l \times l_k$

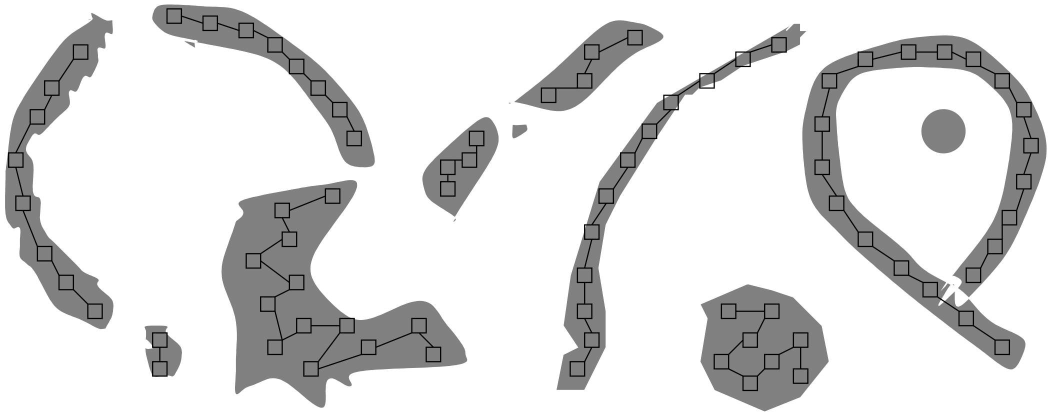


Abbildung 4.7: Approximation von Flächensegmenten durch Kohonen-Netze

Bounding-Box des Segmentes s_i als

$$m_{s_i} = \max\{l_l, l_k\}/d_{mid} \;, \tag{4.15}$$

wobei $d_{mid} = 5, ..., 10$ gebräuchliche Werte sind.

Die Knoten des Netzes werden zufällig mit Positionen innerhalb der Box initialisiert. Danach werden dem Netz K_{s_i}sukzessiv die Elemente aus s_i präsentiert und die Knoten gemäß Gleichung (4.14) adaptiert. Beispiele für die Approximation zeigt Abbildung 4.7.

4.2.3 Nachbearbeitung fehlerhafter und fehlender Verbindungen

Ziel der Nachbearbeitung ist es, Verbindungen zu trennen, die keinem typischen Kabelverlauf entsprechen, und triviale Verbindungen zwischen Netzen herzustellen.

Fehlerhafte Verbindungen

Abbildung 4.8 zeigt Beispiele für die Auswirkung "ungünstiger" Segmentierungsergebnisse auf die Approximation von Kabelverläufen. Man sieht zwei Typen von Segmentanordnungen, die zu fehlerhaften Ergebnissen führen:

- Kreuzungssituationen
- Segmentierungsfehler durch Bildverschmierungen

Kreuzungssituationen sind in diesem Zusammenhang die häufigste Fehlerquelle; sich kreuzende Kabel gleicher Farbe oder - noch wahrscheinlicher - ein Kabel, das sich im Bild mit sich selber kreuzt, werden als ein Segment erfaßt, und die topologische Anordnung durch

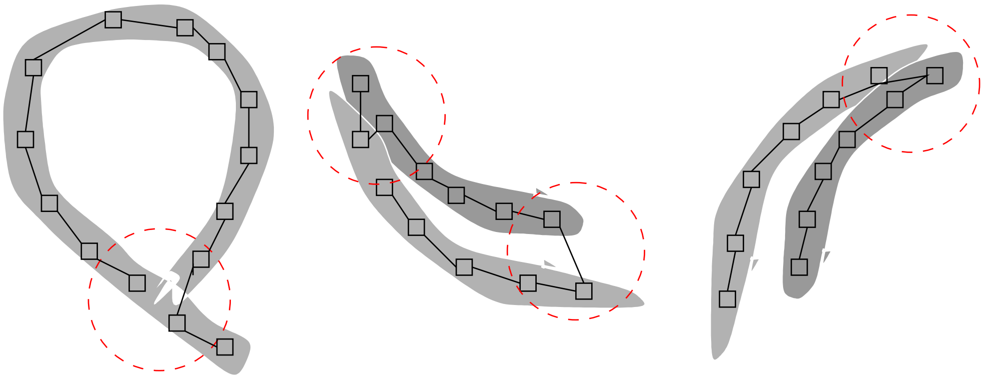


Abbildung 4.8: Beispiel für fehlerhafte Approximation der Topologie

das Kohonen-Netz falsch abgebildet. *Segmentierungsfehler aufgrund schlechter Bilddaten* sind ein Problem der Praxis. Da nicht in jeder Situation Objekt und Kamera idealen Abstand zueinander haben, treten Bildunschärfen auf. Ein weiteres Problem sind Störungen, hervorgerufen durch den kompletten Versuchsaufbau mit Roboterarm, Kamerasystem und Workstation. Gesucht ist ein *einfaches* Kriterium, zu entscheiden, ob in einem Kohonen-Netz der Schritt vom m-ten zum $m+1$-ten Knoten noch als "glatt" zu akzeptieren ist. Sei K_n ein 2-dimensionales Kohonen-Netz mit n Knoten $(k_1, ..., k_n)$. Man betrachtet nun den Winkel zwischen zwei aufeinanderfolgenden Streckensegmenten $s_{i-1} = [k_{i-1}, k_i]$ und $s_i = [k_i, k_{i+1}]$ für $i \in 2, ..., n-1$. Überschreitet dieser einen Grenzwert α, ist also

$$\angle(s_{i-1}, s_i) > \alpha \ , \qquad (4.16)$$

so wird die dem Knoten k_i folgende Kante als fehlerhaft interpretiert. Diese Verbindung wird entfernt, d.h das Kohonen-Netz hier in zwei Netze $K_i = (k_1, ..., k_i)$ und $K_{n-i} = (k_{i+1}, ..., k_n)$ zerlegt.

Man wendet dieses Vorgehen iterativ auf alle Kohonen-Netze an, bis ein jedes bzgl. Gleichung (4.16) stabil ist, d.h. keine Zerlegungen mehr stattfinden. Abbildung 4.9 zeigt die Korrekturen und neu entstehenden Netze aus Abbildung 4.8.

Fehlende Verbindungen

Nach den im vorangehenden Absatz beschriebenen "fehlerhaften" Verbindungen wird nun versucht, "offensichtlich fehlende" Verbindungen zu rekonstruieren. Abbildung 4.9 zeigt unter anderem Verbindungsmöglichkeiten, die ob des Abstandes und der Orientierung des Netzes als - im Sinne der glatten Fortsetzbarkeit - *offensichtlich* bezeichnet werden sollen (mit v bezeichnet). Betrachtet werden hierbei zwei Arten von (nicht existenten) Verbindungen :

- Verbindungen zwischen Netzen und damit Segmenten,

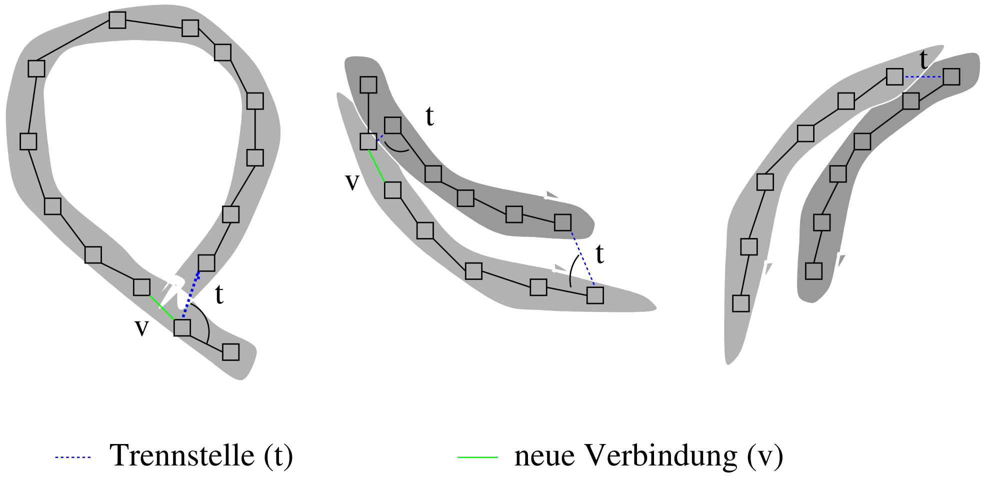


Abbildung 4.9: Trennen (gepunktete Linie) und Verbinden (durchgehende Linie) der Netze

- Verbindungensmöglichkeiten, die aus der Korrektur fehlerhafter Verbindungen resultieren.

Beispiele für letztere findet man in Abbildung 4.9. Durch das Zerlegen der Netze entstehen neue Verbindungsmöglichkeiten, welche im Idealfall den topologisch "korrekten" Verbindungen entsprechen. Analog zum Kriterium aus Gleichung (4.16) zur Trennung von Netzen, dient auch hier der Winkel aufeinanderfolgender Liniesegmente zueinander als Maß. Seien $K_n = (k_1^n, ..., k_n^n)$ und $K_m = (k_1^m, ..., k_m^m)$ 2-dimensionale Kohonen-Netze. Gilt für das Endsegment von K_n $e_n = [k_{n-1}^n, k_n^n]$ und das Anfangssegment $a_m = [k_1^m, k_2^m]$

$$\|k_1^m - k_n^n\| < \beta \tag{4.17}$$

und

$$\angle(e_m, a_m) < \gamma \;, \tag{4.18}$$

so werden die beiden Netze zu einem Netz K_{n+m} mit der Knotenfolge $(k_1^n, ..., k_n^n, k_1^m, ..., k_m^m)$ verbunden. Verbunden wird also wenn der Abstand klein genug ist (β) und die Orientierung von Anfangs- und Endsegment ähnlich (γ) ist. Ein Beispiel bietet ebenfalls Abbildung 4.9.

4.3 Glättung der 2D-Rekonstruktion durch Kalman Filter

Im vorangegangen Abschnitt 4.2 wurde der Verlauf der Segmente approximiert, diese Approximation ist genau so gut wie ihre Trainingsmenge, bestimmt durch Segmentierung verrauschter Bilddaten. Diese Kohonen-Netze wurden dann - losgelöst von den Trainingsmengen - nach modell- (glatter Verlauf) und damit verlaufsorientierten Verfahren korrigiert. Dabei sind teilweise Verläufe entstanden, die nicht mehr einer natürlichen, glatten Form entsprechen. Es haben sich also Fehler durch Segmentierung und Korrektur ergeben, die nun auf Basis der Kohonen-Netze korrigiert werden sollen. Hierfür verwenden wir das *Kalman Filter* [Kalman60], [Sorensen70]. Die Positionen der Knoten sind die Zustände eines Systems. Die topologische Ordnung der Netze liefert die "zeitliche" Reihenfolge. Abschnitt 4.3.1 liefert eine kurze Einleitung in die Theorie der Kalman Filterung nach [Zhang95]. In Abschnitt 4.3.2 wird die Anwendung auf Kohonen-Netze beschrieben.

4.3.1 Das Standard Kalman Filter - Mathematische Grundlagen

Das Verhalten eines dynamischen Systems läßt sich als Entwicklung einer Menge von *Zustandsvariablen* beschreiben. Eine beliebige Kombination der Belegungen der Zustandvariablen sei *Zustandvektor* genannt; er beschreibt einen Zustand des Systems. In der Praxis ist eine exakte Messung der Zustandvariablen im Normalfall nicht möglich; stattdessen stellt man fest, daß Meßergebnisse Funktionen der Zustandvariablen sind, die Messungen durch zufälliges Rauschen gestört sind. Auch das System in seiner Gesamtheit kann durch Rauschen überlagert sein. Somit stellt sich das Problem, die Zustandsvariablen aus den verrauschten Beobachtungen heraus zu schätzen.

Sei nun s Zustandsvektor und x' der Meßvektor, dann kann man ein dynamischen System (nach [Zhang95]) wie folgt beschreiben (diskrete Zeitschritte !):

$$s_{i+1} = h_i(s_i) + n_i, \tag{4.19}$$

$$f_i(x'_i, s_i) = 0, \qquad i = 0, 1, ... \tag{4.20}$$

Hierbei ist n_i der Vektor, welcher die zufällige Störung des Systems beschreibt. Diese wird häufig als *white noise* modelliert, dh.:

$$E[n_i] = 0 \qquad \text{and} \qquad E[n_i n_i^T] = Q_i \; . \tag{4.21}$$

Die den Fehler des Systems charakterisierende Kovarianzmatrix Q_i ist eine a-priori nicht bekannte Größe; in praktischen Anwendungen wird sie fast immer auf Basis von *Experimenten und Erfahrungswerten*, also eher geraten, bestimmt. Den zufälligen Fehler der Messungen (η) an sich wollen wir auch als additives white noise annehmen, so daß man die tatsächlich beobachtete Messung als

$$x_i = x'_i + \eta_i \tag{4.22}$$

erhält, wobei für die Erwartungswerte gilt:

$$E[\eta_i] = 0 \tag{4.23}$$

$$E[\eta_i \eta_j^T] = \begin{cases} \theta_{\eta_i} & , \quad i = j \\ 0 & , \quad i \neq j \end{cases} \; . \tag{4.24}$$

Die Meßfehler-Kovarianz θ_{η_i} muß genauso wie Q_i entweder durch Signaluntersuchungen oder "Raten" gewonnen werden. Normalerweise werden Q_i und η_i unabhängig voneinander ermittelt. Es sei also vorausgesetzt, daß zwischen dem Rauschen des Systems und der Messungen keine Korrelation besteht, also gilt:

$$E[\eta_i n_j T] = 0. \tag{4.25}$$

Im Rahmen dieser Arbeit wird das *Standard Kalman Filter* verwendet. Ist h_i eine lineare Funktion mit

$$s_{i+1} = H_i s_i + n_i \ , \tag{4.26}$$

und es ist möglich (basierend auf $f_i(x_i', s_i) = 0$) explizit eine lineare Beziehung

$$x_i = F_i s_i + \eta_i \tag{4.27}$$

anzugeben, dann kann man das Kalman Filter direkt anwenden. Die einzelnen Größen des Kalman Filters berechnen sich wie folgt:

- Vorhersage des Zustands:
  $$\hat{s}_{i|i-1} = H_{i-1}\hat{s}_{i-1}$$
- Vorhersage der Zustands-Kovarianz-Matrix:
  $$P_{i|i-1} = H_{i-1}P_{i-1}H_{i-1}^T + Q_{i-1}$$
- Kalman-Gain-Matrix:
  $$K_i = P_{i|i-1}F_i^T(F_i P_{i|i-1}F_i^T + \theta_{\eta_i})^{-1}$$
- Korrektur der Zustandsschätzung:
  $$\hat{s}_i = \hat{s}_{i|i-1} + K_i(x_i - F_i\hat{s}_{i|i-1})$$
- Korrektur der Zustands-Kovarianz-Matrix:
  $$P_i = (I - K_i F_i)P_{i|i-1}$$
- Initialisierung:
  $$P_{0|0} = \theta_{s_0}$$
  $$\hat{s}_{0|0} = E[s_0]$$

Abbildung 4.10 zeigt die Funktionsweise der Kalman-Filterung als Blockdiagramm. es gelten die obigen Initialisierungen.

Zum Zeitpunkt t_i wird auf Basis der vohergehenden Zustandsschätzung $\hat{s}_{i-1}$ die Vorhersage $\hat{s}_{i|i-1}$ generiert. Aus der Zustands-Kovarianz-Matrix P_{i-1} wird die Vorhersage $P_{i|i-1}$

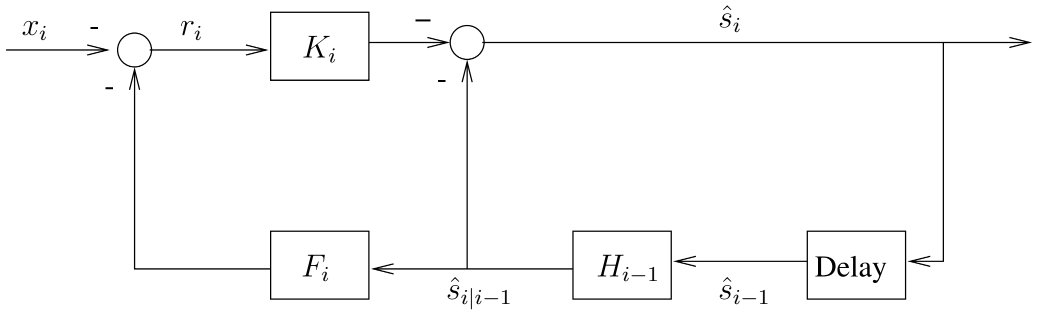


Abbildung 4.10: Kalman Filter Blockdiagramm

abgeleitet, genauer gesagt extrapoliert. Mittels derer werden dann die neue *Kalman-Gain-Matrix* K_i und Kovarianzmatrix P_i berechnet. Man berechnet ebenso $F_i\hat{s}_{i|i-1}$, die beste Schätzung der Messung zum Zeitpunkt t_i. Die Differenz aus tatsächlicher Messung x_i und dieser Schätzung

$$r_i = x_i - F_i\hat{s}_{i|i-1} \tag{4.28}$$

wird das *Residual* genannt. Abschließend wird r_i mit K_i gewichtet um den Korrekturterm

$$c_i = K_i(x_i - F_i\hat{s}_{i|i-1}) \tag{4.29}$$

zu erhalten. Der korrigierte Zustand ergibt sich dann zu $\hat{s}_i = \hat{s}_{i|i-1} + c_i$.

Das Kalman Filter liefert einen linearen, rekursiven Algorithmus, eine optimale Schätzung des unbekannten Zustandes eines linearen dynamischen Systems aufgrund von verrauschten Messungen in diskreten Zeitintervallen. Eine tiefergehende theoretische Fundierung des Kalman Filters findet man u.a. in [Kalman60], [Jazwinsky70] oder auch [Maybeck82].

Im Folgenden wird die Anwendung der Theorie auf den Spezialfall Kohonen-Netz beschrieben.

4.3.2 Anwendung auf Kohonen-Netze

In dieser Arbeit wird das Kalman-Filter zur Schätzung und Glättung des Kabelverlaufs - repräsentiert als eine Folge von 2D-Koordinaten, den Kohonen-Knoten - verwendet.

Sei $K_n = (K_1, ..., K_n)$ ein Kohonennetz, dann soll der Prozeß lokal beschrieben sein durch

$$k'_{i+1} = (k_i - k_{i-1}) + k_i \tag{4.30}$$

Die Betrachtung beginnt mit $i = 2$. Ein Zustand hat also die Form

$$s_i = \begin{pmatrix} s_i^x \\ s_i^y \end{pmatrix} \tag{4.31}$$

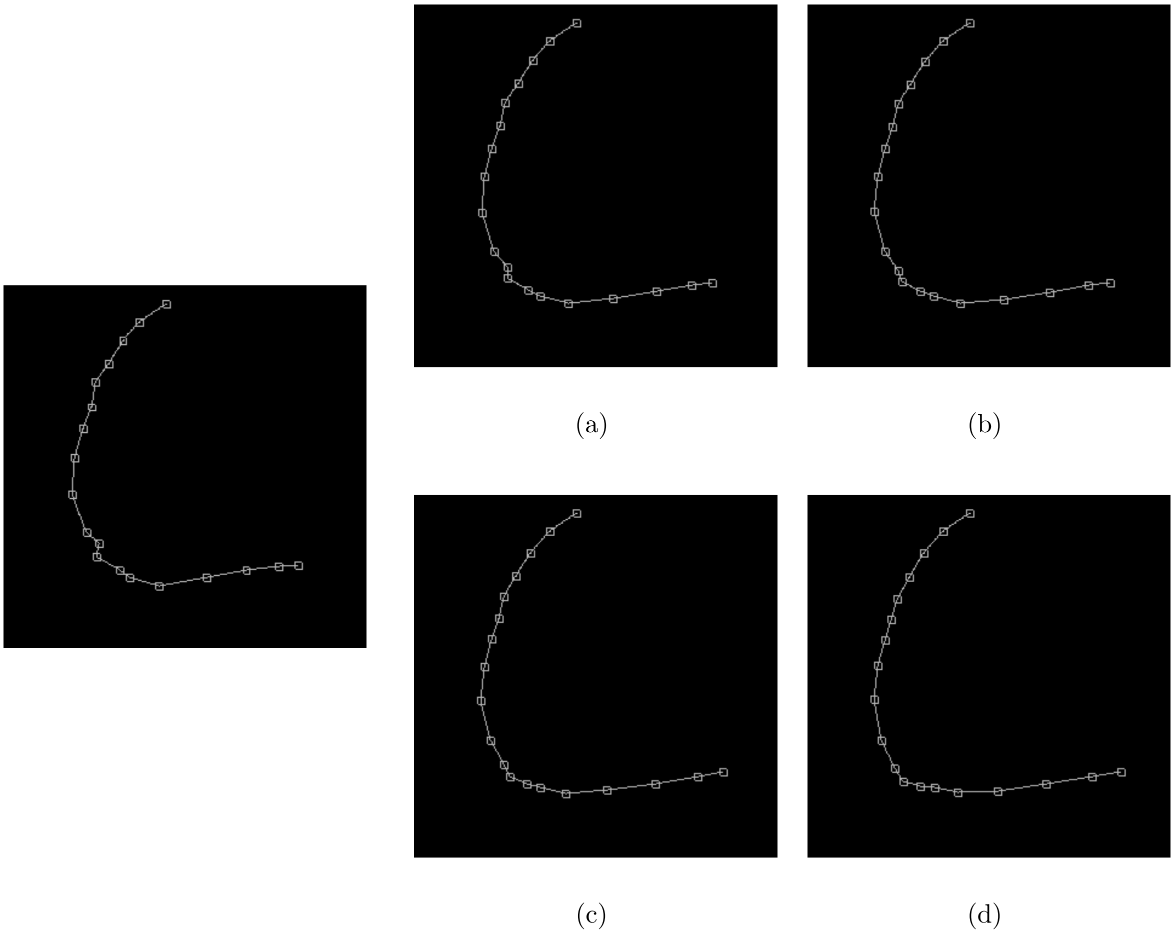


Abbildung 4.11: Kohonen-Netz und Glättung durch iteriertes Kalmanfiltern (a)-(d)

Insgesamt leitet sich die Transitionsvorschrift der Form

$$s_{i+1} = H_i s_i + \omega \tag{4.32}$$

$$= \begin{pmatrix} 1 + \frac{s_i^x - s_{i-1}^x}{s_i^x} & 0 \\ 0 & 1 + \frac{s_i^y - s_{i-1}^y}{s_i^y} \end{pmatrix} s_i + \omega \tag{4.33}$$

ab. Das Systemrauschen wird sei einfach durch die Kovarianzmatrix

$$Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{4.34}$$

beschrieben. Da alle Zustandsgrößen auch gleichzeitig Meßgrößen sind, gilt auch für die Messungsmatrix

$$F = F_i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \text{für alle i = 1,... .} \tag{4.35}$$

Die Meßfehlerkovarianzen θ_x und θ_y werden ebenfalls als Eins angenommen. Abbildung 4.11 zeigt exemplarisch die Glättung eines Kohonen-Netzes durch iteriertes Anwenden des Kalman Filters. Links befindet sich die Ausgangssituation, während (a)-(d) die Ergebnisse nach 1-4 Iterationen darstellen.

4.4 Generisches Formmodell für Kabel : Splines

Nach einer kurzen motivierenden Einleitung, wird in Abschnitt 4.4.2 die sogenannte *B-Spline-Darstellung* von Splines erläutert und abschließend ein *Kabeltest* auf Basis der Splines vorgestellt.

4.4.1 Was ist "gebogen" ? - die Minimaleigenschaft

Im frühen 19.Jahrhundert, der Anfangszeit des Schiffdesigns, bestand das Problem, mit freier Hand durch eine gegebene Menge von Punkten eine glatte Kurve zu zeichnen. Man löste das Problem dadurch, daß man die Datenpunkte mit Metallgewichten ("ducks" genannt) versah und eine dünne elastische Holzlatte ("spline") zwischen diesen hindurch legte (siehe Abbildung 4.12). Die Holzlatte nimmt eine solche Position ein, daß ihre elastische Energie minimiert wird. Das mathematische Äquivalent dieser Latte ist eine Kurve s, deren elastische Energie durch

$$E = \int (\kappa(s))^2 ds \tag{4.36}$$

gegeben ist, wobei κ die Krümmung der Kurve bezeichnet. Die Krümmung der meisten Kurven ist durch Integrale und Quadratwurzel gegeben und daher schwierig zu behandeln. Man approximiert Gleichung (4.36) deshalb durch ein einfacheres Integral:

$$\hat{E} = \int \left[\frac{d^2}{du^2} s(u) \right]^2 du. \tag{4.37}$$

Hierbei ist $\hat{E}$ ein Vektor, den man erhält, in dem man jede Komponente von s integriert. In dieser Arbeit werden C^2-stetige, kubische, interpolierende Splines verwendet, welche die

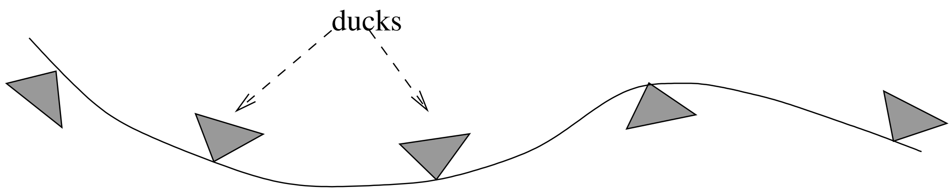


Abbildung 4.12: Spline-Interpolation : der "spline" muß durch die Datenpunkte, gekennzeichnet durch die "ducks", verlaufen

folgende vorteilhafte Eigenschaft haben: Von allen C^2 -stetigen Funktionen, die gegebene Datenpunkte an gegebenen Parameterwerten interpolieren und denselben Endbedingungen genügen, liefert der kubische Spline den kleinsten Wert für jede Komponente von $\hat{E}$. Zur Verifikation sei auf [Farin94] oder [Hoschek92] verwiesen. Genau diese Eigenschaft wird hier auch dem Verlauf eines Kabels unterstellt.

Im folgenden Abschnitt werden kurz die mathematische Grundlagen der Berechnung von Splines in der B-Spline Darstellung erläutert.

4.4.2 Spline-Funktionen in der B-Spline Darstellung

Spline-Funktionen spielen in der Approximationstheorie eine wichtige Rolle; man findet daher umfangreiche Literatur aus Sicht der Statistik (z.B. [Eubank88]), geometrischen Datenverarbeitung(s.o.) oder auch Robotik (z.B. [Olomski89], [Ude96]).

Eine Splinefunktion der Ordnung $2m$ auf den Knoten $\tau_1 < ... < \tau_{\tilde{N}}$ ist durch

$$s(t) = \sum_{i=0}^{2m-1} \gamma_i t^i + \sum_{j=1}^{\tilde{N}} \sum_{i=0}^{q_j} \delta_{ij}(t-\tau_j)_+^{2m-1-i}, \qquad 0 \leq q_j < m \;, \tag{4.38}$$

definiert, wobei

$$(t)_+ = \begin{cases} 0 & , \quad t < 0 \\ t & , \quad \text{sonst} \end{cases} \;. \tag{4.39}$$

Spline-Funktionen der Ordnung $2m$ verhalten sich offenbar auf jedem offenen Intervall (τ_j, τ_{j+1}) als Polynome des Grades $2m-1$. Der Raum aller Splinefunktionen auf der Knotenfolge $\tau = \{\tau_i\}$ wird als $I\!P_{m,\tau,q}$ bezeichnet. Er ist linear und von der Dimension $N = 2m + \sum_{j=1}^{\tilde{N}} q_j$.

Darstellung 4.38 ist für die numerische Berechnung wenig geeignet. Man generiert eine berechnungsfreundlichere Basis von $I\!P_{m,\tau,q}$ mit Hilfe der sogenannten *B-Splines*. Sei $t = \{t_i\}_{i=1}^{N+2m}$ sei eine monoton wachsende Folge von Knotenpunkten, in der zwei neue Knoten $a < \tau_1$ und $b > \tau_{\tilde{N}}$ genau $2m$-mal und die Knoten τ_i q_i-mal auftreten. B-Splines auf $\{t_i\}$ sind definiert durch:

$$B - i(t) = (t_{i+2m} - t_i)[t_i, ..., t_{i+2m}] \cdot (-1)_+^{2m-1}, \qquad i = 1, ..., N \;. \tag{4.40}$$

Hierbei sei mit $[t_i, ..., t_{i+2m}]f$ die $2m$-te dividierte Differenz der Funktion f auf den Knoten $t_i, ..., t_{i+2m}$ bezeichnet (vgl. [Böhmer74]). Ferner ist $t_i < t_{i+m}$ für alle $m < i \leq N$ gefordert. Für $m = 2$ bekommt man die kubischen B-Splines. B-Splines auf dem Intervall $[t_1, t_{N+2m}] = [a, b]$ gehören zu $I\!P_{m,\tau,q}$ und bilden auf diesem Intervall eine Basis des Raumes ([deBoor78]). Jede Spline-Funktion kann somit als Linearkombination der B_i ausgedrückt werden

$$S(t) = \sum_{i=1}^{N} \alpha_i B_i(t), \qquad t \in [a, b] \;. \tag{4.41}$$

Hauptvorteil zu 4.38 ist die Minimalträgereigenschaft

$$B_i(t) = 0 \qquad \text{für alle } t \notin [t_i, t_{i+2m}] \ . \tag{4.42}$$

Somit folgt, daß für $t_j < t < t_{j+1}$ gilt

$$s(t) = \sum_{i=j-2m+1}^{j} \alpha - iB_i(t) \ . \tag{4.43}$$

Zur Berechnung eines Wertes der Spline-Funktion in der B-Spline Darstellung müssen nur die Werte der $2m$ Basis-Funktionen berechnet werden. B-Splines sind desweiteren überall positiv

$$B_i(t) > 0 \qquad \text{für alle } t \in [t_i, t_{i+2m}] \tag{4.44}$$

und normalisiert, das heißt

$$\sum_{i=1}^{N} B_i(t) = 1 \qquad \text{für alle } t \in [a, b] \ . \tag{4.45}$$

Mit Hilfe der in [deBoor78] beschriebenen Rekursionsformeln lassen sich die Werte einzelner B-Splines effizient berechnen. Es steht somit ein Werkzeug zur Berechnung der Splines zur Verfügung.

4.4.3 Verbinden mit B-Splines - Der Kabeltest

Aus der Annahme, daß Splines den Verlauf von Kabelobjekten geeignet beschreiben, wird im Folgenden ein Algorithmus abgeleitet, welcher mögliche Vezbindungen zwischen den bis dato erhaltenen Kabelsegmenten überprüft und das derart beschrieben Objekt als Kabel oder nicht klassifiziert. Hierfür sei mit

$$K^j = (k_1^j, ..., k_{n_j}^j)$$

für $j \in \{1, ..., m\}$ und $n_j \in \{I\!N\}$ eine Menge von 2-dimensionalen Kohonen-Netzen gegeben. Es werden nun für alle $j \in \{1, ..., m\}$ der Endknoten mit jedem Anfangsknoten $l \in \{1, ..., m\}$ verbunden und für das resultierende Kohonen-Netz

$$K^{jl} = (k_1^j, ..., k_{n_j}^j, k_1^l, ..., k_{n_l}^l)$$

der entsprechende Spline berechnet. Dieser wird nun auf seine Glattheit geprüft. Hierzu wird der Spline durch eine Reihe von möglichen Stützstellen repräsentiert, wie in Abbildung 4.13 zu sehen. Diese Folge von Knoten wird nun mit dem trivialen Test Gleichung (4.16) auf Glattheit geprüft.

Der Spline stellt folglich eine Art von Extrapolation dar; es wird versucht zu testen, ob zwischen zwei Segmenten eine energieminimierende, glatte Verbindung möglich ist. Abbildung 4.14 zeigt füer einen einfachen Fall alle möglichen Kombinationen und das Endresultat.

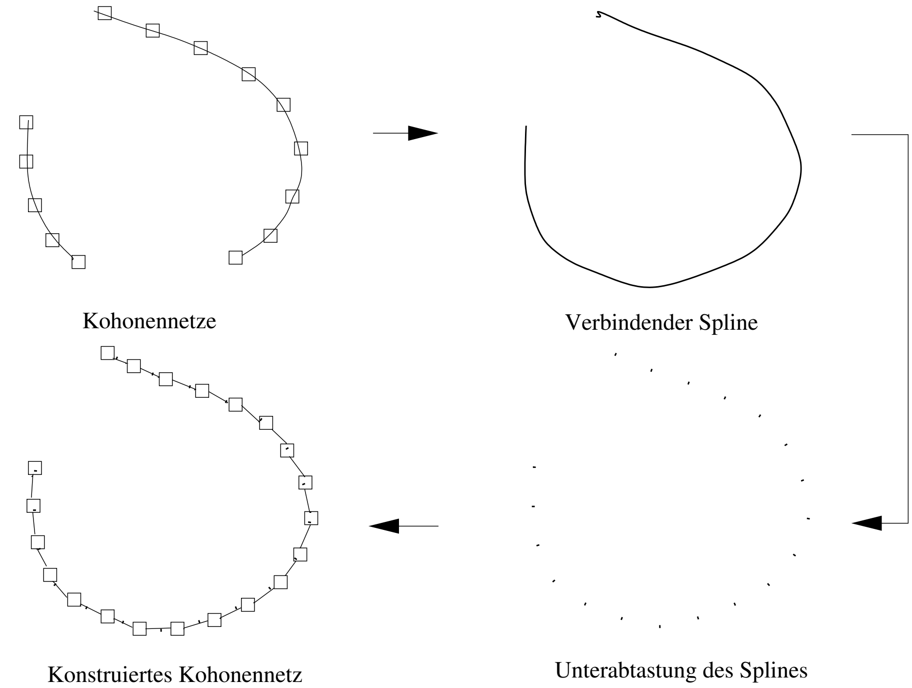


Abbildung 4.13: Spline in der Darstellung als Kohonennetz

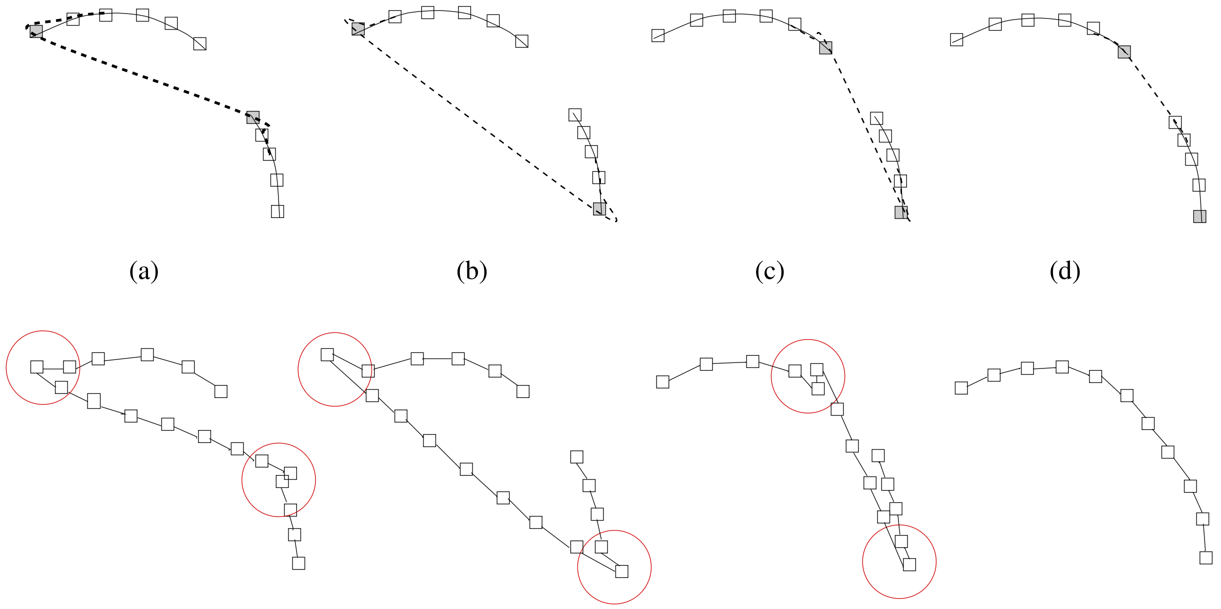


Abbildung 4.14: Splinetest - ein einfaches Beispiel und alle möglichen Verbindungen

4.5 Beispiel (Teil II) : Formbasierte Objekterkennung

Zum Abschluß des Kapitels folgt nun der zweite Teil der Beispielszenen aus Abschnitt 4.5.

Beipiel 1 : Kabelsalat

Durch die Vorauswahl werden kleinere Regionen entfernt. Die geglätteten Approximiationen der Kabel, lassen sich mit den verwendeten Methoden nicht mehr weiter zusammenfügen (Abbildung 4.15).

Beipiel 2 : Einfache Szene

Hier werden die Nicht-Kabelobjekte durch die Vorauswahl entfernt und nach der Glättung läßt sich eine Verbindung rekonstruieren (eingekreist) (Abbildung 4.16) .

Beipiel 3 : Monitor

Beim Monitor-Beispiel werden nach dem Entfernen der kleinen Segmente drei Verbindungen hergestellt (Abbildung 4.17).

Abschließend sind in Abbildung 4.18 für alle drei Beispiele noch einmal die Szene mit den eingezeichneten Kohonen-Netzen und die jeweils rekonstruierten Kabelverläufe als Splines zu sehen.

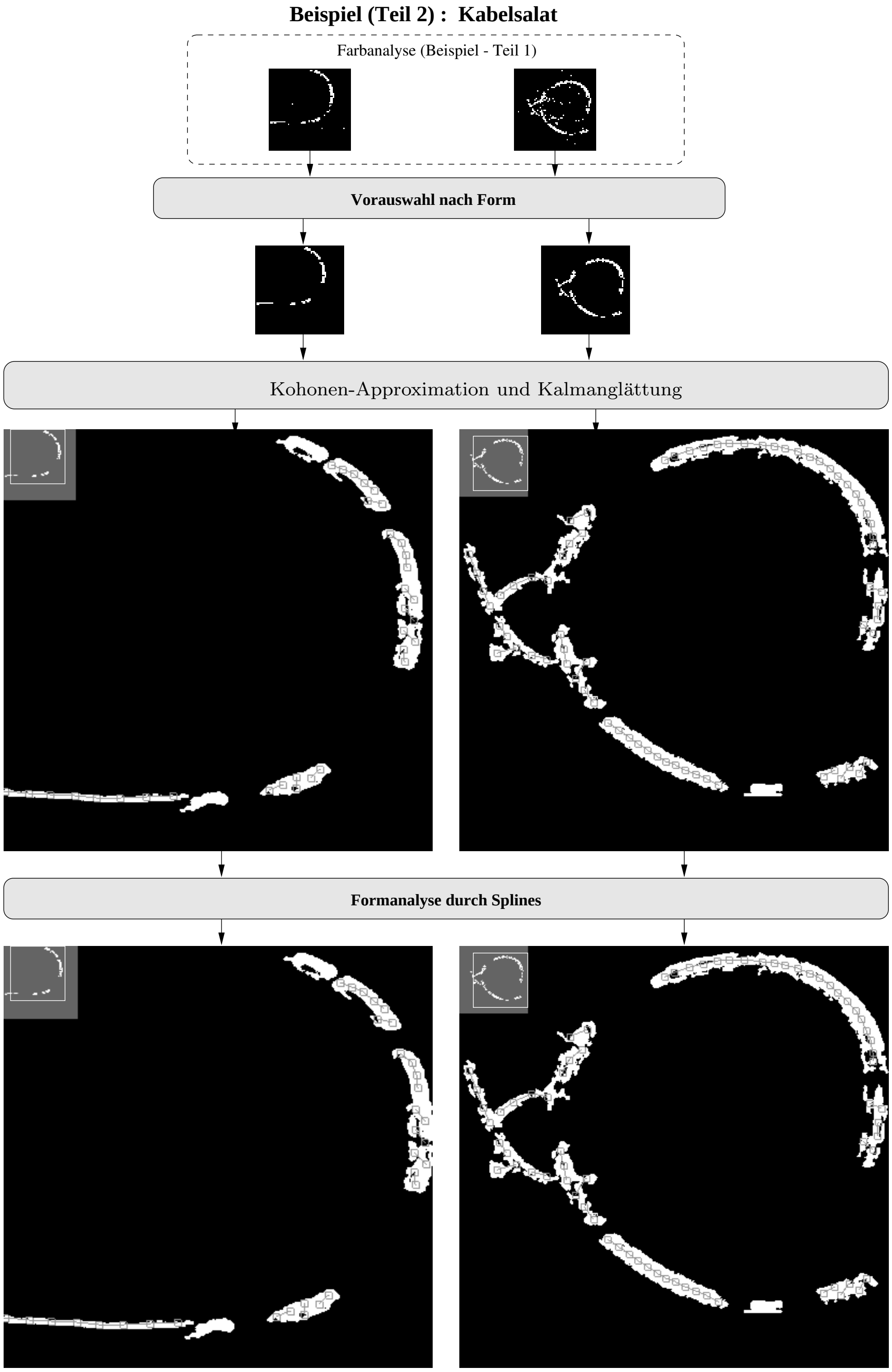


Abbildung 4.15: Zweiter Teil des Beispiels "Kabelsalat" - Formbasierte Objekterkennung

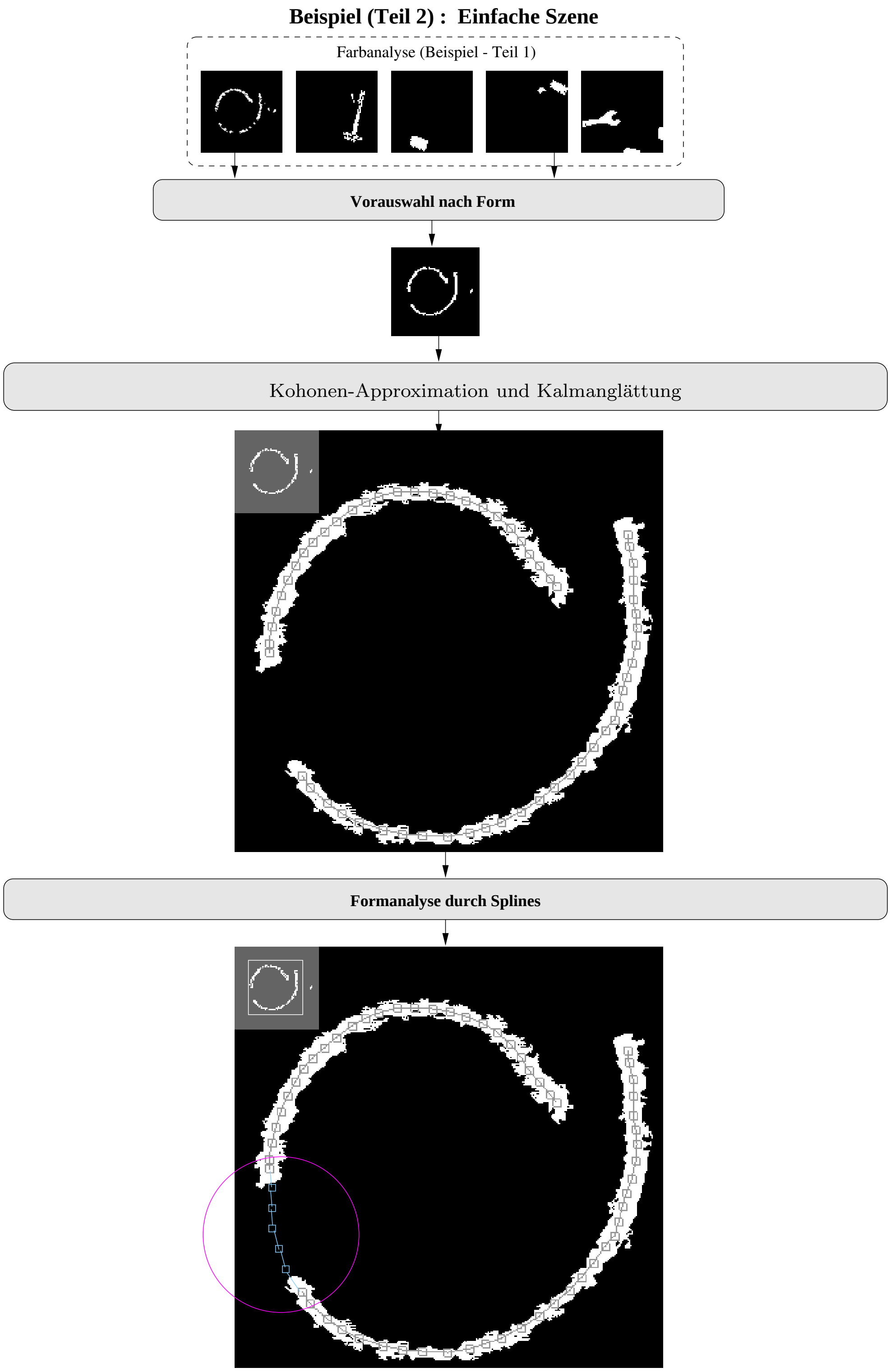


Abbildung 4.16: Zweiter Teil des Beispiels "Einfache Szene" - Formbasierte Objekterkennung

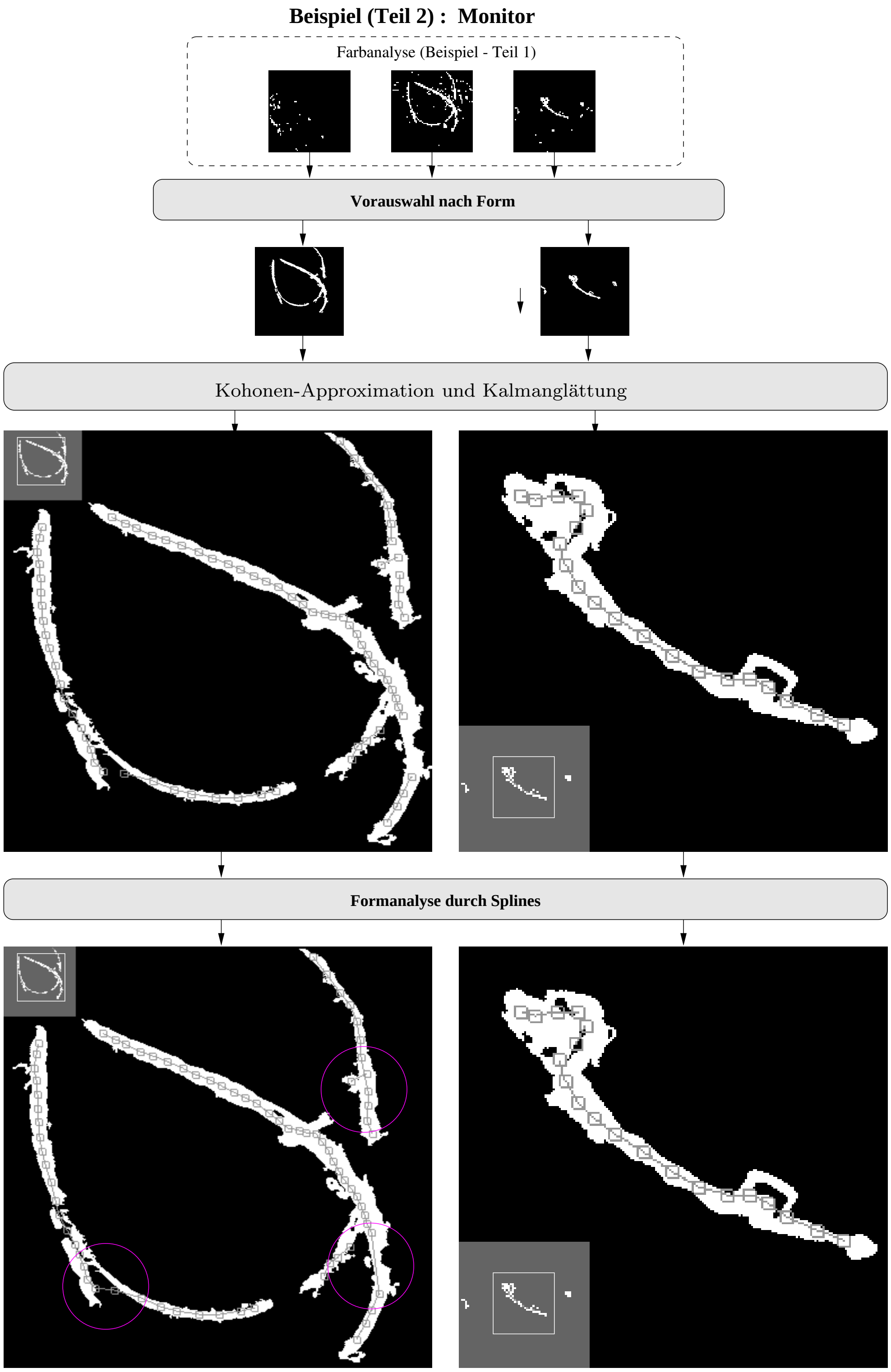
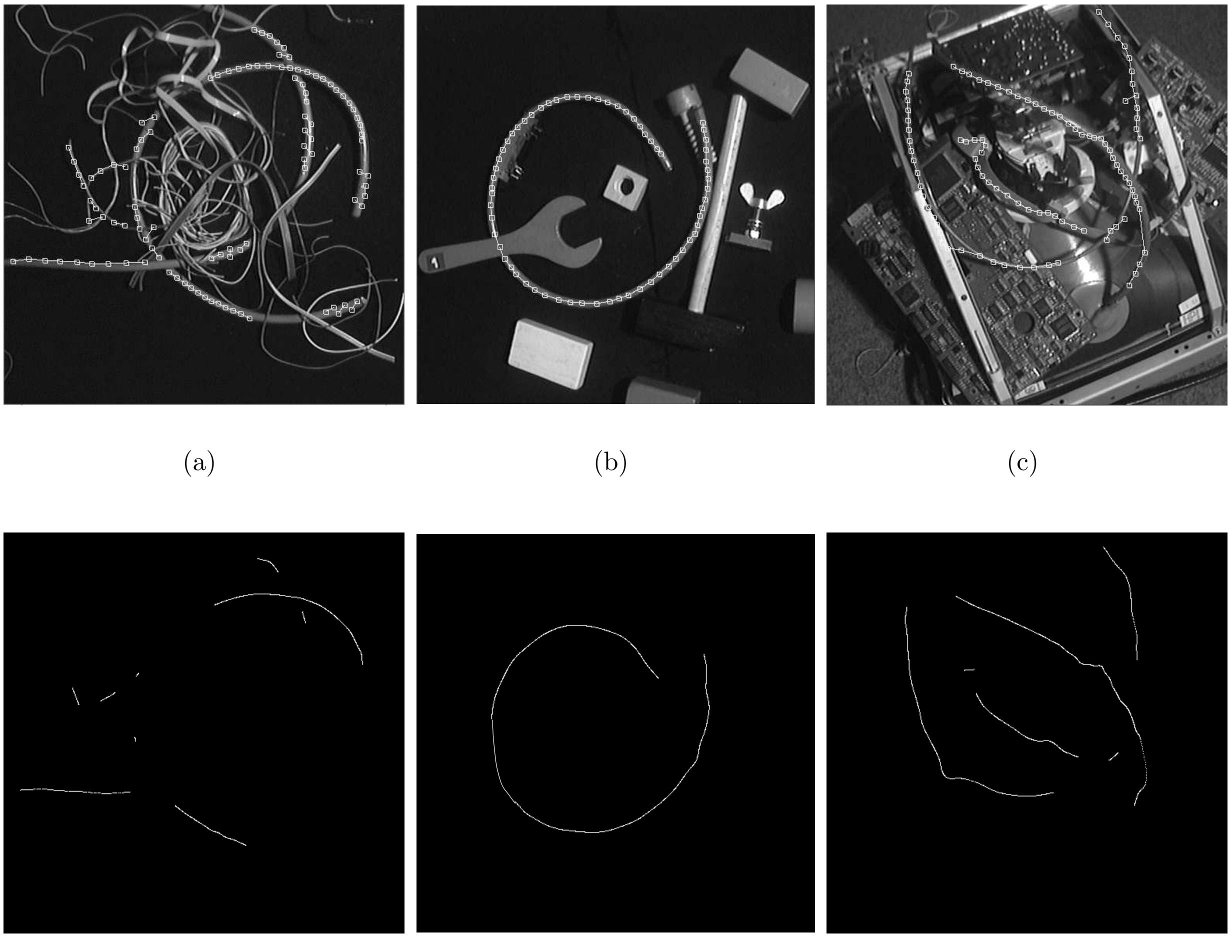


Abbildung 4.17: Zweiter Teil des Beispiels "Monitor" - Formbasierte Objekterkennung



(a) (b) (c)

(d) (e) (f)

Abbildung 4.18: Erkannte Kabelverläufe und dazugehörige Splines als Binärbild (Grundlage für Rekonstruktion)

Kapitel 5

3D - Rekonstruktion mittels der Epipolaren Geometrie

Grundzielsetzung der Arbeit ist das Greifen der Objekte (Kabel). Hierzu werden sie in ihrer Lage im Welt(=Roboter)koordinatensystem rekonstruiert. Dieses Kapitel beschreibt die dazu benötigten Methoden, graphisch dargestellt in Abbildung 5.1. In Abschnitt 5.1 wird die Kalibrierung der Kameras unter Annahme des Lochkameramodells beschrieben. Anschließend wird, darauf basierend, die Stereokalibrierung, der Übergang von (2D , 2D)-Koordinaten zu 3D-Weltpositionen erläutert. In Abschnitt 5.2 wird das Problem der Korrespondenzen zwischen Bildpunkten in den Stereobildern mittels der *epipolaren Geometrie* betrachtet. Basis hierzu sind die Binärbilder des Kabelverlaufs. Die rekonstruierten 3D-Positionen werden zur Akkumulation der Daten in einem *3D-Besetztheitsgitter* (Kapitel 6) gesammelt.

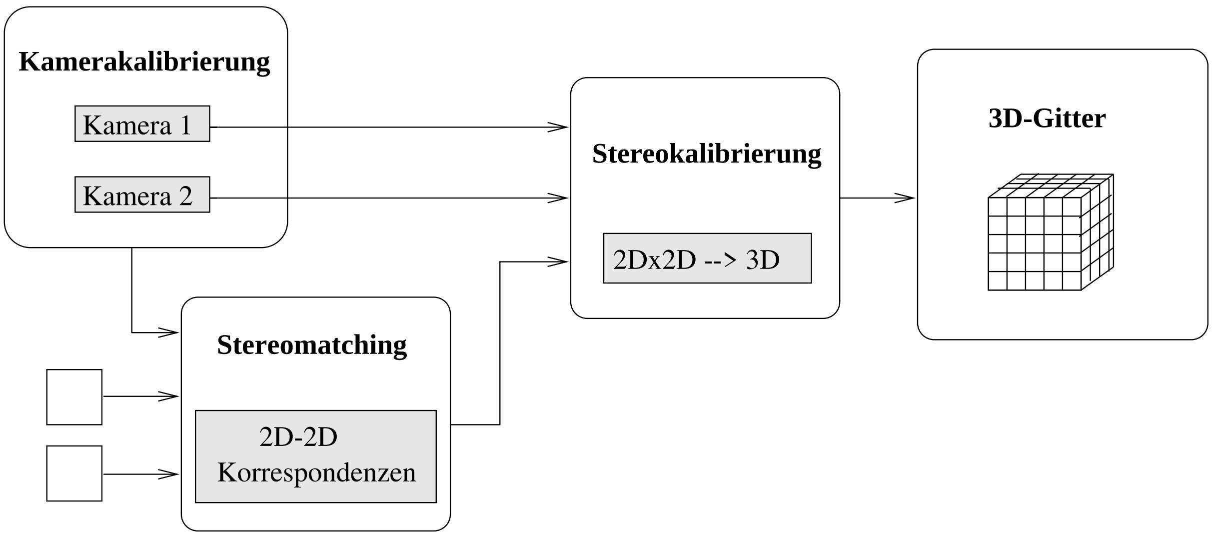


Abbildung 5.1: Übersicht der in Kapitel 5 vorgestellten Systemkomponenten

5.1 Kamerakalibrierung

Im folgenden Abschnitt wird die Transformation der Bildkoordinaten in die Welt- bzw. Roboterkoordinaten beschrieben. Die verwendete Optik mit geringer Brennweite macht eine vorherige Bildentzerrung notwendig. Abschnitt 5.1.1 erläutert das dafür eingesetzte Verfahren. Die folgenden Abschnitte 5.1.2 und 5.1.3 beschreiben die Methoden zur Kamerakalibrierung und Berechnung der Koordinatentransformation von 2D×2D nach 3D. Abschnitt 5.1.4 beschreibt die Anpassung der Kalibrierungsdaten an eine veränderte Kameraposition. Abschließend wird in Abschnitt 5.1.5 eine Fehleranalyse der Kalibrierung mit Zahlenbeispiel durchgeführt.

5.1.1 Bildentzerrung

Da die Objektszenen aus relativ geringer Distanz betrachtet werden, gleichzeitig aber ein gewisser Überblick über die Szene notwendig ist, wird ein Weitwinkelobjektiv eingesetzt. Der Einsatz eines Kameraobjektives mit geringer Brennweite (f = 5.6 mm) bedeutet eine verzerrte Darstellung im Bild.

Das in den folgenden Abschnitten vorgestellte Verfahren zur Kamerakalibrierung und zum Stereomatching basiert auf der Annahme des Lochkameramodells. Hieraus erwächst die Notwendigkeit, vorab eine Korrektur der Verzerrungen vorzunehmen. In Abbildung 5.2 ist der Unterschied in der Bilddarstellung zwischen einer 25mm- und der verwendeten 5.6mm-Optik dargestellt; zur Verdeutlichung sind jeweils noch die jeweiligen Gradientenbetragsbilder hinzugefügt. Die hier sichtbare Verzerrung ist radial, d.h., daß von einem Zentrum ausgehend mit wachsendem Abstand größere Verzerrungen auftreten. In [Jähne93](S.23) findet man ein Modell (zurückgehend auf [Lenz88]), welches die Entzerrung geeignet approximiert. Hierzu werden Entzerrung in x- und y-Richtung getrennt betrachtet. Erster Schritt ist die Bestimmung des Zentrums der Verzerrung. Hierzu wird die Kamera (wie in Abbildung 5.3(a) dargestellt) senkrecht zur Weltebene des $N \times N$-Kalibrierungsrasters positioniert. Nun werden die Rasterpunkte $P_r(i,j)$ (*Ist-Positionen*) extrahiert. Die soge-

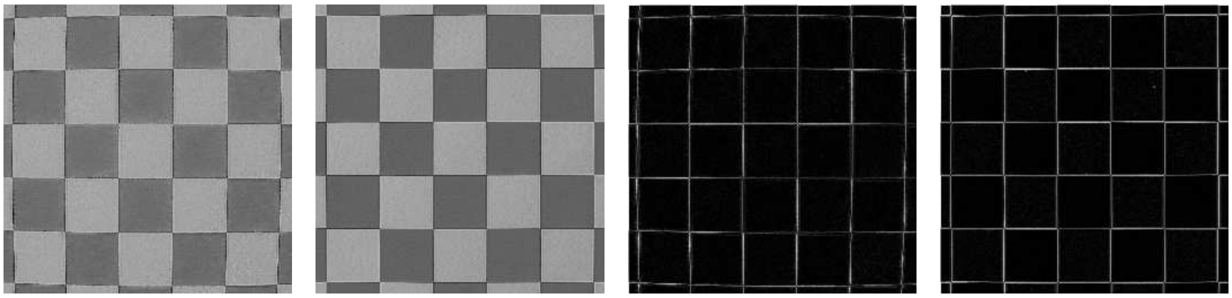
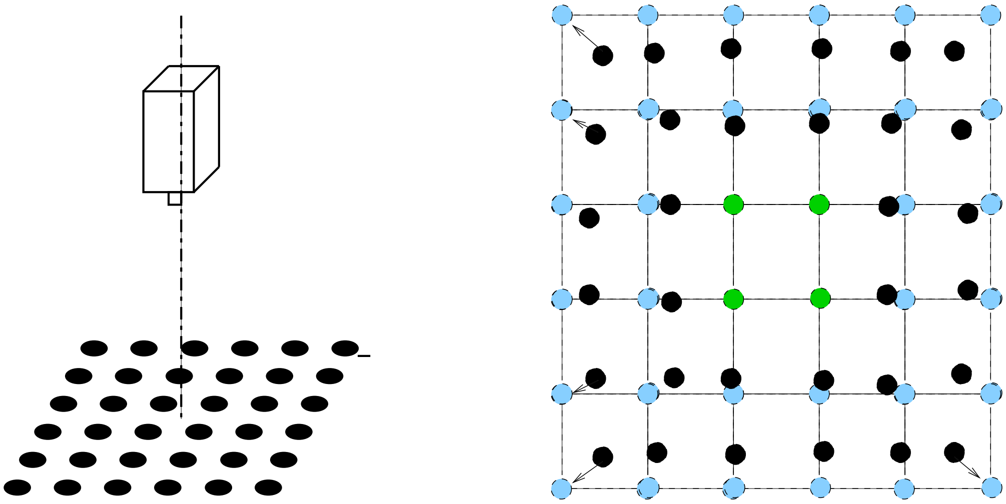


Abbildung 5.2: Schachbrettmuster aufgenommen mit 5.6mm (jeweils links) und 25mm Optik (jeweils rechts) und zugehörige Gradientenbetragsbilder



(a) (b)

Abbildung 5.3: Orientierung der Kamera zur Bildentzerrung (a) und Paßpunkte-Raster (b)

nannten *Paßpunkte* $P_p(i,j)$ (*Soll-Positionen*) werden als Extrapolation der mittleren vier Punkte des Kalibrierungsrasters, welche als fehlerfrei angenommen werden, errechnet (Abbildung 5.3(b)). Der Fehler für einen Rasterpunkt $P_r(i,j) = (x^r_{ij}, y^r_{ij})$ ergibt sich schlicht zu:

$$E(i,j) = \sqrt{(x^r_{ij} - x^p_{ij})^2 + (y^r_{ij} - y^p_{ij})^2}. \tag{5.1}$$

Eine geeignete Näherung des Zentrums erhält man durch den Bildpunkt mit der geringsten Verzerrung (x_z, y_z). Setzt man

$$E_{max} = \max\{E(i,j)|i,j = 1,...,N\},$$

so berechnet er sich als gewichtetes Mittel durch

$$x_z = \frac{1}{E_{sum}} \sum_{i=0}^{N} \sum_{j=0}^{N} (E_{max} - E(i,j)) \cdot x^r_{ij} \tag{5.2}$$

und

$$y_z = \frac{1}{E_{sum}} \sum_{i=0}^{N} \sum_{j=0}^{N} (E_{max} - E(i,j)) \cdot y^r_{ij} \; . \tag{5.3}$$

mit $E_{sum} = \sum_{i=0}^{N} \sum_{j=0}^{N} (E_{max} - E(i,j))$. Für einen gegebenen Bildpunkt mit den Koordinaten $(x,y) \in 512 \times 512$ berechnet sich nun der entzerrte Bildpunkt, genauer seine

Koordinaten, zu

$$x' = \frac{x - x_z}{1 + \alpha_x \|(x - x_z, y - y_z)\|^2} + x_z. \tag{5.4}$$

bzw.

$$y' = \frac{y - y_z}{1 + \alpha_y \|(x - x_z, y - y_z)\|^2} + y_z \tag{5.5}$$

Der Faktoren $\alpha_{x/y}$ sind die einzigen, die Verzerrung beschreibenden Parameter und müssen für die jeweilige Kamera bestimmt werden. In diesem Fall sollen die $\alpha_{x/y}$ so gewählt werden, daß der Gesamtfehler

$$E_{ges} = \sum_{i=0}^{N} \sum_{j=0}^{N} E(i,j) \tag{5.6}$$

minimiert wird. Hierzu benötigte numerische Verfahren findet man u.a. in [Stoer89] und sollen hier nicht weiter betrachtet werden. In der Praxis findet man Werte in der Größenorbnung von $\alpha_{x/y} \approx 10^{-5}$.

Die entzerrende Abbildung ist in unserem Falle definiert als

$$f_{ent} : 512 \times 512 \rightarrow I\!R \times I\!R\ .$$

Im Regelfall ist f_{ent} keine bijektive Abbildung auf 512×512, so daß bei einer Bildentzerrung undefinierte Pixelwerte im Zielbild entstehen. Abbildung 5.4 zeigt eine Szene und

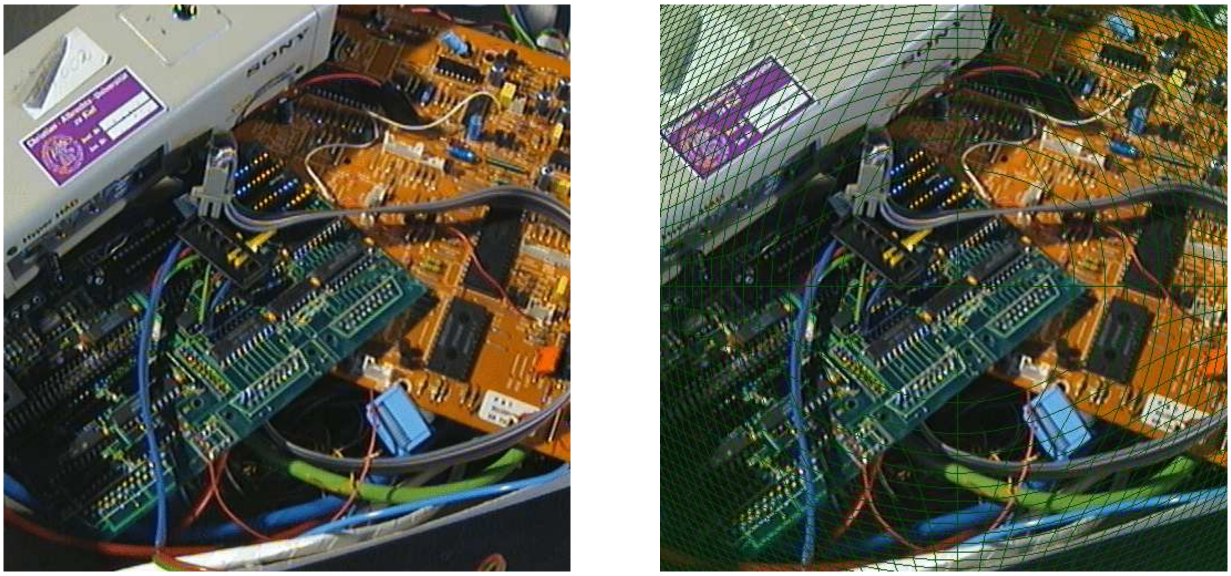


Abbildung 5.4: Auftreten undefinierter Pixelwerte (rechts) bei Bildentzerrung

das zugehörige entzerrte Bild. Diese undefinierten Bildpunkte werden durch Glättung des Bildes aus den benachbarten Pixelwerten gewonnen.

Da jedes Kamerabild entzerrt wird, bevor es verarbeitet werden kann, kommt der Geschwindigkeit dieser Operation hohe Bedeutung zu. Zur Beschleunigung der Berechnung, wird die Entzerrungsabbildung f_{ent} als Look-Up-Table gespeichert.

Im folgenden Abschnitt sind einmal beispielhaft Kamerabilder, Fehler und Entzerrungsparameter dargestellt.

Ein Beispiel

Abbildung 5.5 zeigt ein Kamerabild und die Bildverzerrung (als Pfeile zwischen Bild- und Passpunkten). In nachstehender Tabelle 5.1 sind die minimalen, maximalen und durchschnittlichen Abweichungen in Pixeln aufgeführt.

Betrachtete man die Beträge der Fehler, so gelangt man von - zur Lösung von Korrespondenzproblemen inakzeptablen - $\approx$ 25 Pixeln zu immerhin ca. 6 Pixeln maximaler Abweichung. Auffällig ist auch der geringe durchschnittliche Fehler und die großen Bildregionen mit Fehlern von maximal 2 Pixeln, ein Schwellwert, der gemeinhin als Mindestanforderung an die Exaktheit für die Verwendung geometrischer Verfahren zur 3D-Rekonstruktion gilt.

	ohne Entzerrung			α	mit Entzerrung		
	durchnittlicher Fehler	minimaler Fehler	maximaler Fehler		durchnittlicher Fehler	minimaler Fehler	maximaler Fehler
Kamera 1	7.826	0.015	25.136	5.5e-6	2.534	0.012	7.021
Kamera 2	6.946	0.081	24.837	4.8e-6	2.045	0.073	5.987

Tabelle 5.1: Verzerrung der Passpunkte in Pixeln

5.1.2 Projektionsmatrix

Ausgehend von der Annahme des Loch-Kamera-Modells als Beschreibung der Projektion von Weltkoordinaten in die Bildebene (sinnvoll nur bei vorheriger Bildentzerrung), läßt sich nun die Beziehung zwischen einem drei-dimensionalen Weltkoordinatensystem und einem zwei-dimensionalen Kamerakoordinatensystem mittels einer *Projektionsmatrix* $\widetilde{P}$ der Form

$$\widetilde{P} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} = (Pp) \tag{5.7}$$

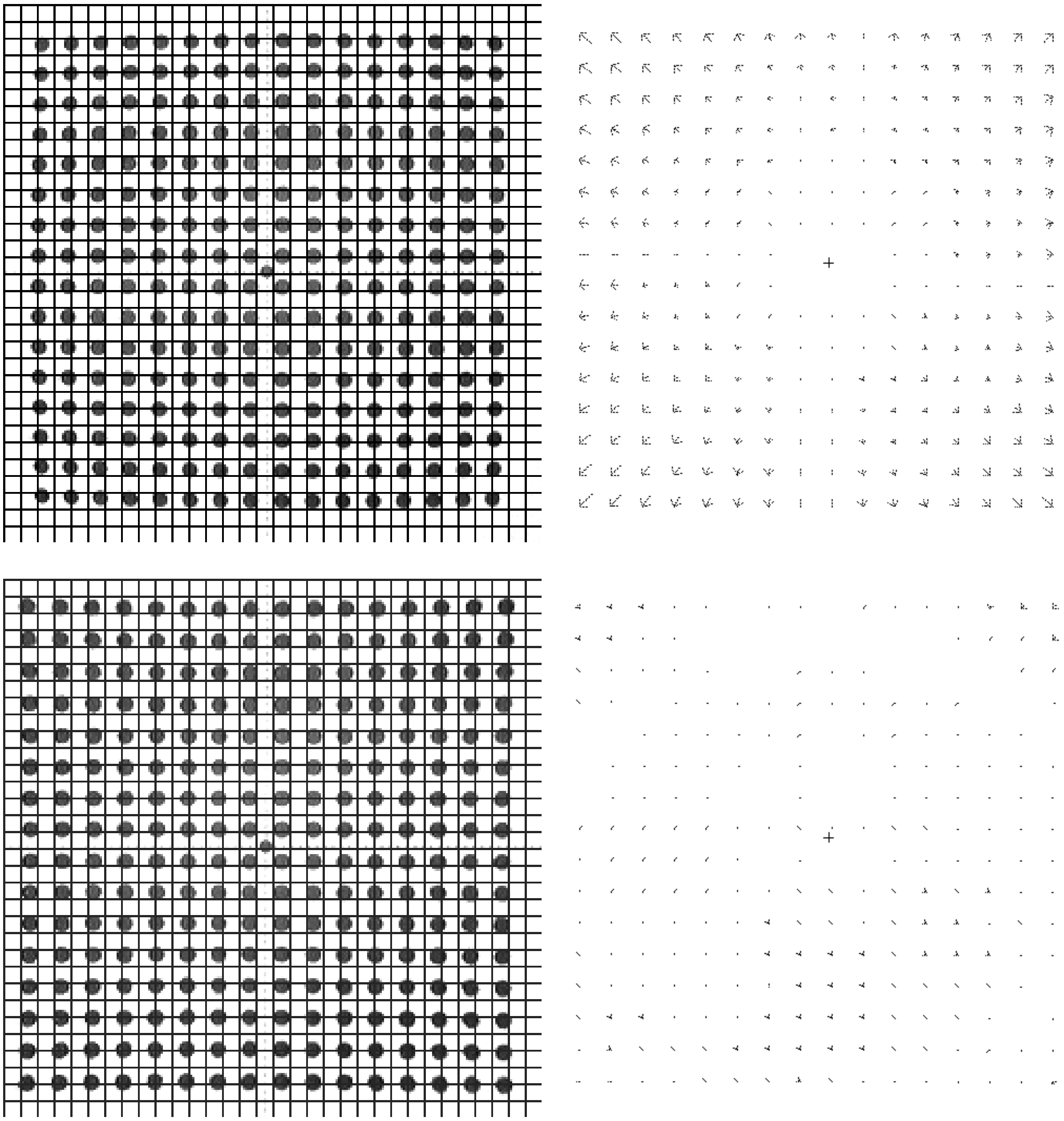


Abbildung 5.5: Kamerabild und Verzerrungsfehler vor (oben) und nach (unten) der Entzerrung

beschreiben [Faugeras93]. Die hintere Blockschreibweise als 3×3 Matrix P und 1×3-Vektor p sei hier bereits für späteren Gebrauch eingeführt. $\widetilde{P}$ beschreibt hierbei die Abbildung des projektiven Raumes $\mathcal{P}^3$ in die projektive Ebene $\mathcal{P}^2$ durch

$$\widetilde{P} : \mathcal{P}^3 \to \mathcal{P}^2 \ , \ \widetilde{M} \mapsto \widetilde{m} \ := \widetilde{P}\widetilde{M} \ . \tag{5.8}$$

$\widetilde{M}$ bzw. $\widetilde{m}$ sind die projektiven Koordinaten von $\mathcal{P}^3$ bzw. $\mathcal{P}^2$. Zur besseren Unterscheidung von den euklidischen Koordinaten sind projektive durch das Symbol $\sim$ gekennzeichnet.

Somit erhält man für einen Weltpunkt (X_w, Y_w, Z_w) bis auf einen skalaren Faktor λ den zugehörigen Bildpunkt $(x, y, 1)$ als

$$\begin{pmatrix} x' \\ y' \\ \lambda \end{pmatrix} = \widetilde{P} \cdot \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{5.9}$$

Die extrinsischen und intrinsischen Parameter der Kamera sind implizit in $\widetilde{P}$ enthalten. Zur Berechnung ist es notwendig eine Menge von Weltpunkten und die korrespondierenden Bildpunkte zu bestimmen, um dann per linearer Ausgleichsrechnung für diese Beispielpunkte $\widetilde{P}$ zu berechnen. Die Weltpunkte werden in unserem Fall durch ein quadratisches 2D-Punkteraster der Seitenlänge N - wie zum Beispiel in Abbildung 5.6 dargestellt - markiert. Hierbei ist die Position des Mittelpunktes in Weltkoordinaten und der Abstand zwischen den Mittelpunkten der Markierungen bekannt. Die Punkte liegen parallel zur xy-Ebene des Weltkoordinatensystems. In Abbildung 5.6 ist der Aufbau dargestellt. Die räumliche Tiefe der Beispielpunkte wird durch die Bewegung des Armes in z-Richtung erreicht. Ausgehend von einer Startposition (im folgenden *Kalibrierungsposition der Kamera* genannt) steuert der Roboter n Positionen an. Für jede Position werden die relativen Weltkoordinaten des Rasters berechnet und ebenso die Positionen der Rasterpunkte im Bild. Man erhält auf diese Weise $N \cdot N \cdot n$ Weltpunkte $(X_{w_i}, Y_{w_i} Z_{w_i})$ und die korrespondierenden Bildpunkte (x_i, y_i).

Aus diesen Punkten wird nun die Projektionmatrix $\widetilde{P}$ der Kamera ermittelt. Setzt man jeden Weltpunkt $(X_{w_i}, Y_{w_i} Z_{w_i})$ und die zugehörigen Bildpunkte (x_i, y_i) in das Gleichungssystem (5.9) ein und formt dieses um, so erhält man $N \cdot N \cdot n$ Gleichungen der Form:

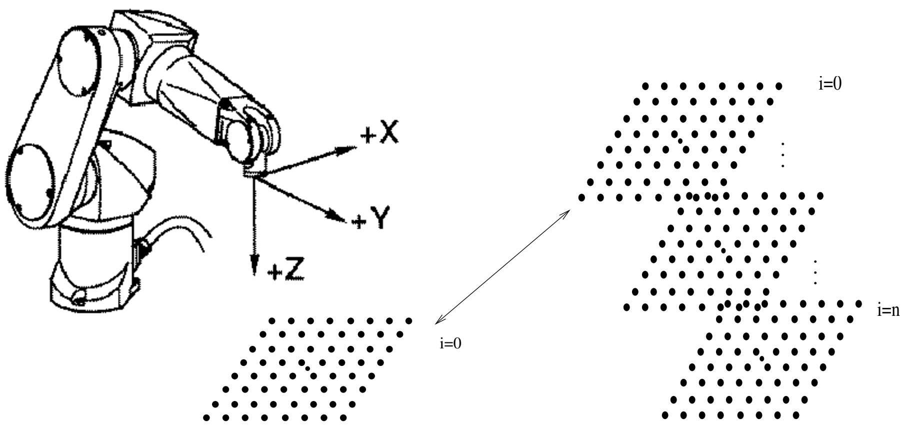


Abbildung 5.6: Kamerakalibrierung mittels 2D-Raster und Bewegung des Roboters

$$(p_{11}, p_{12}, p_{13}) \cdot \begin{pmatrix} X_{w_i} \\ Y_{w_i} \\ Z_{w_i} \end{pmatrix} - x_i \, (p_{31} p_{32} p_{33}) \cdot \begin{pmatrix} X_{w_i} \\ Y_{w_i} \\ Z_{w_i} \end{pmatrix} + p_{14} - x_i p_{34} = 0 \tag{5.10}$$

$(i = 1, \cdots, N \cdot N \cdot n)$ und ebenso Gleichungen der Form:

$$(p_{21}, p_{22}, p_{23}) \cdot \begin{pmatrix} X_{w_i} \\ Y w_i \\ Z_{w_i} \end{pmatrix} - y_i \, (p_{31} p_{32} p_{33}) \cdot \begin{pmatrix} X_{w_i} \\ Y_{w_i} \\ Z_{w_i} \end{pmatrix} + p_{24} - y_i p_{34} = 0 \tag{5.11}$$

$(i = 1, \cdots, N \cdot N \cdot n)$. Man setzt nun

$$p_{34} = 1 \; , \tag{5.12}$$

um sicherzustellen, daß $\widetilde{P}$ ungleich Null ist, da ein solches Resultat physikalisch sinnlos wäre. Ziel ist nun die Berechnung der Koeffizienten p_{ij}. Hierzu läßt sich aus den $2 \cdot N \cdot N \cdot n$ Gleichungen (5.10) bzw. (5.11) folgendes Gleichungssystem im Matrixform aufstellen:

$$\begin{pmatrix} X_{w_1} & Y_{w_1} & Z_{w_1} & 1 & 0 & 0 & 0 & 0 & -x_1 X_{w_1} & -x_1 Y_{w_1} & -x_1 Z_{w_1} \\ 0 & 0 & 0 & 0 & X_{w_1} & Y_{w_1} & Z_{w_1} & 1 & -y_1 X_{w_1} & -y_1 Y_{w_1} & -y_1 Z_{w_1} \\ X_{w_2} & Y_{w_2} & Z_{w_2} & 1 & 0 & 0 & 0 & 0 & -x_2 X_{w_2} & -x_2 Y_{w_2} & -x_2 Z_{w_2} \\ 0 & 0 & 0 & 0 & X_{w_2} & Y_{w_2} & Z_{w_2} & 1 & -y_2 X_{w_2} & -y_1 Y_{w_2} & -y_2 Z_{w_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{w_k} & Y_{w_k} & Z_{w_k} & 1 & 0 & 0 & 0 & 0 & -x_k X_{w_k} & -x_k Y_{w_k} & -x_k Z_{w_k} \\ 0 & 0 & 0 & 0 & X_{w_k} & Y_{w_k} & Z_{w_k} & 1 & -y_k X_{w_k} & -y_k Y_{w_k} & -y_k Z_{w_k} \end{pmatrix} \cdot \begin{pmatrix} P_{11} \\ \vdots \\ P_{14} \\ \vdots \\ P_{32} \\ P_{33} \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{pmatrix} \tag{5.13}$$

Hierbei ist $k = N \cdot N \cdot n$. Dieses Gleichungssystem läßt sich nun z.B. per Householderverfahren (z.B.[Stoer89](S.180ff.)) lösen. Man erhält so die gesuchte Projektionsmatrix. In unserem Fall wird die Kalibrierung für beide Kameras separat ausgeführt. Die Projektionsmatrix der Kamera i sei von nun an mit $\widetilde{P_i}$ $(i = 1, 2)$ bezeichnet. Mittels der Projektionsmatrix läßt sich jedem Bildpunkt ein *Sehstrahl* zuordnen, der alle geometrisch möglichen zugehörigen 3D-Positionen (abhängig von λ) beschreibt (siehe Abbildung 5.7).

5.1.3 Stereokalibrierung

Nachdem die Projektionsmatrizen der einzelnen Kameras ermittelt sind, bleibt die Berechnung der Koordinatentransformation von (2×2D) nach 3D. Man bezeichnet dieses als das *Zuordnungsproblem*. Abbildung 5.7 zeigt die Situation graphisch. Zu lösen ist anschaulich die Frage nach dem Schnittpunkt der Sehstrahlen der Bildpunkte m_1 und m_2.

Unter Verwendung von $\widetilde{P}_1 = (P^1 p^1)$ und $\widetilde{P}_2 = (P^2 p^2)$ läßt sich das Gleichungsystem (5.9)

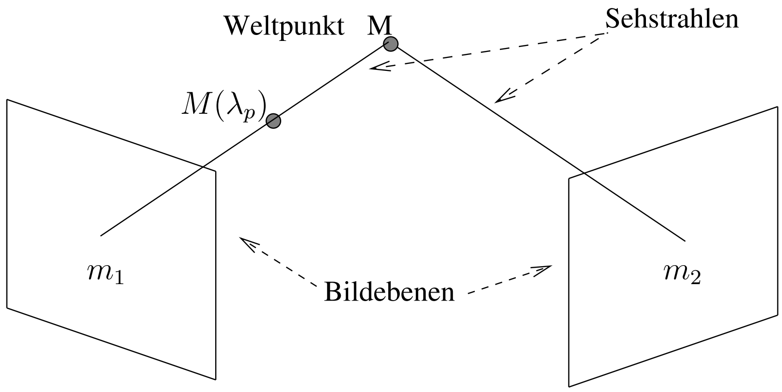


Abbildung 5.7: Zuordnungsproblem zwischen Bildpunkten m_1, m_2 und Weltpunkt M

umformen zu:

$$\begin{pmatrix} -p_1^1 \\ -p_2^1 \\ -p_3^1 \\ -p_1^2 \\ -p_2^2 \\ -p_3^2 \end{pmatrix} = \begin{pmatrix} P_{11}^1 & \cdots & P_{13}^1 & -x^1 & 0 \\ \vdots & \ddots & \vdots & -y^1 & 0 \\ P_{31}^1 & \cdots & P_{33}^1 & -1 & 0 \\ P_{11}^2 & \cdots & P_{13}^2 & 0 & -x^2 \\ \vdots & \ddots & \vdots & 0 & -y^2 \\ P_{31}^2 & \cdots & P_{33}^2 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ \lambda_1 \\ \lambda_2 \end{pmatrix} \tag{5.14}$$

Durch Einsetzen der 2D-Punkte $m_1 = (x^1, y^1)$ bzw. $m2 = (x^2, y^2)$ und der Werte der Projektionsmatrizen $\widetilde{P}_1$ und $\widetilde{P}_2$ erhält man durch Lösung dieses Gleichungssystems den gesuchten 3D- Weltkoordinatenpunkt M.

5.1.4 Veränderung der Weltkoordinaten

Aus dem Versuchsaufbau, genauer der Montage der Kameras am Roboterarm, erwächst die Notwendigkeit, die Kalibrierungsmatrix $\widetilde{P}$ der veränderten Position der Kamera anzupassen. Mit $r_{cal} = (r_{1_{cal}}, r_{2_{cal}}, r_{3_{cal}})$ sei die Position der Roboterhand zum Zeitpunkt der Kalibrierung und mit $e_{cal} = (yaw_{cal}, pitch_{cal}, roll_{cal})$ die entsprechende Rotation beschrieben. Ist nun im Anwendungsfall der Roboter (und sind damit die Kameras) in die Position $r_{new} = (r_{1_{new}}, r_{2_{new}}, r_{3_{new}})$ mit den Gelenkwinkeln $e_{new} = (yaw_{new}, pitch_{new}, roll_{new})$ gefahren, so müssen die Kalibrierungsmatrizen dieser Bewegung angepaßt werden. Im einzelnen werden eine Rotation und eine Translation durchgefürt. Die Darstellung der Gelenkwinkel ist in sogenannten *ZYZ-Eulerwinkeln* (eine umfassende Beschreibung dieses und weiterer Koordinatensysteme zur Rotationsbeschreibung findet sich in [Craig89]). Die diese Rotation beschreibende Matrix R errechnet sich als

$$R = \begin{pmatrix} \cos(\alpha)\cos(\beta)\cos(\gamma)-\sin(\alpha)\sin(\gamma) & -\cos(\alpha)\cos(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta) \\ \sin(\alpha)\cos(\beta)\cos(\gamma)+\cos(\alpha)\sin(\gamma) & -\sin(\alpha)\cos(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta) \\ -\sin(\beta)\cos(\gamma) & \sin(\beta)\sin(\gamma) & \cos(\beta) \end{pmatrix}, \tag{5.15}$$

wobei α, β und γ die Differenzwinkel zwischen e_{new} und e_{cal} (*yaw*, *pitch* und *roll* entsprechend) sind. Die Translation t erhält man als Differenz der Kalibrierungsposition und der neuen Roboterposition als

$$t = e_{new} - e_{cal} \ . \tag{5.16}$$

Setzt man nun mit den Resultaten aus den Gleichungen 5.15 und 5.16

$$\widetilde{K} = \begin{bmatrix} R & t \\ 0_3^T & 1 \end{bmatrix} , \tag{5.17}$$

so gilt für einen Weltpunkt $\widetilde{M}_{cal}$ die Beziehung

$$\widetilde{M}_{cal} = \widetilde{K}\widetilde{M}_{new} \ . \tag{5.18}$$

Aus

$$\widetilde{m} = \widetilde{P}_{cal}\widetilde{M}_{cal} \tag{5.19}$$

leitet man mit Gleichung (5.18) direkt ab

$$\widetilde{P}_{new} = \widetilde{P}_{cal}\widetilde{K} \ . \tag{5.20}$$

Durch Gleichung (5.20) ist die Auswirkung des Wechsels der Roboterposition auf die Kalibrierungsmatrix beschrieben.

5.1.5 Kalibrierungs- und Rekonstruktionsfehler

Abschließend stellt sich die Frage nach der Güte des Kalibrierungsverfahrens. Hierbei werden zwei Fehlermaße betrachtet:

- der *Kalibrierungsfehler* und
- der *Rekonstruktionsfehler*.

Unter Ersterem versteht man den Fehler, resultierend aus der Ausgleichsrechnung zur Berechnung der Projektionsmatrix $\widetilde{P}_i$ (der Kamera i). Als Rekonstruktionsfehler sei hier die Abweichung der 3D-Rekonstruktion eines Weltpunktes aus den entsprechenden, korrespondierenden Bildpunkten von der tatsächlichen 3D-Position bezeichnet.

Kalibrierungsfehler

Die Kalibrierungsmatrix $\widetilde{P}$, wie in Abschnitt 5.1.2 berechnet, ist eine Approximation der tatsächlichen, im allgemeinen nicht-linearen (z.B. Verzerrung Abschnitt 5.1.1) projektiven Abbildung. Als Test des Fehlers betrachten wir die Rückprojektion der Weltpunkte des Projektionsrasters in die Bildebene mittels $\widetilde{P}$. Hier ist für einen Weltpunkt

$$w^{(i,j)} = (x_w^{(i,j)}, y_w^{(i,j)}, z_w^{(i,j)})$$

des Kalibrierungsrasters (an der Position (i,j))

$$\begin{aligned} \widetilde{w}_p^{(i,j)} &= \widetilde{P} \cdot \begin{bmatrix} w^{(i,j)} \\ 1 \end{bmatrix} && (5.21) \\ &= (\widetilde{x}_p^{(i,j)}, \widetilde{y}_p^{(i,j)}, \lambda) && (5.22) \end{aligned}$$

Als Fehler wird nun der euklidische Abstand zwischen dem Bildpunkt

$$w_p^{(i,j)} = (x_p^{(i,j)}, y_p^{(i,j)})$$

und dem tatsächlich zugehörigen Bildpunkt des Kalibrierungsrasters an der Stelle (i,j)

$$w_k^{(i,j)} = (x_k^{(i,j)}, y_k^{(i,j)})$$

betrachtet. Summiert über das gesamte Kalibrierungsraster ergibt sich für den Kalibrierungsfehler einer Schicht m

$$E_m = \sum_{i=0}^{N} \sum_{j=0}^{N} \sqrt{(x_p^{(i,j)} - x_k^{(i,j)})^2 + (y_p^{(i,j)} - y_k^{(i,j)})^2}, \tag{5.23}$$

wobei N die Seitenlänge des Kalibrierungsrasters ist. Für den vollständigen Kalibrierungsquader mit n Schichten berechnet sich der Fehler als

$$E_{cal} = \sum_{j=0}^{n} E_j. \tag{5.24}$$

Mit E_{cal}^i sei nun der Kalibrierungsfehler der Kamera i bezeichnet.

Rekonstruktionsfehler

Der Rekonstruktionsfehler ist ein Qualitätsmaß für die Stereokalibrierung. Es werden die Punkte des Kalibrierungsrasters im Stereobild detektiert und mittels des Vorgehens aus Abschnitt 5.1.3 daraus die 3D-Positionen berechnet. Diese werden dann mit den tatsächlichen 3D-Koordinaten verglichen. Für ein quadratisches Kalibrierungsraster der Seitenlänge N sei mit

$$E_m = \sum_{i=0}^{N} \sum_{j=0}^{N} \sqrt{(x_p^{(i,j)} - x_k^{(i,j)})^2 + (y_p^{(i,j)} - y_k^{(i,j)})^2 + (z_p^{(i,j)} - z_k^{(i,j)})^2}\ , \tag{5.25}$$

der durchschnittliche Fehler einer Schicht m gegeben. Folglich berechnet sich der Gesamtfehler eines n-schichtigen Kalibrierungsquaders durch

$$E_{rek} = \sum_{j=0}^{n} E_j. \tag{5.26}$$

Mit E_{rek} ist nun der Rekonstruktionsfehler bezeichnet.

Ein Beispiel

In diesem Abschnitt ist ein Kalibrierungsbeispiel beschrieben. Hierbei werden die Daten des Bildenzerrungs-Beispieles aus Abschnitt 5.1.1 verwendet. Der Kalibrierungs- und der Rekonstruktionsfehler sind sowohl graphisch als auch numerisch dargestellt. Es werden eine Kalibrierung mit vorheriger Bildentzerrung und eine ohne verglichen.

Bei der Betrachtung des Kalibrierungsfehlers wird aus Übersichtsgründen jeweils nur Kamera 1 betrachtet, da bei beiden Kameras tendenziell derselbe Effekt beobachtet wurde. Abbildung 5.8 zeigt spaltenweise die Rückprojizierten Kalibrierungspunkte ohne (links) und mit (rechts) vorheriger Entzerrung der Kamerabilder. Es sind repräsentativ die erste ($n = 0$) eine mittlere ($n = 4$) und letzte ($n = 7$) Schicht des Kalibrierungsquaders ausgewählt. Die schwarzen Punkte sind die Bilder der Punkte des Kalibrierungsrasters, grau sind die rückprojizierten Weltpunkte (des Rasters). Man erkennt unmittelbar die Auswirkung (Verzerrungen) und damit Notwendigkeit der Bildentzerrung, um brauchbare Ergebnisse zu erzielen.

In Tabelle 5.2 sind die Fehler (durchnittliche Abweichung in Pixeln, Minimal- und Maximalwert) für alle Schichten ($n = 0, ..., 7$) dargestellt. Die Fehler sind in den mittleren Schichten am geringsten. Grund hierfür sind zwei Effekte: zum einen die Anordnung als Gitter, zum anderen ein mit wachsender Schicht zunehmender Fehler bei der Rekonstruktion der Schwerpunkte der Rasterpunkte im Bild.

	ohne Entzerrung			mit Entzerrung		
Schicht	durchnittlicher Fehler	minimaler Fehler	maximaler Fehler	durchnittlicher Fehler	minimaler Fehler	maximaler Fehler
0	18.381	1.454	46.954	1.686	0.0726	3.209
1	13.471	1.067	38.587	1.316	0.0436	2.843
2	9.788	1.161	31.441	1.031	0.0019	3.116
3	8.243	1.099	22.733	0.847	0.0179	3.158
4	8.687	0.122	31.334	0.783	0.0575	3.259
5	11.237	0.127	43.269	0.860	0.0919	3.275
6	15.519	0.952	56.268	0.975	0.0367	4.267
7	20.646	0.695	69.973	1.241	0.1033	3.891

Tabelle 5.2: Fehler der Kalibrierung in Pixeln

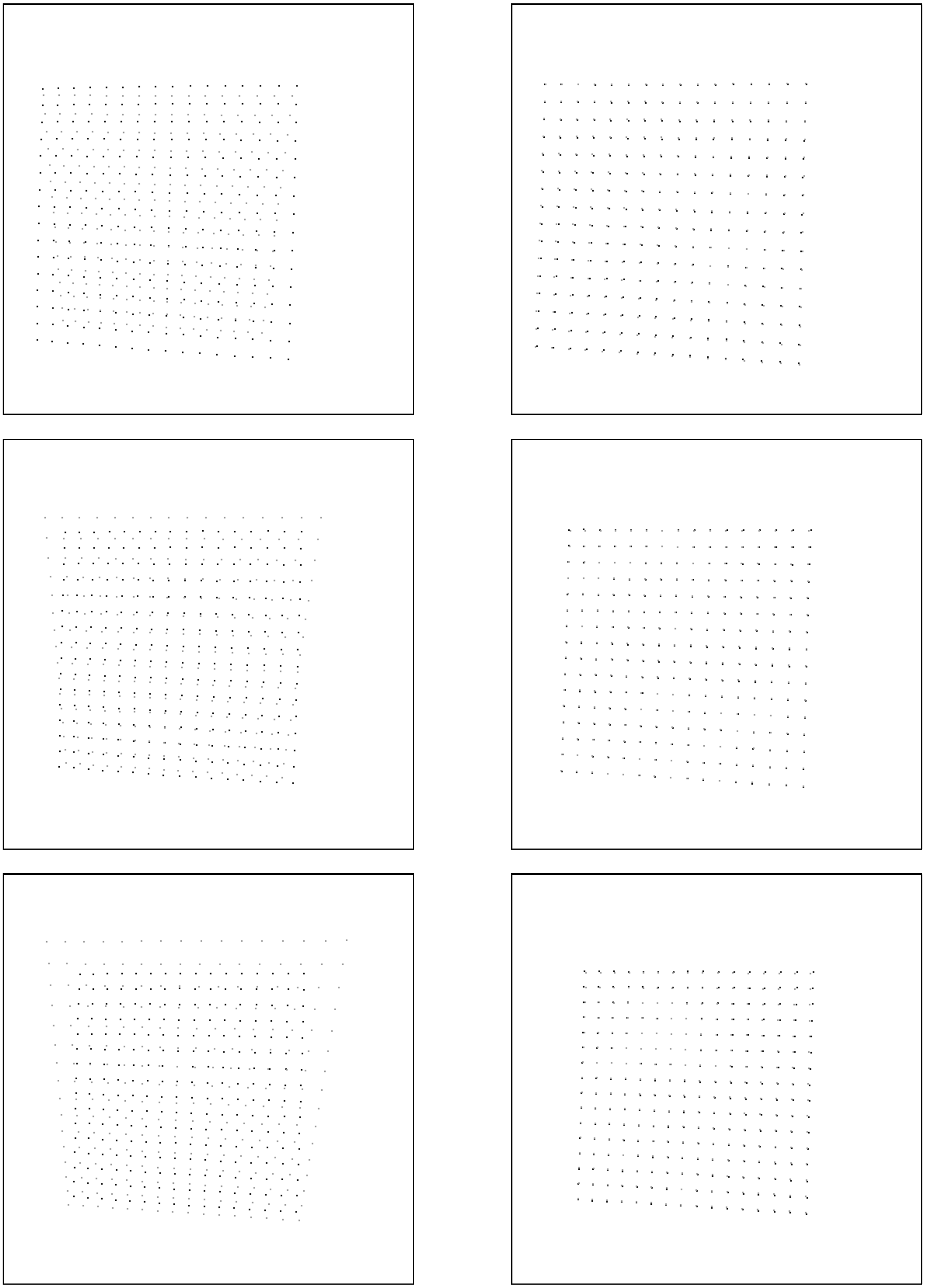


Abbildung 5.8: Vergleich der Rückprojektionen der Punkte des Kalibrierungsquaders (Schichten n=0,4,7) ohne (links) und mit (rechts) vorheriger Bildentzerrung

5.2 Stereomatching

In den vorangegangenen Abschnitten wurde gezeigt, wie mit den Mitteln der projektiven Geometrie korrespondierenden Bildpunkten ein Weltpunkt zugeordnet werden kann. Dieser Abschnitt nun beschäftigt sich mit dem *Stereomatching*, der Ermittlung dieser 2D-2D-Korrespondenzen.

In dieser Arbeit wird das Stereomatching zweier Bilder als zweistufiger Vorgang aufgefaßt:

- Vorverarbeitung / Merkmalsextraktion
- Matching.

Der Schritt *Vorverarbeitung / Merkmalsextraktion* dient der Reduktion der möglichen Entsprechungen. Es werden für die gesuchten Korrespondenzen Punkte vorausgewählt. In unserem Fall sollen nicht die Stereobilder als ganzes (maximal) gematched werden, sondern nur spezielle Bildelemente (*feature matching*), namentlich die "Kabel". Dieser Schritt der Vorverarbeitung ist bereits im vorangegangen Kapitel geleistet worden; Ergebnis der Kabelerkennung ist ein Binärbild.

Ziel des *Matchings* ist es, Paare von Bildpunkten zu finden, die einander entsprechen, Bild desselben Weltpunktes sind. Grundlage ist die *Epipolare Geometrie* eines Stereokamerasystems, dargestellt in Abbildung 5.9. Im Falle dieser Arbeit sind Korrespondenzen zwischen Binärbildern gesucht. In Abschnitt 5.2.1 wird die Berechnung einer *Epipolaren* bzgl. eines Bildpunktes erläutert. Abschnitt 5.2.2 zeigt das Vorgehen zur Ermittlung der Korrespondenzen aus algorithmischer Sicht.

5.2.1 Die epipolare Geometrie

Man betrachte einen Punkt m_1 im linken (Abbildung 5.9) Kamerabild, welcher Projektion eines Weltpunktes M (euklidische Koordinaten) ist, Dann werden alle Punkte der durch das optische Zentrum C_1 und M definierten Graden auf m_1 projiziert. Das Bild dieser Geraden in der rechten Bildebene R_2 wird *Epipolare* genannt und ist eine Gerade. Man betrachtet nun den Punkt im Unendlichen auf der Geraden durch C_1 und M $\widetilde{D}$, da M a-priori unbekannt ist. Es gilt dann

$$\widetilde{m}_1 = \widetilde{P}_1 \widetilde{D} = (P_1 \widetilde{p}) \begin{pmatrix} D \\ 0 \end{pmatrix} \tag{5.27}$$

Der Schnitt des optischen Stahles mit der Ebene im Unendlichen D berechnet sich als

$$D = P_1^{-1} \widetilde{m}_1. \tag{5.28}$$

Der Epipol $\widetilde{e}_2$ (Projektion des optischen Zentrums $\widetilde{C}_1$ in die linke Bildebene R_2) und die Projektion von $\widetilde{D}$ auf R_2 legen die Epipolare eindeutig fest. Man hat nun durch das

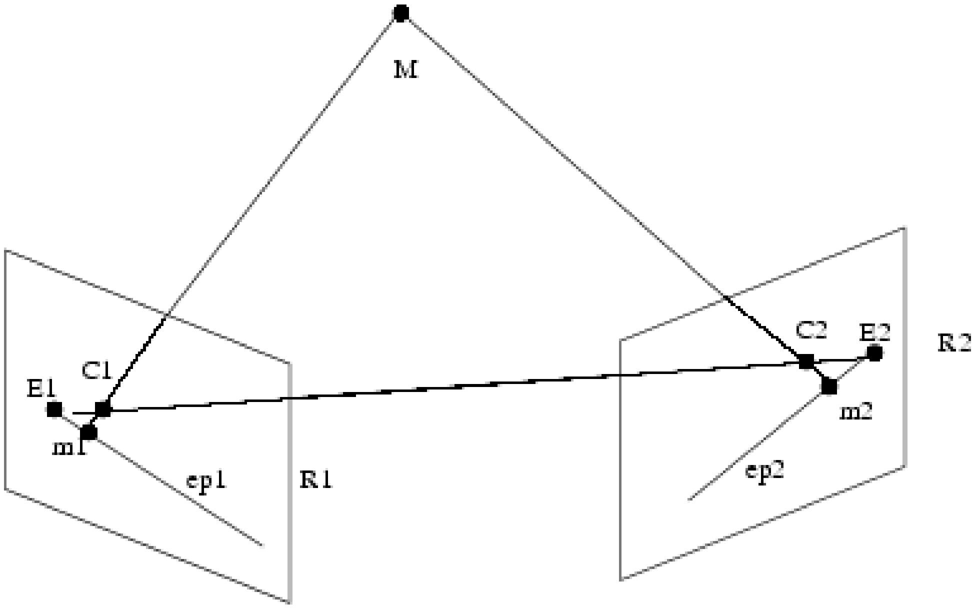


Abbildung 5.9: Die epipolare Geometrie

Kreuzprodukt $\widetilde{e}_2 \times \widetilde{d}_2$ eine projektive Darstellung der Epipolaren ep_2. Als Punktgleichung hat man also

$$\widetilde{m}_2^T(\widetilde{e}_2 \times \widetilde{d}_2) = 0 \ . \tag{5.29}$$

Das Kreuzprodukt $\widetilde{e}_2 \times \widetilde{d}_2$ läßt sich leicht als Matrix-Vektor-Multiplikation der Form $F\widetilde{m}_2$ schreiben. Setzt man für $\widetilde{e}_2 = (e_1, e_2, e_3, 1)^T$ die Matrix

$$\widetilde{E}_2 = \begin{pmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{pmatrix},$$

so gilt für alle Vektoren $v : \widetilde{E}_2 v = \widetilde{e}_2 \times \widetilde{m}_2$, d.h. $\widetilde{E}_2$ ist die antisymetrische Matrix, die das Kreuzprodukt mit dem Vektor $\widetilde{e}_2$ beschreibt. Man erhält somit

$$F = \widetilde{E}_2 P_2 P_1^{-1} \widetilde{m}_1 \ . \tag{5.30}$$

Man sieht, daß die Beziehung zwischen einem Punkt und der zugehörigen Epipolaren linear in projektiven Koordinaten ist. Man erhält insgesamt als Beziehung zwischen $\widetilde{m}_1$ und einem beliebigen Punkt $\widetilde{m}_2$ auf der Epipolaren ep_2 die Gleichung

$$\widetilde{m}_2^T F \widetilde{m}_1 = 0. \tag{5.31}$$

Man nennt diese die *Longuet-Higgens-Gleichung*. Sie charakterisiert ep_2 eindeutig. Es sei bemerkt, daß die *Fundamental-Matrix* F invariant gegen Rotationen und Translationen des

Gesamt-Stereosytems ist, da sie das geometrische Verhältnis der beiden Kameras zueinander beschreibt; dieser Aspekt ist für diese Arbeit aufgrund der Montage der Kameras am bewegten Roboterarm von Bedeutung. So wird sie nur einmal nach Gleichung 5.30 berechnet und kann von da an eingesetzt werden. Zur Laufzeit des Systems werden nur noch mittels Gleichung (5.31) die benötigten Epipolaren berechnet.

Mit $epi(p)$ sei die zu Bildpunkt p gehörige Epipolare bezeichnet.

5.2.2 Algorithmus zur Bestimmung der Stereokorrespondenzen

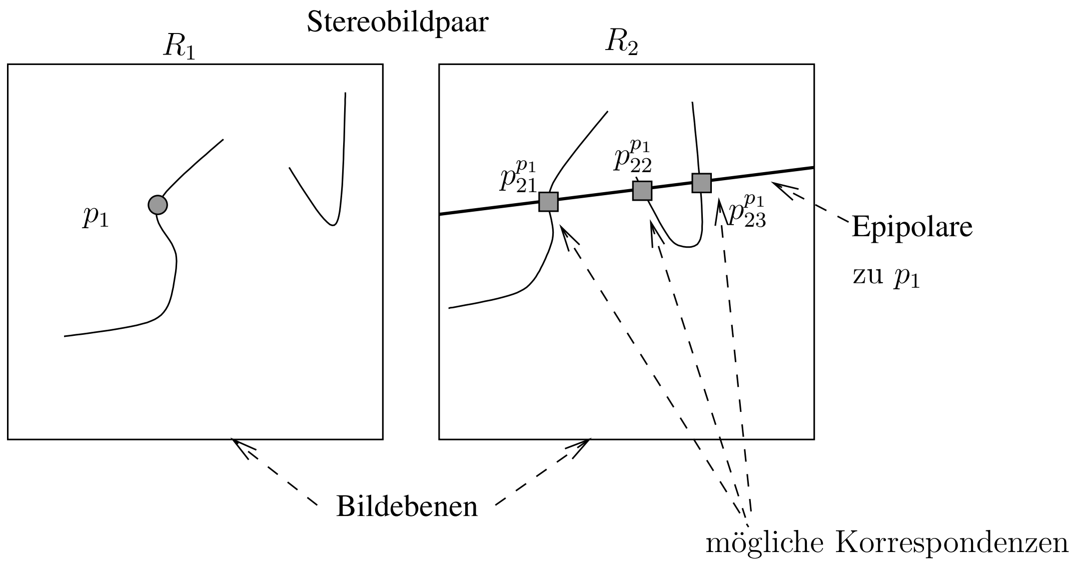


Abbildung 5.10: Mehrdeutigkeit der Korrespondenzensuche mittels der Epipolaren

Durch die Epipolare Geometrie ist ein Werkzeug gegeben zu jedem Bildpunkt p_1 in Bild R_1 eine Menge $cand(epi(p_i))$ diesem entsprechender Bildpunkte $p_{21}^{p_1}, ..., p_{2m}^{p_1}$ in R_2 zu ermitteln. Abbildung 5.10 zeigt ein Beispiel hierfür.

Gesucht ist aber eine *eindeutige* Zuordnung. Hierzu wird für jeden Bildpunkt $p_{2i}^{p_1}$ mit $i \in \{1, ..., n\}$ die zugehörige Epipolare in R_1 berechnet und jeweils der euklidischen Abstand zwischen der Geraden $epi(p_{2i}^{p_1})$ und p_1 ermittelt. Gilt

$$d_{min} := \min\{dist(epi(p_{2i}^{p_1}), p_1) | i \in \{1, ..., m\}\} > \epsilon \tag{5.32}$$

für ein gegebenes $\epsilon \in I\!R$, so werden die Paare $(p_{2i}^{p_1}, p_1)$ mit $dist(epi(p_{2i}^{p_1}), p_1) = d_{min}$ als Stereokorrespondenzen akzeptiert und mittels Gleichung (5.14) in 3D-Koordinaten umgerechnet. Ergebnis ist eine Menge von 3D-Positionen. In Abbildung 5.11 ist dieses graphisch veranschaulicht. Durch die Verwendung der Minimumbildung ist Eindeutigkeit nicht immer gegeben. Auch Situationen, in denen der korrespondierende Punkt gar nicht im Binärbild liegt, sorgen für Fehlmatches. Diese werden aber in Kauf genommen, da sie zahlenmäßig

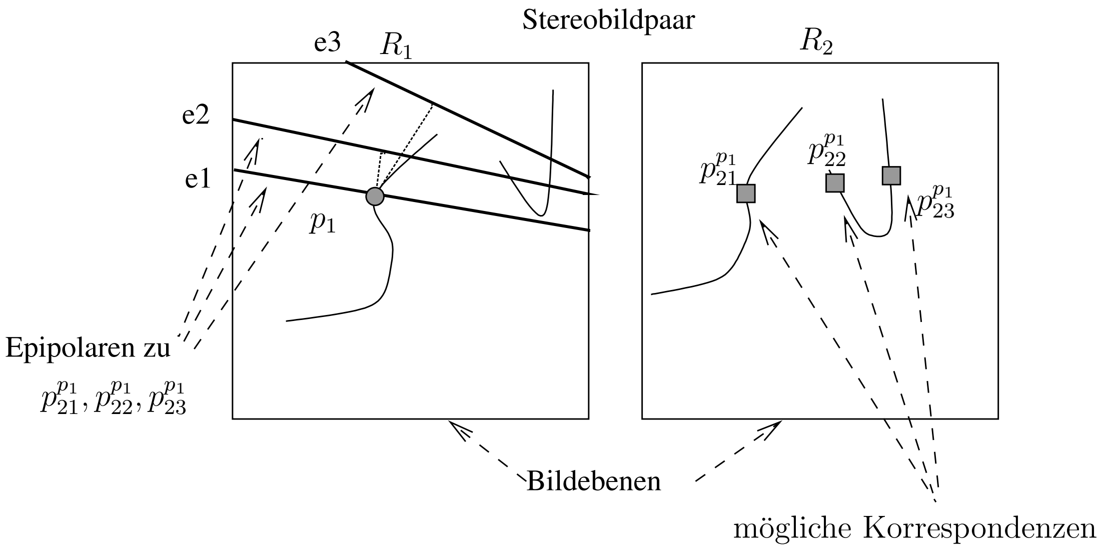


Abbildung 5.11: Auswahl aus Kandidaten mittels Epipolaren

gering sind und über eine größere Anzahl von Stereobildern in 3D-Koordinaten durch einfache Schwellwertbildung aus dem 3D-Gitter zu entfernen sind.

Mittels des hier vorgestellten Vorgehens können nun zu einem Stereobildpaar die $2D-2D$ Korrespondenzen berechnet und in 3D-Koordinaten überführt werden. Abschnitt 5.3 liefert ein Beispiel hierfür. Die dort ermittelten 3D-Daten werden in der Fortsetzung in Abschnitt 6.3 zur Rekonstruktion der Kabel verwendet.

Im folgenden Kapitel 6 wird nun eine Methode vorgestellt, aus diesen Punkten, eine Objektrekonstruktion zu erhalten.

5.3 Beispiel (Teil I) : Stereokorrespondenzen

In diesem Abschnitt wird ein Beispiel für das Auffinden der Stereokorrespondenzen gegeben. Abbildung 5.12 zeigt den Versuchsaufbau, die drei Perspektiven des Kamerasystems. Der Roboter fährt 3 Positionen, wie in der Abbildung angedeutet, an. Es wird jeweils ein Stereobildpaar aufgenommen. In den Abbildung 5.13 - 5.15 sind die gefundenen Korrespondenzen eingezeichnet (Farbe: Magenta). Die Stereobildpaare sind übereinander gesetzt. Die Binärbilder der Kabelverläufe sind im Druckbild leider schwer zu erkennen. Aus Gründen der Übersicht der Darstellung ist auf eine paarweise Kennzeichnung der Korrespondenzpaare verzichtet. Die Darstellung der so gewonnenen 3D-Positionen findet man in Abschnitt 6.3.

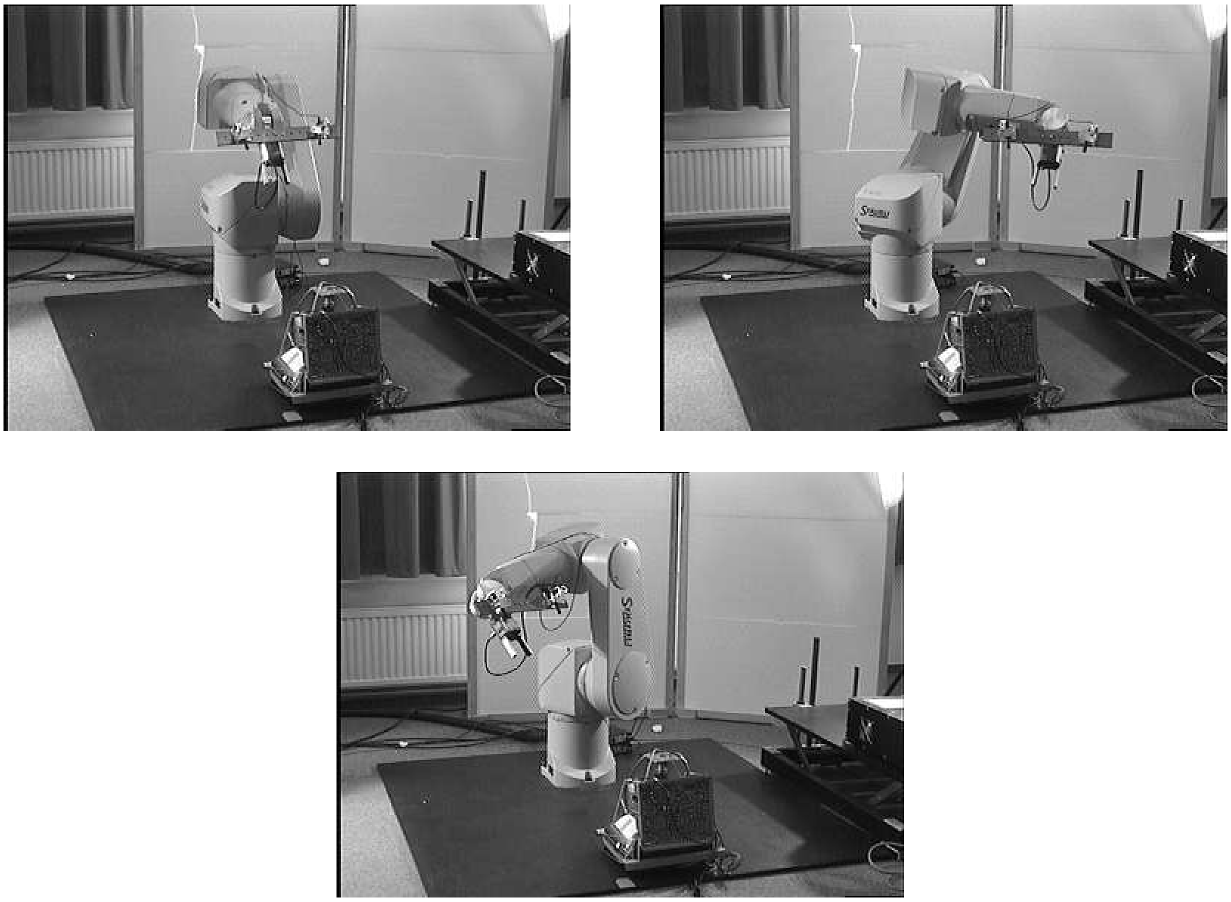


Abbildung 5.12: Versuchsaufbau und Sichten des Rekonstruktionsbeispiels

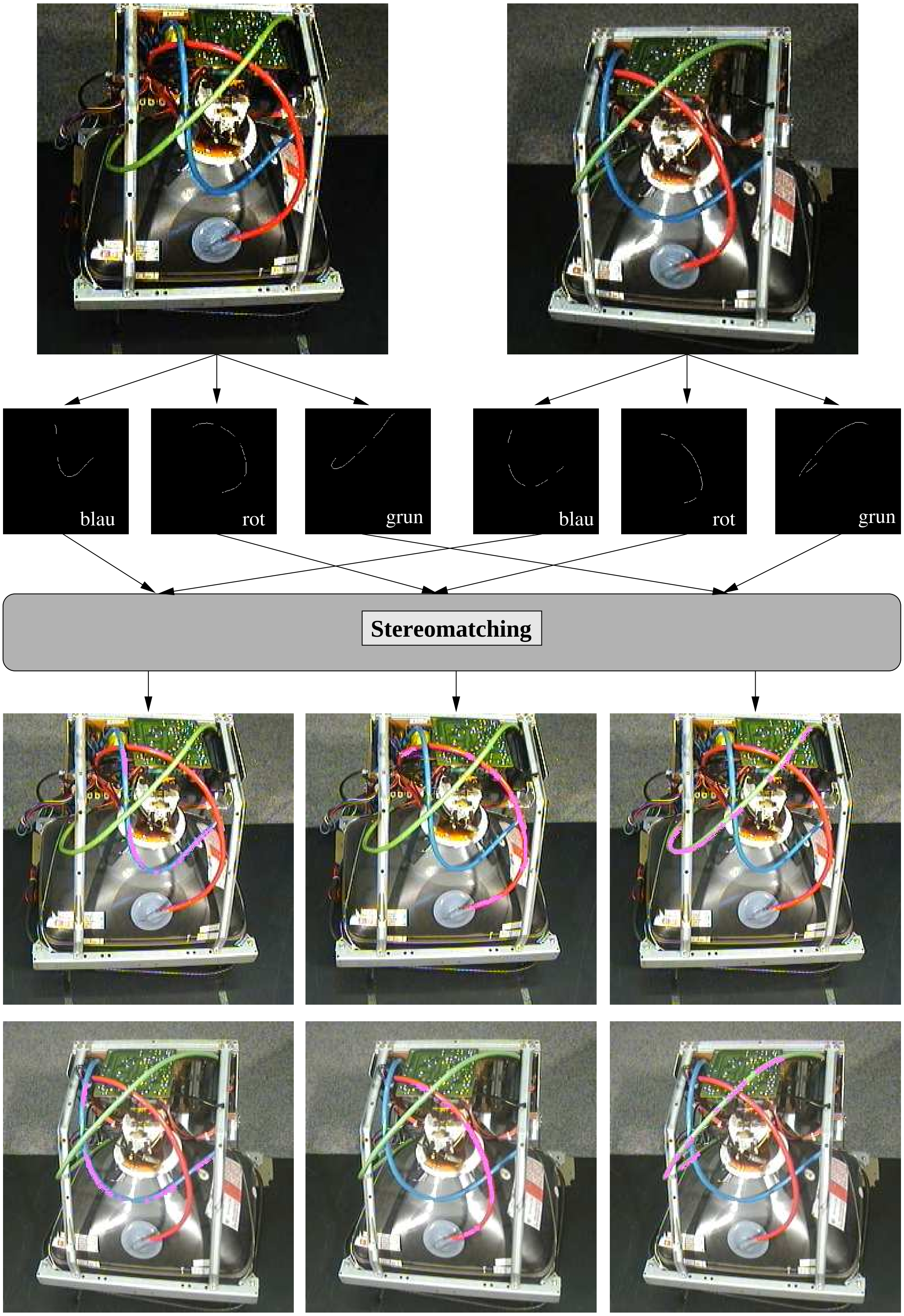


Abbildung 5.13: Beispiel Rekonstruktion Teil I, Sicht 1

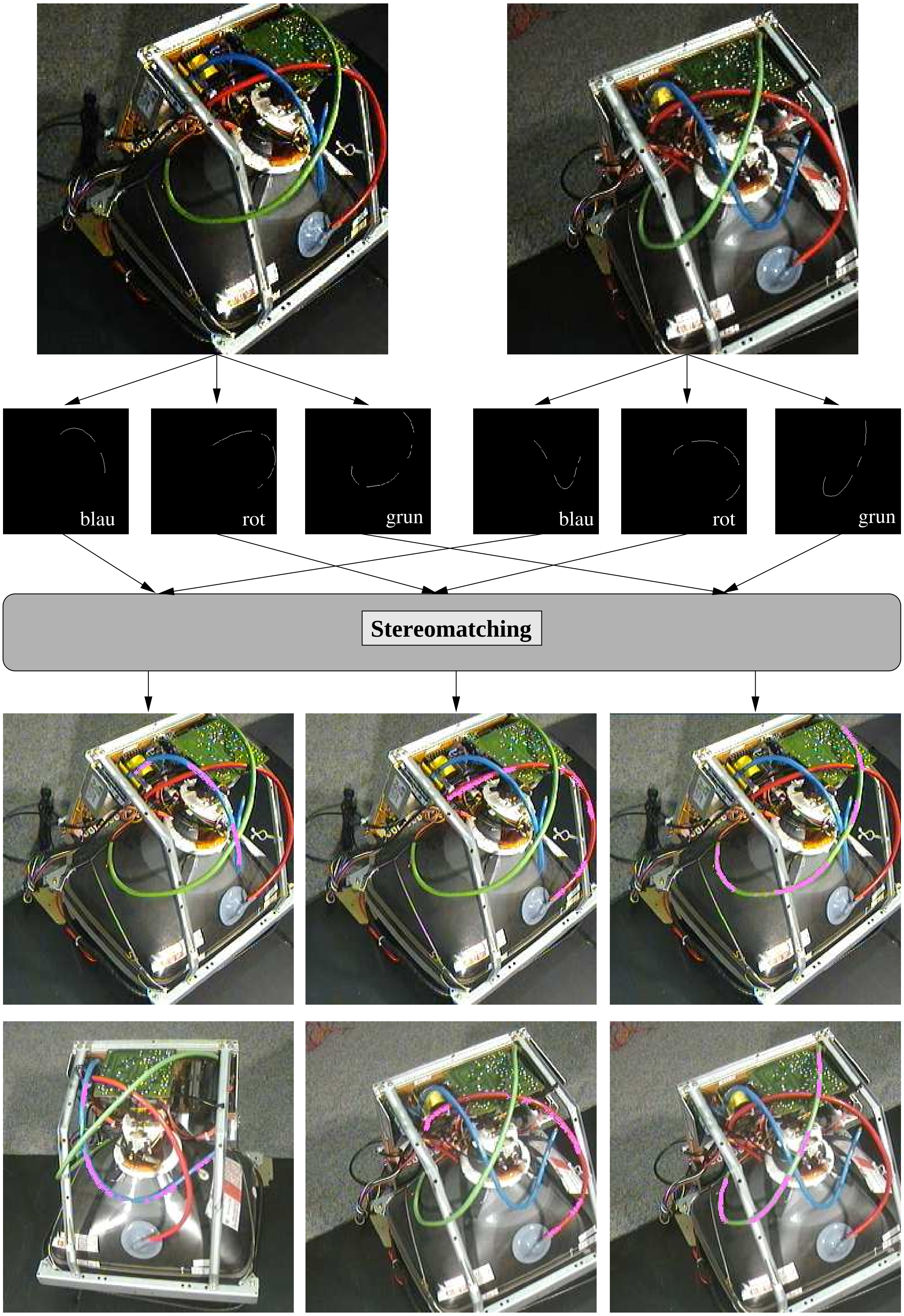


Abbildung 5.14: Beispiel Rekonstruktion Teil I, Sicht 2

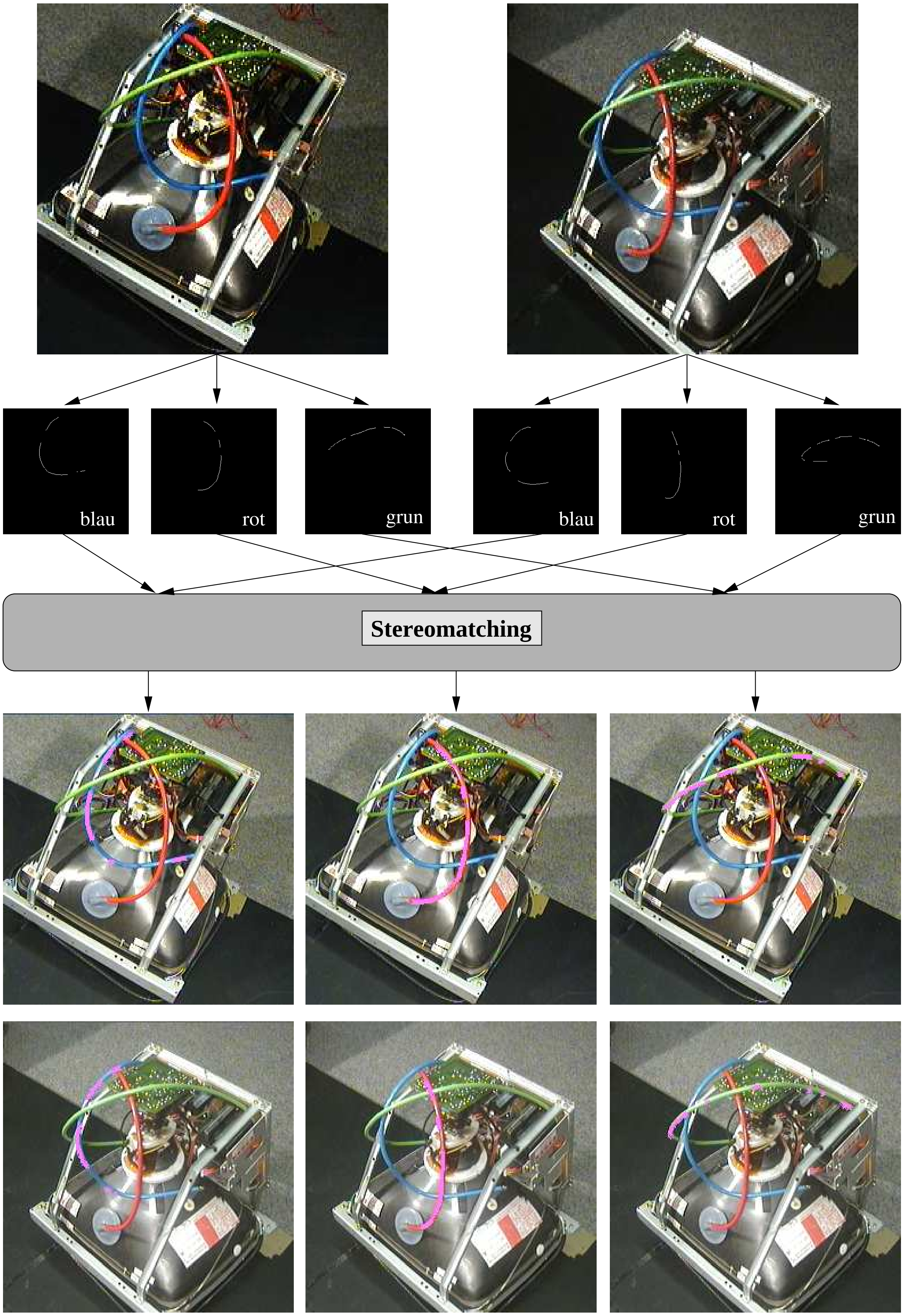


Abbildung 5.15: Beispiel Teil I, Stereomatching ,Sicht 3

Kapitel 6

3D Besetztheitsgitter zur Objektrekonstruktion

Im folgenden Kapitel werden 3D-Besetztheitsgitter als Werkzeug zur Objektrekonstruktion vorgestellt. Ihre Vorteile liegen in

- der Unterabtastung des Raumes und
- der Integration der Information über die Zeit,

so daß ungenaue “Sensordaten” (hier Bildinformation) “örtlich unscharf” gesammelt werden. Abschnitt 6.1 erläutert die Grundidee und bisherige Einsatzgebiete von 2D-Gittern, während in Abschnitt 6.2 der Übergang zu 3D-Gittern motiviert und deren Definition gegeben wird. Zum Abschluß des Kapitels wird mit der Fortsetzung des Beispiels aus Abschnitt 5.3 eine Anwendung der Gitter demonstriert.

6.1 Die Idee des Besetztheitsgitters

Die Grundidee der Besetztheitsgitter stammt bereits aus dem Jahre 1983. Sie wurden am *Mobile Robot Labortory* der *Carnegie Mellon University of Pittsburgh* entwickelt. Man suchte eine Struktur, welche es erlaubt ungenaue Daten von Sonarsensoren zu kombinieren, und daraus ein ungefähres Modell der Umgebung zu gewinnen. Einsatzgebiet war und ist die Navigation mobiler Roboter. Als grundlegende Arbeiten hierzu sind z.B. [Moravec-Elfes85] und [Elfes87] zu nennen. Eine abschließenden Überblick über den aktuellen Stand liefern u.a. [Martin-Moravec96] und [Thrun96]. Kernpunkte dieser Arbeiten sind die Integration von Sensordaten und die Gewinnung eines Sensormodelles. Jede Zelle des Gitters repräsentiert einen Bereich der Umgebung. Je nach Grösse zwischen 10 cm und bis zu 2 m. Eine feinere Unterteilung ist nicht sinnvoll, da Sonarmessungen räumlich nur sehr ungenaue

Ergebnisse liefern. Die Gitter dienen der Integration dieser Messungen. Jede Zelle enthält einen Wert für die Besetztheit der zugehörigen Region.

Die in den oben genannten Arbeiten vorgestellten Ideen lassen sich auch auf 3-dimensionale Gitter übertragen.

6.2 3D Besetztheitsgitter

Die Idee 3-dimensionaler Besetztheitsgitter wird erstmals von Hans Moravec eingeführt [Moravec96] . Ist es für den Einsatz von Sonarsensoren, welche üblicherweise auf einer Ebene montiert sind, ausreichend, die Daten zweidimensional zu sammeln, so bietet der Einsatz von Kamerasystemen die Möglichkeit, Objektinformationen für beliebige Umgebungspunkte zu erlangen. Es entsteht ein räumliches Besetztheitsmodell der Umgebung. In [Moravec96] wird beschrieben, wie ein Raum mittels eines mobilen Roboters und eines montierten Stereokamerasystems erfaßt wird. Hierzu werden in den Stereobildern mit einem *Interest-Operator* Bildbereiche ausgewählt, die mit großer Wahrscheinlichkeit auch im anderen Stereobild mittels Korrelationsverfahren ermittelt werden. Die so gefundenen Stereopaare liefern eine Tiefeninformation, welche in das Gitter eingetragen wird. Zur genauen Beschreibung von Interest-Operator, Sensormodell und Korrelationsverfahren sei auf den Artikel verwiesen. Ein Problem der 3D-Gitter ist ihr immenser Speicherbedarf.

Im Unterschied zu diesem Ansatz ist Ziel dieser Arbeit nicht die Exploration des Raumes, sondern das Rekonstruieren von Objekten innerhalb eines eingeschränkten Arbeitsbereiches. Wir verwenden daher von der absoluten Ausdehnung kleinere Gitter mit höherer Auflösung.

6.2.1 Der Aufbau eines 3D-Gitters

Als Arbeitsbereich ist ein Würfel von $x_d \times y_d \times z_d\ cm^3$ in Weltkoordinaten vorgegeben. Der Ursprung liegt bei (x_0, y_0, z_0). Abbildung 6.1(a) zeigt einen solchen Arbeitsbereich (für unsere Anwendung in typischer Weise auf einem Drehteller positioniert).

Dieser Bereich wird nun durch ein 3-dimensionales Gitter abgetastet. Die Anzahl der Zellen in x-, y- und z-Richtung sei identisch auf $n+1 \in I\!N$ mit n gerade gesetzt. Die Auflösung d des Gitters sei definiert durch

$$d = \frac{x_d}{n} = \frac{y_d}{n} = \frac{z_d}{n}\ . \tag{6.1}$$

Eine Gitterzelle soll einen Bereich des Raumes repräsentieren. Dieses soll auch für echte Randpunkte des Arbeitsbereiches gelten. Das Gitter wird folglich so gesetzt, daß der Arbeitsbereich um eine halbe Zellenbreite überlappend abgedeckt wird. Abbildung 6.1(b) zeigt dieses Vorgehen exemplarisch für den in Abbildung 6.1(a) gewählten Arbeitsbereich.

Ein würfelartiges 3D-Gitter ist also ein Sechstupel $\mathcal{G} = (n, (x_0, y_0, z_0), d, \mathrm{pos}, \mathrm{depos}, G)$, wobei $n \in I\!N$, $(x_0, y_0, z_0) \in I\!R^3$, $d \in I\!R$. Durch die Funktion G sei der Zugriff auf die Gitterzellen möglich. Mit $G(i, j, k)$ sei also der Zugriff auf den Inhalt der (i, j, k)-ten Zelle bezeichnet. Die Abbildung *pos*, definiert durch

$$\mathrm{pos} : I\!R^3 \to \{0, ..., n\}^3 \tag{6.2}$$

ordnet jedem Weltpunkt eine Gitterzelle zu, wobei als Ursprung des Gitters die Zelle gesetzt ist, die (x_0, y_0, z_0) enthält. Die "Umkehrabbildung" *depos* dagegen ordnet jedem Gitterpunkt per Definition das Zentrum der Gitterzelle als Weltpunkt zu; in Formeln also:

$$\mathrm{Depots} : \{0, ..., n\}^3 \to I\!R^3 \ , \tag{6.3}$$

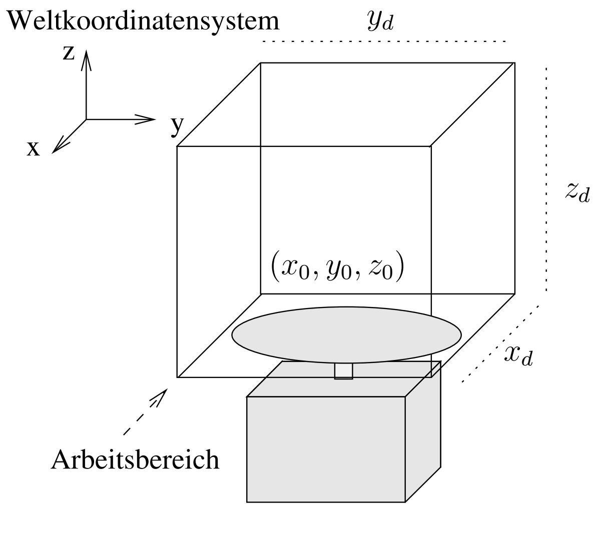
mit

$$\mathrm{Depots}(i, j, k) = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \begin{pmatrix} i \cdot d \\ j \cdot d \\ k \cdot d \end{pmatrix} \ . \tag{6.4}$$

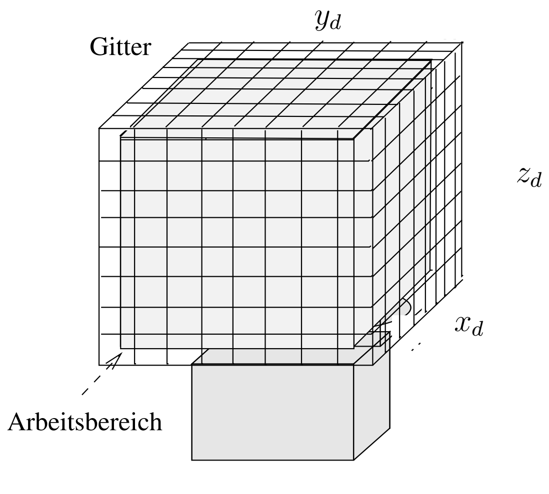
Nachdem Aufbau und Lage des Gitters definiert sind, wird nun das Eintragen von Information in das Gitter, der Update-Operator erläutert.

6.2.2 Update-Operation

Sei durch $\mathcal{G} = (n, (x_0, y_0, z_0), d, \mathrm{pos}, \mathrm{depos}, G)$ ein Gitter gegeben. Ferner sei $M = \{m_1, ..., m_z\}$ eine Menge von Weltpunkten, also $m_i \in I\!R^3$. Dann ist *update*$(\mathcal{G}, \mathcal{M})$ "punktweise" durch



(a)



(b)

Abbildung 6.1: Arbeitsbereich (a) und relative Lage des 3D-Gitters dazu (b) die Zuweisungsoperation

$$G(\text{pos}(m_i)) := G(\text{pos}(m_i)) + 1 \tag{6.5}$$

für $i = 1, ...z$ definiert. Hier wird Information akkumuliert. Die Werte der Zellen werden inkrementell erhöht, so daß insbesondere bei der Addition von Rekonstruktionsergebnissen mehrerer Sichten ein höherer Eintrag eine größere Wahrscheinlichkeit der Objektzugehörigkeit ausdrückt. Eine Zelle votiert mit ihrem Wert mehr oder weniger für die Zugehörigkeit der zugeordneten Region zum Objekt.

Bei Positionierung des Gitters auf einem Drehteller ist es zusätzlich noch notwendig, mit *pos* und *depos* berechnete Koordinaten entsprechend der Drehung des Tellers zu rotieren.

Im folgenden Abschnitt 6.3 sind die Ergebnisse der Rekonstruktion des Beispiels aus Abschnitt 5.3 in ein Gitter eingetragen.

6.3 Beispiel (Teil II) : Rekonstruktion

Die in Abschnitt 5.3 gewonnenen Rekonstruktionsdaten werden in diesem Abschnitt nun in ein 3D-Gitter der Dimension $100 \times 100 \times 100$ eingetragen. Das Gitter ist parallel zum Roboterkoordinatensystem, den Monitor einschließend angeordnet. In den Abbildungen 6.2 - 6.4 sind der Reihe nach die Rekonstruktionen des blauen, roten und grünen Kabels zu sehen. Die ersten drei Zeilen sind jeweils die Ergebnisse der drei Sichten, dargestellt aus jeweils drei Perspektiven. Die vierte und letzte Zeile zeigt die Addition der drei Sichten in ein Gitter.

Man sieht an den Ergebnissen, daß die Kabel in den einzelnen Bildern dem Verlauf nach richtig rekonstruiert sind, eine Abweichung bzgl. der Rekonstruktion zwischen den Sichten (Parallelverschiebung) aber erkennbar ist. Entscheidend ist aber auch hier, daß der Abstand im Rahmen der Toleranz (gegeben durch Greifer) liegt.

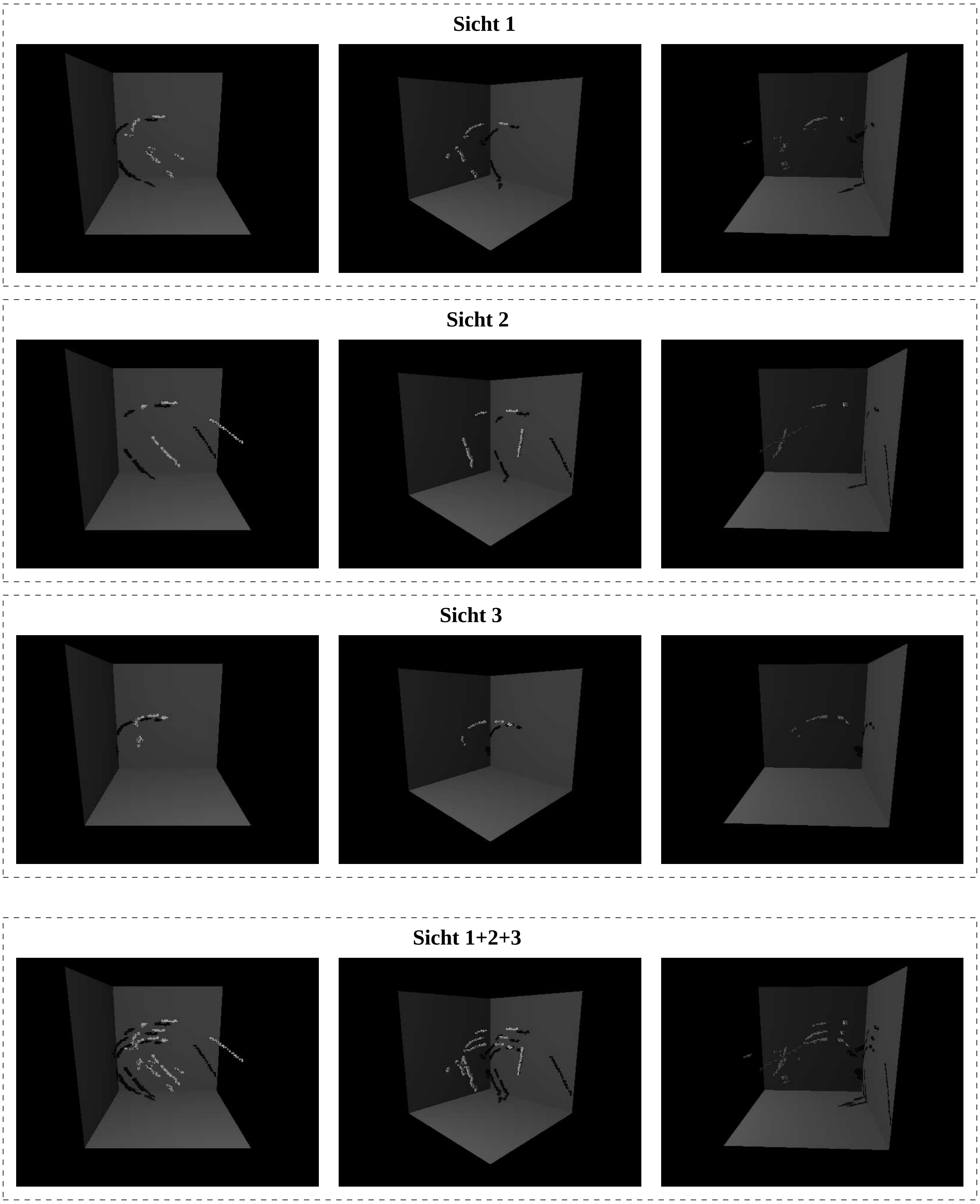


Abbildung 6.2: Beispiel Rekonstruktion Teil II, Sichten 1-3, blaues Kabel

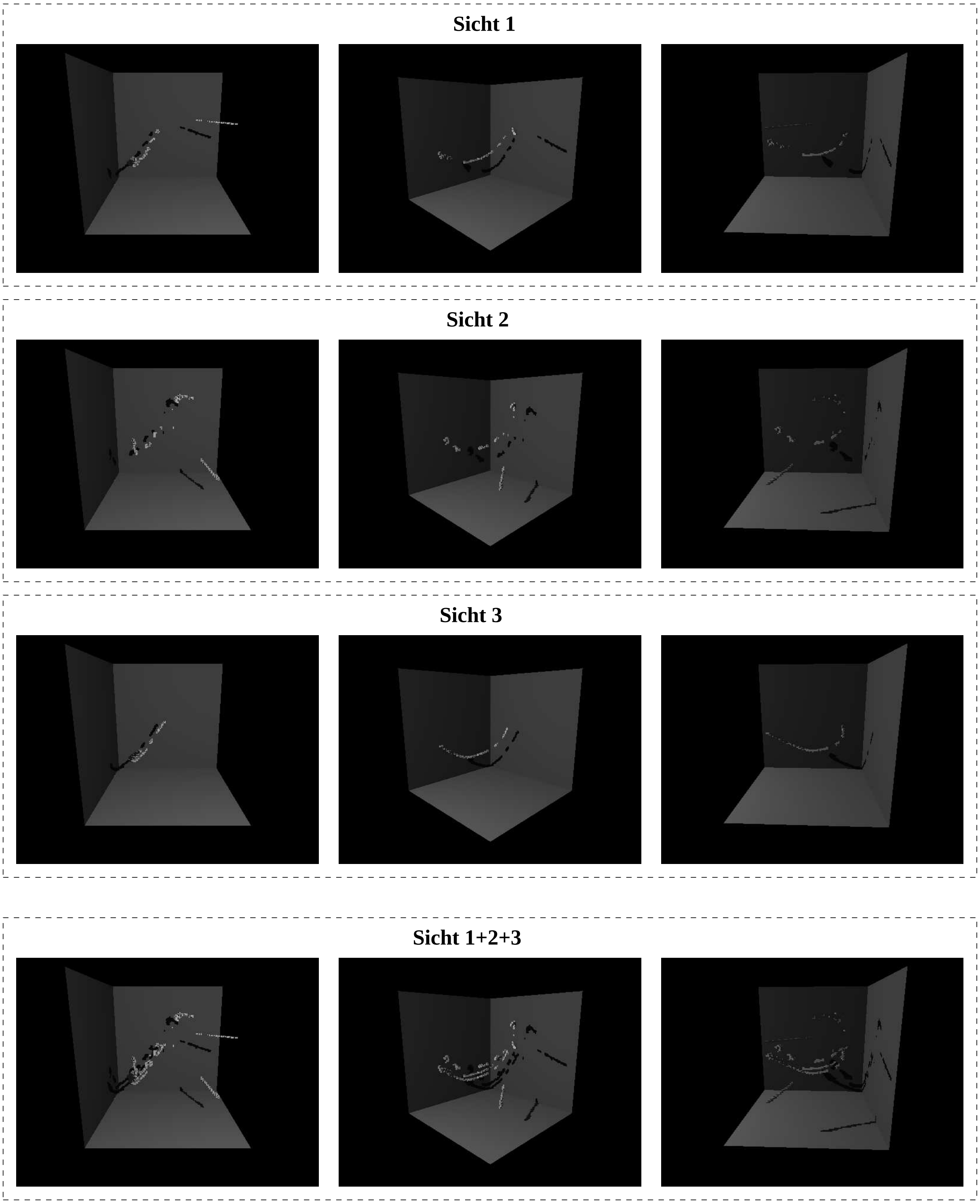


Abbildung 6.3: Beispiel Rekonstruktion Teil II, Sichten 1-3, rotes Kabel

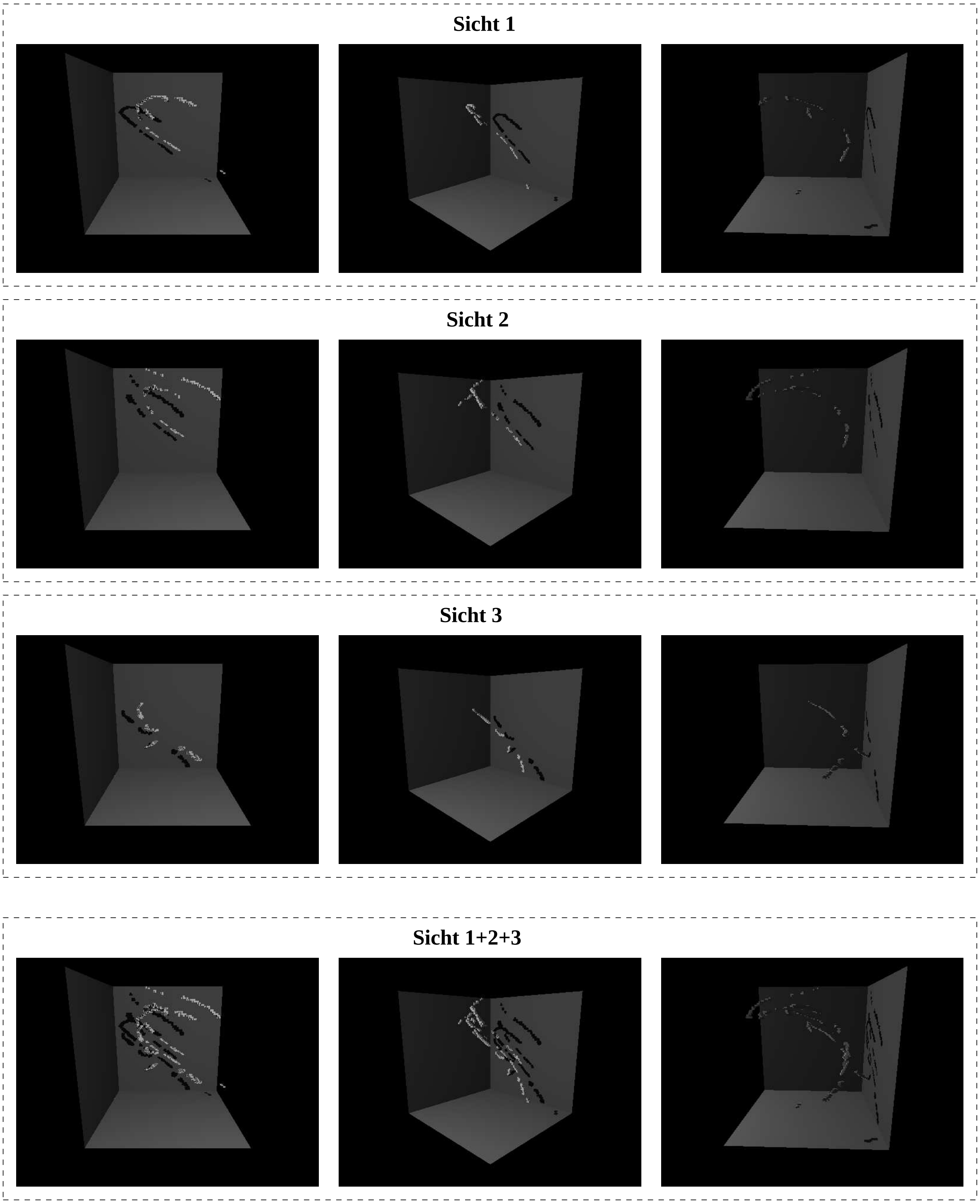


Abbildung 6.4: Beispiel Rekonstruktion Teil II, Sichten 1-3, grünes Kabel

Kapitel 7

Ein Anwendungsbeispiel des Gesamtsystems

Als Brückenschlag zum Ausgangspunkt der Arbeit, der Idee in Elektroschrottszenarien Kabel zu detektieren und demontieren, stellt dieses Kapitel ein prototypisches System zur Demontage von Kabelleitungen an Monitoren vor. Der Grundaufbau und das Vorgehen entsprechen hierbei dem Versuchsaufbau aus Abbildung 1.4 und der Darstellung eines Roboter-Vision-Systems aus Abbildung 2.1.

Da die Demontage am Monitor, bedingt durch die Bildröhre, ein Risiko bedeutet (zur Erinnerung: Hindernisdetektion ist im Rahmen dieser Arbeit nicht implementiert) beschränkt sich die Erkennung auf ein rotes Kabel, das geeigneten Abstand zur Bildröhre hat (siehe Abbildung 7.1 - aufgrund der Auflösung schwer erkennbar!).

Zur Ausführung der Demontage-Aufgabe sind vier Problemstellungen zu lösen:

- Sichtengenerierung,
- Bildanalyse (Kabellokalisation),
- 3D-Rekonstruktion,
- Greifen des Kabels zur Demontage.

Die *Sichtengenerierung* erfolgt durch Festlegung dreier Kamerapositionen. Abbildung 7.1 zeigt die drei Sichten von zwei unterschiedlichen Beobachtungspunkten aus. Die Positionen sind so gewählt, daß eine gleichmäßige, ca. 90° abdeckende Sicht gegeben ist. Für die hier verwendeten Sichten wurde auf einen weiteren Freiheitsgrad, die Rotation des Drehtellers, verzichtet.

Die *Bildanalyse* erfolgt so, wie in den Kapiteln 3-4 vorgestellt. Aus den daraus resultierenden Binärbildern der Kabelverläufe wird mittels der Methoden aus den Kapiteln 5-6 eine

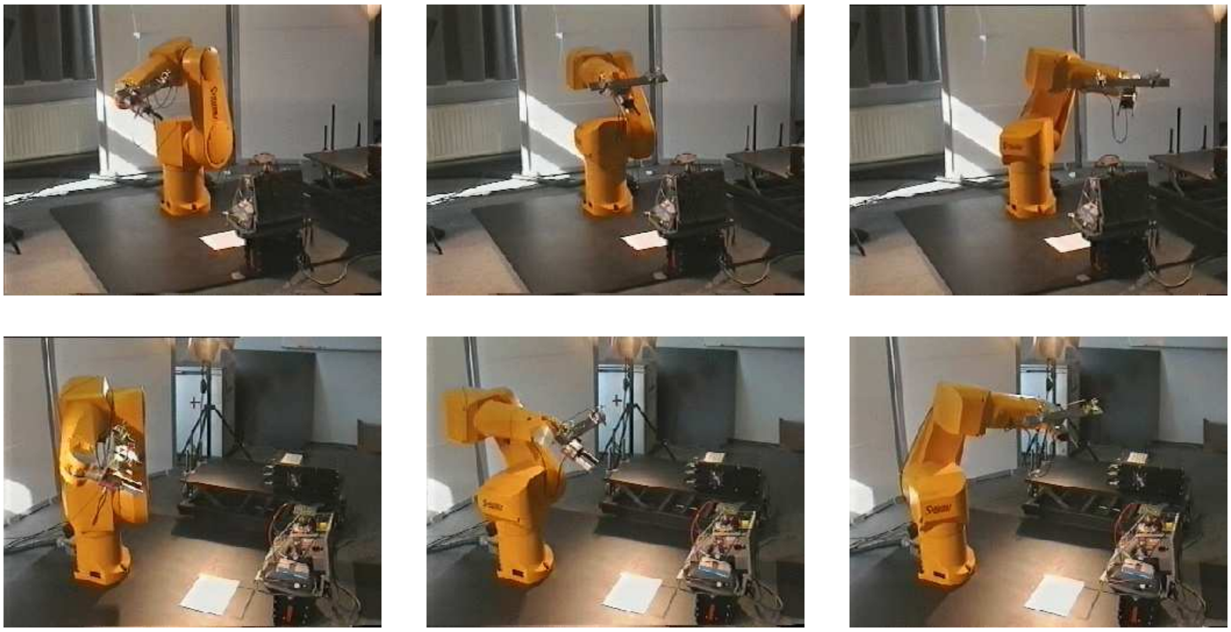


Abbildung 7.1: Rekonstruktion aus drei Kamerasichten, dargestellt aus zwei Perspektiven

3D-*Rekonstruktion* erstellt. Die Abbildungen 7.2 und 7.3 zeigen die Lokalisations- bzw. Rekonstruktionsergebnisse als Screenshots. Die Abbildungen sollen die Ausgaben des Systems zur Laufzeit demonstrieren und sind als Druckbild nur schwer erkennbar. Nachdem das Objekt lokalisiert und rekonstruiert ist, bleibt als letzte Aufgabe das *Greifen*. Angesichts

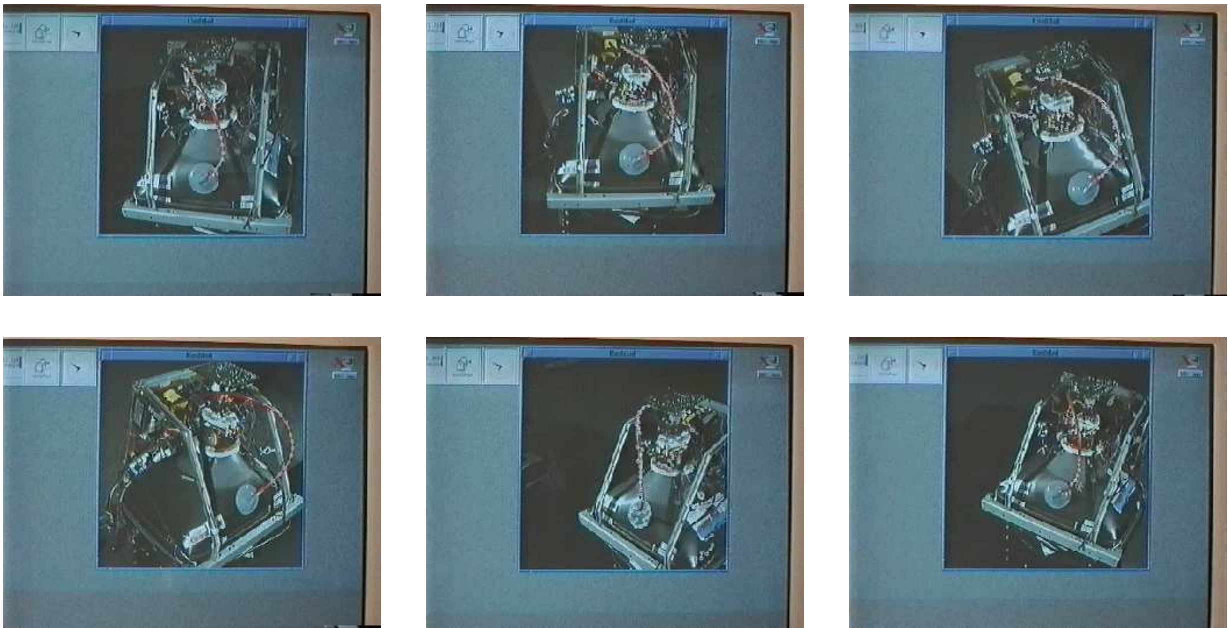


Abbildung 7.2: Kabelerkennung in den Stereobildern (Screenshots)

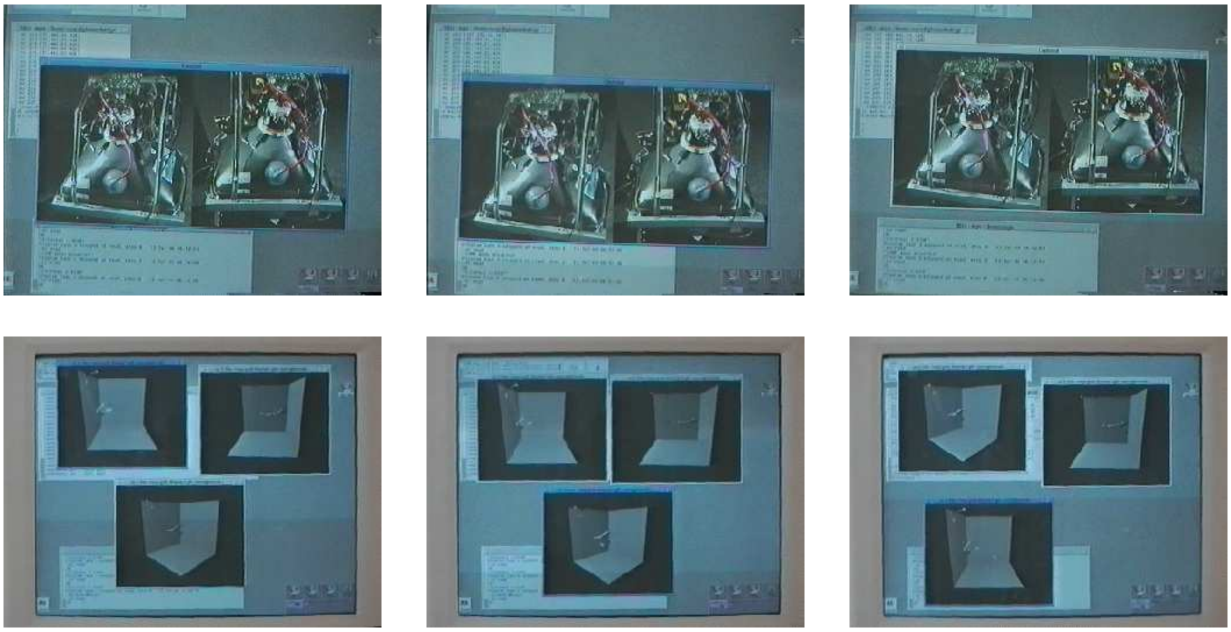
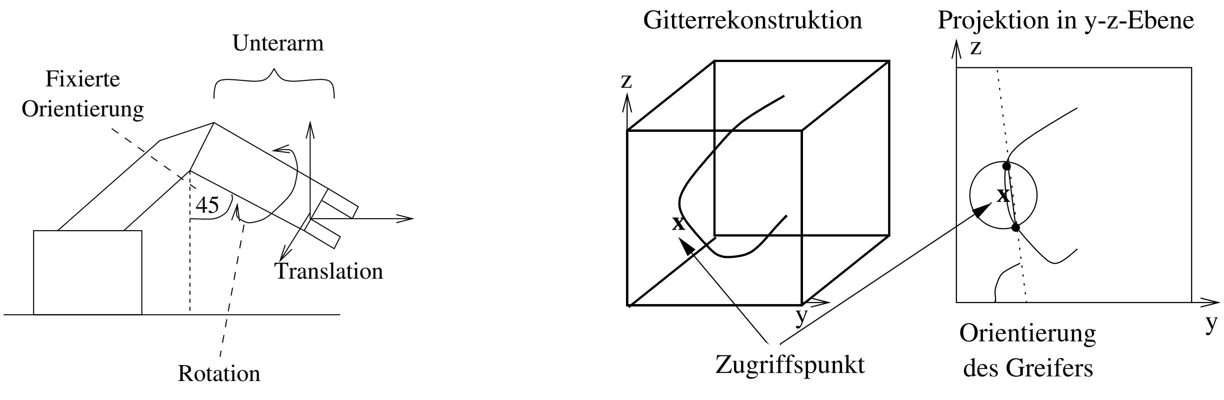


Abbildung 7.3: Stereomatching und Gitter-Rekonstruktion des Kabels

der klar definierten Aufgabe beschränken wir uns auf eine verhältnismäßig eingeschänkte Greifbewegung, eingeschränkt in dem Sinne, daß nicht alle Freiheitsgrade des Roboterarms ausgenutzt werden. Dieses ist auch in Hinblick auf die Vorsichtsmaßnahmen bzgl. der Bildröhre sinnvoll. Abbildung 7.4(a) zeigt die genutzten Freiheitsgerade: Rotation des "Unterarmes" und Translation in alle Dimensionen.

In Abbildung 7.4(b) ist die bzgl. des Kabels gewählte Orientierung illustriert. Als Zu-



(a) (b)

Abbildung 7.4: Freiheitsgrade für den Greifvorgang (a) und Orientierungswahl (b)

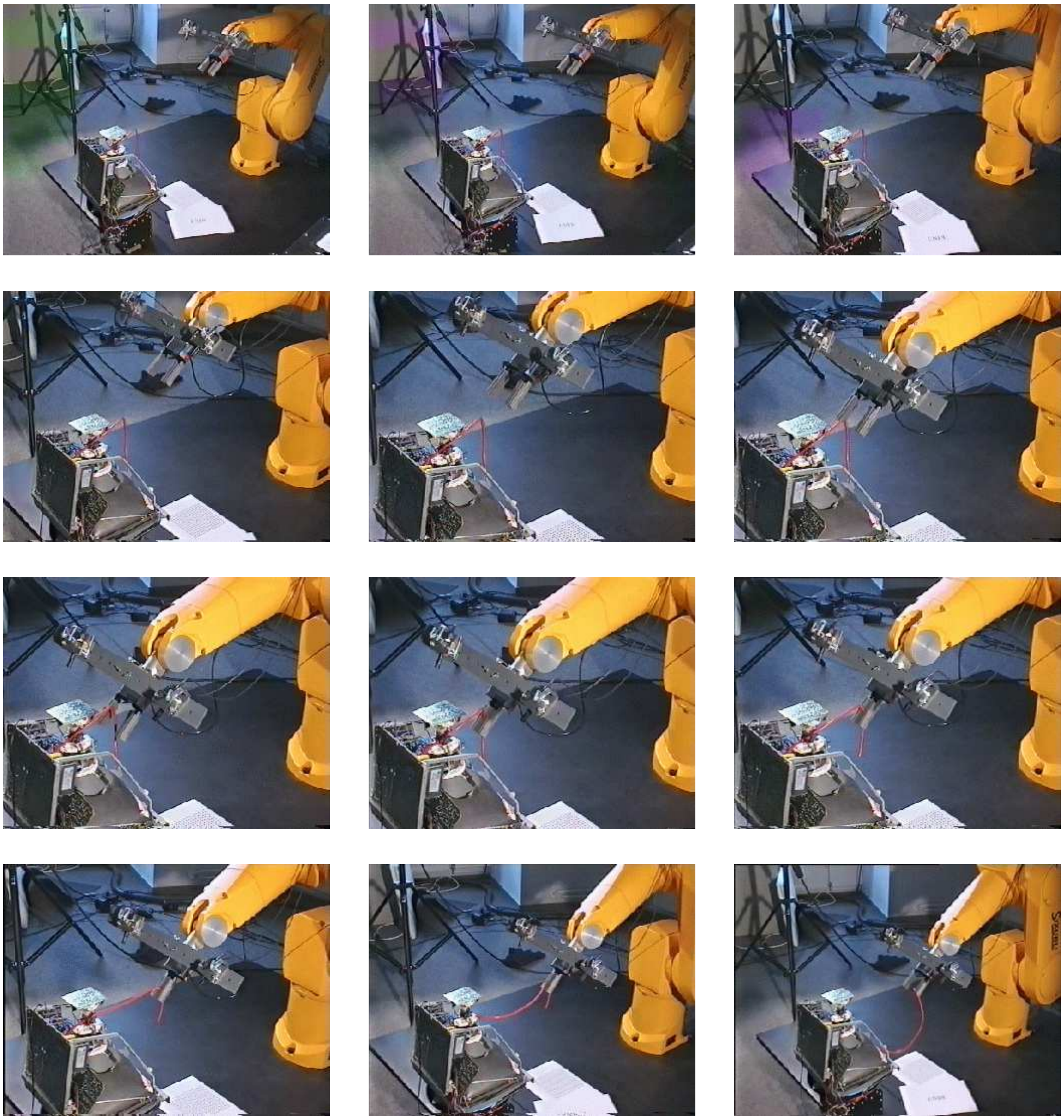


Abbildung 7.5: Bildfolge: Greifen des Kabels und Demontage

griffspunkt wird der im Gitter der z-Achse des Roboter-Basis-Koordinatensystems (siehe Abbildung 2.2) nächstgelegene Punkt gewählt. Der Gitterinhalt wird auf die z-y-Ebene des Basis-Koordinatensystems projiziert, um den Zugriffspunkt herum zwei Punkte gleichen (euklidischen) Abstandes gewählt und durch diese beiden Punkte eine Gerade gelegt. Die Orientierung(Rotation) des Unterarmes (und damit der Finger) ist durch diese - wie in Abbildung 7.4(b) ablesbar- gegeben. Abbildung 7.5 zeigt eine Bildfolge des Greifens des roten Kabels und dessen Demontage.

Kapitel 8

Zusammenfassung und Ausblick

Zusammenfassung

Ausgehend von der Idee, in Elektroschrott-Szenen Kabel erkennen zu wollen, haben wir festgestellt, daß die Komplexität der Szenen, die Bildqualität und das Fehlen einer geometrischen Objektbeschreibung einen alternativen Ansatz zur Lösung verlangen: Qualitatives Lokalisieren und Rekonstruieren - oder die Idee, Fehler zu tolerieren, wenn nur das Ergebnis im Sinne der Zielsetzung ausreichend ist.

Im Rahmen der Arbeit werden im Bild die Objekte auf Basis einer Fuzzy-Art-Datenbank zuerst farbbasiert lokalisiert, und dann diese Ergebnisse formbasiert verifiziert. So wird erreicht, daß nur die Objekte oder Objektfragmente als Kabel klassifiziert werden, die sowohl die Farb- als auch Formkriterien erfüllen. Anschließend erfolgt eine 3D-Rekonstruktion mittels der epipolaren Geometrie und von unter Verwendung von 3D-Besetztheitsgittern zur Akkumulation der 3D-Information. Die 3D-Rekonstruktion erfordert den Einsatz eines kalibrierten Kamerasystems. Hieraus resultiert die Notwendigkeit einer Entzerrung der Kamerabilder der 5.6mm-Optik, so daß eine ausreichende Genauigkeit der Kalibrierung erreicht wird. Die 3D-Gitter ermöglichen es, aus mehreren Ansichten bzw. deren einzelnen Rekonstruktionsergebnissen eine Gesamtrekonstruktion des Objektes zu erlangen.

In einem prototypischen Beispiel zur Demontage haben wir noch einmal die Gesamtfunktionaliät des Systems demonstriert. Man sieht, daß das Ziel, Greifen der Objekte, erreicht wird, trotz offensichtlicher Ungenauigkeit in der Gesamtrekonstruktion.

Die Wahl einer generischen Objektbeschreibung läßt leicht eine Erweiterung des Systems auf andere Objekte zu. Weitere Einsatzmöglichkeiten liegen zum Beispiel in der Demontage der Rahmen des Monitors und anderer “länglicher” Objekte.

Ausblick

Zielsetzung der Arbeit ist die Entwicklung des Gesamtsystems, was zur einer Top-Down-Entwicklung führt. Die einzelnen Komponenten sind Lösungen der Teilprobleme, lassen aber noch Raum für Verbesserungen.

Bildqualität und Farbsegmentierung

Die Hauptsensibilität des Systems liegt im Bereich Bildaufnahme und Farbbildanalyse. Die Lösung der Hardwareprobleme ist sicherlich durch eine integrierte Fertigung der Teilkomponenten (Roboter, Kamera, Spannungsversorgung etc.) möglich. Im Bereich Farbbildanalyse liegen die Aufgaben weiterführender Arbeiten schwerpunktmäßig bei der Verbesserung der Farbrauminterpretation und -analyse. Speziell sensiblere Verfahren zur Segmentierung sind erfolgsversprechend. Auf Basis von Verbesserungen in diesem Bereich ließen sich mit unserem formbasierten Ansatz und der Rekonstruktion stabilere und fehlerminimalere Ergebnisse erzielen.

Lokale Objektverfolgung

Ein weiteres Standbein der Kabelerkennung könnte eine Integration von Tracking-Verfahren sein, die Objekte im Bild/in Bildfolgen verfolgen. In unserem Falle, ist ein Ansatz der lokalen Kabelverfolgung durch einen erscheinungsbasierten Ansatz angedacht.

Verbesserung der Effizienz

Abschließend sei angemerkt, daß die Implementierung nicht geschwindigkeitsoptimiert ist. Das Bearbeiten der Stereobilder kann genauso unabhängig voneinander erfolgen, wie parallel dazu das Eintragen der Rekonstruktionsergebnisse der vorherigen Sicht in das Gitter erfolgen kann. Es ergeben sich also hier auf hoher Ebene Parallelisierungsmöglichkeiten, genauso wie eine weitere Verfeinerung der reinen Bildoperationen Geschwindigkeitsgewinn verspricht.

Literaturverzeichnis

[Aloimonos93] Y. Aloimonos *Active vision revisited* Active Perception, pp.1-18. Lawrence Erlbaum Associates Publishers. New Jersey, 1993.

[Blaszka94] T. Blaszka, R. Deriche *A Model Based Method for Characterization and Location of Curved Image Features* Rapport de recherche n 2451 INRIA , Cedex, December 1994.

[Bishop95] C.H. Bishop *Neural Networks for Pattern Recognition* Clarendon Press, pp. 332-384, Oxford 1995.

[Böhmer74] K. Böhmer *Spline-Funktionen* Teubner Verlag, 1974.

[deBoor78] C. de Boor *A Practical Guide To Splines* Springer-Verlag, New York, 1978.

[CaGro87] G.A. Carpenter, S. Grossberg *ART 2: self-organisation of stable category recognition codes for analog input patterns* Applied Optics. Vol.26,pp. 4919-4930. December 1987.

[CaGroRo91] G.A. Carpenter, S. Grossberg, D.B. Rosen *Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by Adaptive Resonance System* Neural Networks, Volumne 4, pp. 759-771, 1991.

[CoFish83] D.M. Connah, C.A. Fishbourne *Segmentation by colour* Radio electronic engineering, pp. 153-156, April 1983.

[Craig89] J.J. Craig *Introduction to Robotics: Mechanics and Control / 2nd edition* Addison-Wesley Publishing Company, 1989.

[Elfes87] A. Elfes *Sonar-based real-world mapping and navigation* IEEE Journal of Robotics and Automation, 3(3), pp.249-265, 1987.

[Eubank88] Eubank *Spline Smoothing and Nonparametric Regression* M.D., New York 1988.

[Farin94] G. Farin *Kurven und Flächen im Computer Aided Geometric Design* Vieweg, 1994.

[Faugeras93] O. Faugeras *Three-Dimensional Computer Vision: A Geometric Viewpoint* The MIT Press, Cambridge, MA, 1993.

[Foley90] J.D. Foley, A. van Damm, S.K. Feiner, J.F. Hughes Computer Graphics: principles and practice Addison Wesley Publishing Company, second edition, 1990.

[Haberäcker91] P. Haberäcker *Digitale Bildverarbeitung* Carl Hanser Verlag München Wien, 1991.

[Haberäcker95] P. Haberäcker *Praxis der Digitalen Bildverarbeitung* Carl Hanser Verlag München Wien, 1995.

[Hoschek92] J. Hoschek, D. Lasser *Grundlagen der geometrischen Datenverarbeitung* B. G. Teubner, 1992.

[Jähne93] B. Jähne *Digitale Bildverarbeitung* Springer-Verlag, 1993.

[Jazwinsky70] A.M. Jazwinsky *Stochastic Processes and Filtering Theory* Academic, New York, 1970.

[Kalman60] R.E. Kalman *A new approach to linear filtering and prediction problems* J.Basic.Eng., vol.82D, pp.35-45, March 1960.

[Kender91] J.R. Kender, R. Kjeldsen *On Seeing Spaghetti: A Novel Self-Adjusting Seven Paramter Hough Space for Analyzing Flexible Extruded Objects* SPIE Vol.1570 Geometric Methods in Computer Vision, 1991.

[Kohonen82] T. Kohonen *Self-Organized Formation of Topologically Correct Feature Maps* Biological Cybernetics, 43:59-69, 1982.

[Kohonen84] T. Kohonen *Self-Organization and Associative Memory* Springer-Verlag, Berlin, 1984.

[Lenz88] R. Lenz *Zur Genauigkeit der Videometrie mit CCD-Sensoren* Proceedings 10.DAGM Symp. Mustererkennung 1988, Informatik-Fachberichte 180, S.179-189, Springer-Verlag Berlin, 1988.

[Martin-Moravec96] M.C. Martin, H.P. Moravec *Robot Evidence Grids* Technical Report, Mobile Robot Laboratory, CMU, 1996.

[Maybeck82] P.S. Maybeck *Stochastic Models, Estimation and Control, volume 1* Academic, New York, 1982.

[Moore89] B. Moore *Art 1 and Pattern Clustering* Proceedings 1988 Connectionist Models Summer School, pp.174-185, 1989.

[Moravec-Elfes85] H.P. Moravec, A. Elfes *High Resolution Maps from Wide Angle Sonar* Proceedings of the 1985 IEEE International Conference on Robotics and Automation, pp 116-121, St. Louis, March 1985.

[Moravec96] H.P. Moravec *Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids* Technical Report, Mobile Robot Laboratory, CMU-RI-TR-96-34, September 1996.

[Murase93] H. Murase, S. Nayar *Visual learning and recognition of 3D objects from appearance* Internatiol Journal of Computer Vision, Vol. 14, pp. 5-24, 1993.

[Niemann83] *Klassifikation von Mustern* Springer Verlag, 1983.

[Niemann90] *Pattern Analysis and Understanding* Springer Verlag, 1990.

[Olomski89] J. Olomski *Bahnplanung und Bahnführung von Industrierobotern* Fortschritte der Robotik, Bd. 4, Vieweg, Braunschweig, 1989.

[Rehrmann95] V. Rehrmann *Segmentierung von Farbbildern* 1.Heidelberger Bildverarbeitungsforum, November 1995.

[Sarkar93] S. Sarkar, K.L. Boyer *Perceptual Organization in Computer Vision: A Review and a Proposal for a Classificatory Structure* IEEE Transactions on Systems, Man and Cybernetics VOL. 23, NO.2, pp.382-397, March/April 1993.

[Samal97] A. Samal, J. Edwards *Generalized Hough transform for natural shapes* Pattern Recognition Letters 18, pp.473-480, 1997.

[Seibert91] M. Seibert *Neural Networks for machine vision: Learning three-dimensional object representations* Ph.D. Thesis, Boston University, 1991.

[Ser95] P-K. Ser, W-C. Siu *A New Generalized Hough Transform for the Detection of Irregular Objects* Journal of visual communication and image representation, Vol.6, No.3, pp. 256-264, September 1995.

[Smith78] A.R. Smith *Color gamut transform pairs* Computer Graphics, 12(3), pp 12-19, 1978.

[Sorensen70] H.W. Sorensen *Least-squares estimation : from Gauss to Kalman* IEEE Spectrum, vol.7, pp 63-68, July 1970.

[Spirkovska94] L. Spirkovska, M.B. Reid *High-Order Neural Networks Appleid to 2D and 3D Recognition* Machine Learning vol.15, pp. 169-199, 1994.

[Stoer89] J. Stoer *Numerische Mathematik* Springer-Lehrbuch, 5. Auflage, 1989.

[SwainBallard91] M.J. Swain, D.H. Ballard *Color Indexing* International Journal of Computer Vision, 7(1),pp. 11-32, 1991.

[Thrun96] S. Thrun *Learning maps for indoor mobil robot navigation* Technical Report, Mobile Robot Laboratory, CMU, 1996.

[Ude96] A. Ude *Rekonstruktion von Trajektorien aus Stereobildfolgen für die Programmierung von Roboterbahnen* Fortschritt-Berichte, Reihe 10, Nr.448, VDI-Verlag, 1996.

[Wahl84] F.M. Wahl *Digitale Bildsignalverarbeitung* Berlin, Heidelberg, New York, Tokyo, Springer Verlag, 1984.

[Zhang95] Z. Zhang *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting* INRIA No.2676, Oktober 1995.

Danksagung

An dieser Stelle möchte ich jedem Dank aussprechen, der in irgendeiner Form zum Gelingen dieser Arbeit beigetragen hat. Von Seiten des Lehrstuhls sind besonders mein Betreuer Dr. Josef Pauli für den inhaltlichen, sowie Gerd Diesner und Hendrik Schmidt für den technischen Support hervorzuheben.

Diese Arbeit ist der Abschluß eines langjährigen Studiums, welches ohne die Unterstützung und Absicherung meiner Eltern nicht möglich gewesen wäre. Ihnen gilt somit der größte Dank für diese für Außenstehende oft unsichtbare Leistung.

Erklärung:

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig verfaßt und ausschließlich die angegebenen Hilfsmittel und Quellen verwendet habe.

Kiel, den