Manipulatorregelung durch fortwährende visuelle Rückkopplung

vorgelegt von
Arne Schmidt
Kiel, im Mai 1998

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

Zusammenfassung

Diese Arbeit beschreibt ein Gesamtsystem, das unter Verwendung eines Roboterarmes und eines den Roboterarm beobachtenden Stereokamerakopfes die spezifische Aufgabe lösen soll, ein Objekt hinsichtlich seiner Grundfläche zu klassifizieren, und dieses dann unter Verwendung von visuell gewonnenen Informationen an ein für dieses Objekt bestimmtes Loch anzunähern. Das System gliedert sich in einzelne Module, wobei jedes Modul eine Verwirklichung eines Wahrnehmungs-Aktions-Zyklus ist. Sowohl die Ausrichtung des Objektes zur Kamera für die Klassifikation als auch die Annäherung des Objektes an das Loch geschieht mit Hilfe einer fortwährenden visuellen Rückkopplung, d.h der Ablauf von visuell gewonnen Informationen aus den beiden Kamerabildern und der daraus resultierenden Roboterbewegung im 3D Weltkoordinatensystem ist in einem geschlossenen Regelkreis verankert. Die einzuführenden Objekte sind Quader und Zylinder. Die Erkennung der Objekte während der Annäherung zum Loch geschieht mit Hilfe orientierungsselektiver Hough-Transformation und Ellipsenextraktion. Die Erkennung benutzt dabei nur allgemeine Annahmen über die Objekte und kein spezifisches Modell aus einer Datenbank.

Inhaltsverzeichnis

1 Einführung **1**

1.1 Gegenstand der Arbeit 1

1.2 Einbettung in die verhaltensbasierte Robotik 3

1.3 Verwandte Arbeiten 3

1.4 Aufbau der Arbeit 6

2 Die Systemumgebung **9**

2.1 Der Roboterarm 9

2.2 Der Kamerakopf 11

2.3 Das Multithread-Prozessorsystem 12

2.4 Der Versuchsaufbau bzw. -ablauf 12

3 Kalibrierung **17**

3.1 Transformationen 17

3.2 Die Jakobimatrix 19

4 Regelungstechnik **23**

4.1 Grundlegende Begriffe der Regelungstechnik 23

4.2 Reglerarten 24

4.3 Der Regelungsalgorithmus 26

4.4 Experimente 29

5 Regelung für die Objektklassifizierung **33**

5.1 Objektausrichtung zur Kamera 33

5.1.1 Ermittlung des Sehstrahls 34

5.1.2 Ausrichtung zum Sehstrahl 36

5.1.3 Vergenzregelung der Kamera 39
5.1.4 Fingersegmentierung und Konturextraktion 40
5.2 Objektklassifikation 43
5.2.1 Grundflächenrepräsentation 43
5.2.2 Grundflächenerkennung 46

6 Regelung für die Annäherung 49
6.1 Istwertbestimmung beim Quader 50
6.1.1 Kantenextraktion 50
6.1.2 Erkennung von Berandungssegmenten 58
6.2 Istwertbestimmung beim Zylinder 61
6.2.1 Ellipsenextraktion und -approximation 62
6.2.2 Ellipsenerkennung 72
6.3 Sollwertbestimmung beim Loch 74
6.4 Regelung bei der Annäherung 78

7 Diskussion und Ausblick 85
7.1 Diskussion 85
7.2 Ausblick 86

A Mathematische Ergänzungen 89
A.1 Herleitung der Modellparameter 89
A.2 Generalisiertes Eigenwertproblem 91

Literaturverzeichnis 95

Kapitel 1

Einführung

1.1 Gegenstand der Arbeit

Der Großteil der heutigen Roboter arbeitet in Fabriken, wo die Umgebung speziell auf den Roboter eingestellt wird, so daß sie Arbeiten in fest vorgegebenen Positionen gut ausführen können. Ändert sich hingegen die Umgebung, kann der Roboter aufgrund fehlender Sensoren auf die Veränderung der Umwelt nicht reagieren. Wird ein Roboter aber mit Sensoren ausgestattet, wie z.B. einer Kamera, erhöht sich die Vielseitigkeit und das Anwendungsgebiet eines Roboters fundamental. Der Roboter erhält durch Wahrnehmung eine Information, auf die er mit einer bestimmten Aktion reagieren kann. Die technische Realisierung dieses *Wahrnehmungs-Aktions-Zyklus* ist Aufgabe der visuellen Robotik. Man unterscheidet hierbei drei verschiedene Anwendungsgebiete:

- das visuell basierte Manipulieren,
- das aktive visuelle Wahrnehmen und
- das visuell basierte Fortbewegen[1].

Dabei versteht man unter aktiver visueller Wahrnehmung die Regelung der visuellen Aufmerksamkeit und unter Manipulation die Veränderung der Lage (oder Form) eines Gegenstandes. Die meisten Anwendungen eines visuell arbeitenden Manipulators arbeiten bei einer Lageveränderung eines gegriffenen Gegenstandes nach der direkten Methode:

1. Schaue Situation an und ermittle Ziel.
2. Vollführe vollständige Bewegung zum Ziel.

Es findet hier also kein geschlossener Wahrnehmungs-Aktions-Zyklus statt, vielmehr die einmalige Aktion auf eine Wahrnehmung. Die Genauigkeit der resultierenden Ein-Schritt-Operation hängt somit direkt von der Genauigkeit des Sensors und des End-Effektors

[1]Dieses Gebiet wird in dieser Arbeit nicht behandelt, da sich der Manipulator und die Kamera fest auf dem Boden befinden.

bzw. deren Kalibrierung ab. Eine Alternative, um die Genauigkeit einer Operation zu erhöhen, ist die Verwendung einer geschlossenen Kontrollschleife, die *fortwährende visuelle Rückkopplung*, also *visuell basierte Regelung* benutzt ("visual servoing"). [Hill79] war der erste, der diesen Begriff erwähnte. Hauptziel unserer Arbeit ist es, ein bestimmtes, an seiner Grundfläche erkanntes Objekt, das sich in der Roboterhand befindet, an ein für das Objekt bestimmtes Loch durch visuelle Regelung anzunähern. Die anzunähernden Objekte sind Quader und Zylinder. Die visuelle Information liefert hierbei ein mit zwei Kameras ausgestatteter Kamerakopf, der die Szene beobachtet. Um dieses Ziel zu erreichen, ist die Arbeit in bestimmte aufeinanderfolgende Module aufgeteilt, die alle zum Gesamtsystem beitragen. Jedes Modul wird in einer bestimmten Phase eingesetzt. Die Phasen bestimmen den chronologischen Ablauf. Die Module der Phasen werden hier einführend beschrieben. Eine genauere Beschreibung findet sich in Kapitel zwei.

In der *Kalibrierungsphase* wird durch visuelle Regelung der Sehstrahl der linken Kamera ermittelt und damit verbunden auch die Ausrichtung der Roboterhand parallel zur Bildebene der linken Kamera. Die visuelle Information hierfür wird durch eine in der Roboterhand befindliche Leuchtdiode geliefert, die auf die Bildebene einen Lichtpunkt projiziert. Das zweite Modul dieser Phase beinhaltet die Regelung der Vergenz der rechten Kamera, bis sich der Lichtpunkt auch in der Mitte der rechten Bildebene befindet. Diese Fixierung des Lichtpunktes soll die spätere Suche des Objektes im Bild auf einen Suchbereich beschränken. Das dritte Modul stellt über Ermittlung von Projektionsmatrizen das Verhältnis zwischen Weltkoordinaten und den jeweiligen Bildkoordinaten her. Die *Klassifikationsphase* beinhaltet die Rotation der zur linken Bildebene ausgerichteten Hand mit dem Objekt, bis die Finger senkrecht im Bild sind. Das zweite Modul dieser Phase ist die Annäherung der Hand zur Kamera, bis eine bestimmte Größe des Fingers der Hand und damit auch implizit des Objektes erreicht ist. Das letzte Modul bewirkt eine Objekterkennung über die Erkennung der Grundfläche des Objektes. Die *Annäherungsphase* ist die letzte Phase des Systems. Hier findet als erstes die Lochbretterkennung und die Erkennung der Löcher statt. Das zweite Modul beinhaltet dann die visuell basierte Regelung des Objektes zum Lochbrett. Abbildung 1.1 zeigt ein Schema der Phasen mit ihren Modulen.

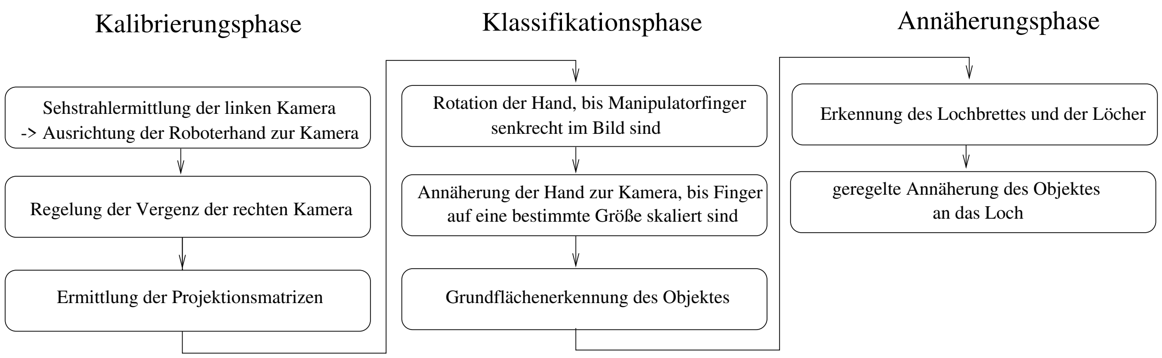


Abbildung 1.1: Übersichtsschema der Phasen mit ihren Modulen.

Es lassen sich bei Betrachtung der Phasen zwei große Anwendungen der Regelung unterscheiden. Zum einen wird die *Regelung für die Objektklassifizierung* benutzt, und zum anderen die *Regelung für die Objektmanipulation*.

Eine andere Sichtweise gegenüber den Modulen liefert der nächste Abschnitt.

1.2 Einbettung in die verhaltensbasierte Robotik

Bis auf die Module der Grundflächen- und Lochbretterkennung beinhaltet jedes Modul einen Wahrnehmungs-Aktions-Zyklus[2]. Hierbei sind verschiedene Arten zu beachten. Die Ermittlung der Projektionsmatrizen ist ein Wahrnehmungs-Aktions-Zyklus, in dem Wahrnehmung und Aktion unabhängig voneinander laufen. Dieser Zyklus dient lediglich zur Extraktion von Wissen. Die Aktion des Roboters unterstützt hierbei die Wahrnehmung. Alle anderen Module beinhalten Zyklen, in denen die Aktion abhängig von der Wahrnehmung ist. Sie zeigen aufgrund der Wahrnehmung ein bestimmtes Verhalten. Die Intentionen der Zyklen sind allerdings verschieden. So ist z.B. das Vergieren des Kopfes eine aktive Wahrnehmung. Es soll die visuelle Aufmerksamkeit gesteuert werden, um später eine bessere Selektion der Bilddaten für die Objektsuche zu gewährleisten. Der Annäherungszyklus ist dagegen - wie oben schon erwähnt- ein klassisches Beispiel für eine visuell basierte Manipulation. Jedes Modul dieser Art ist ein Beispiel für ein *elementares Verhaltensmodul*, das im parallelen Zusammenspiel mit anderen Modulen ein komplexes, verhaltensbasiertes Robotersystem zu bilden[Brooks86]. Die Anwendung der Module ist in unserer Arbeit zwar sequentiell, doch wird auf eine mögliche Parallelisierung mit anderen Modulen noch in Kapitel sieben näher eingegangen.

Der nächste Abschnitt beinhaltet verwandte Arbeiten, die sich auch mit dem Thema der visuell basierten Regelung befassen. Da unsere Arbeit neben der Regelung auch einen Schwerpunkt in der geeigneten Bildverarbeitung hat, wird im zweiten Teil insbesondere auch auf die Bildverarbeitung für einige Systeme eingegangen.

1.3 Verwandte Arbeiten

Viele Arbeiten sind auf dem Gebiet der visuell basierten Regelung veröffentlicht worden. Um einen allgemeinen Überblick über diese Thematik zu erhalten, sind insbesondere die Arbeit von [Corke93] und [Hutchinson96] zu nennen. In [Corke93] wird ein zeitlicher Überblick über die Entwicklung der visuell basierten Regelung gegeben mit vielen Beispielen über Arbeiten auf diesem Gebiet. In dem Artikel von [Hutchinson96] bekommt der Leser eine Einführung in die Grundlagen der visuellen Regelung. Beide Verfasser sprechen insbesondere die verschiedenen Architekturen eines Systems mit visuell basierter Regelung an. Es werden vier große Klassen dieser Regelungsart unterschieden, deren Definition

[2]Betrachtet man die Bildverarbeitung für die Erkennung als Aktion, so sind sogar die Module der Grundflächen und Lochbretterkennung Wahrnehmungs-Aktions-Zyklen.

auf [Sanderson80] zurückgeht. Hierbei werden zwei große Einteilungen vorgenommen, aus denen sich dann die vier Klassen ergeben. Die erste Einteilung berücksichtigt die Ermittlung des Fehlers für die Regelung. Zum einen gibt es hier die *positionsgestützte Regelung* ("position-based control"), bei der Merkmale des Objektes (z.B. Berandungssegmente) aus dem Bild extrahiert werden, und unter Verwendung eines geometrischen Modells des zu bewegenden Objektes und der Kamera die Position und Orientierung des Objektes relativ zur Kamera bestimmt wird. Die Rückkopplung, die das Inkrement liefert, um das sich der Roboter jeweils bewegen soll, wird innerhalb des Weltkoordinatensystems errechnet. Zum anderen gibt es die *bildgestützte Regelung* ("image-based control"), wo die Kontrollwerte für den Roboter direkt aus den Bildmerkmalen und somit auch direkt aus dem Kamerakoordinatensystem errechnet werden. Es tritt hier also keine vorherige Rekonstruktion des Modells auf. Diese zweite Methode hat den Vorteil, daß die Rechenzeit kürzer ist und auf Kalibrierungsfehler besser eingegangen werden kann, der Entwurf des Reglers wird hingegen komplizierter. Die zweite große Einteilung basiert auf dem Ergebnis des Reglers. Ihr liegt dabei die Fragestellung zugrunde, ob entweder die Gelenkstellungen des Roboters direkt ausgerechnet und ihm übergeben werden, oder als Ergebnis ein Punkt im Weltkoordinatensystem geliefert wird, der dann von dem Roboter-Steuerungsprogramm selbst in Gelenkstellungen umgewandelt wird unter Verwendung einer ständigen Abfrage der Gelenkstellungen, um das System zu stabilisieren. Trifft der erste Fall ein, spricht man von *direkter visueller Regelung* ("direct visual servoing"), hier ist die inverse Kinematik des Roboters nur visuell gestützt. Der andere Fall wird als *dynamisches Schauen und Bewegen* ("dynamic look-and-move") bezeichnet. Da unsere Arbeit die Information für die Regelung und das Regelergebnis direkt aus dem Bild gewinnt, läßt sich unsere Regelung als bildgestützte Regelung bezeichnen. Unser Roboter ist hierbei so ausgelegt, daß die inverse Kinematik im Steuerprogramm verankert ist, so daß man nur noch einen Weltkoordinatenpunkt angeben muß, um einen Punkt anzufahren (vgl. Kap. 2). Daher läßt sich unsere visuell basierte Regelung unter Verwendung der oben betrachteten Einteilungen insgesamt als *bildgestütztes dynamisches Schauen und Bewegen* beschreiben. Der von uns verwendete grundlegende Regelmechanismus stützt sich dabei auf den Artikel von [Hager95], der aber so modifiziert wurde, daß eine in diesem Artikel noch benötigte direkte Angabe von explizit errechneten intrinsischen und extrinsischen Kameraparametern umgangen werden kann. Es wird vielmehr eine Regelung direkt mit Hilfe der Projektionsmatrizen der Kameras erstellt. Dies hat einen großen Vorteil gegenüber der Genauigkeit der Regelung (vgl. Kap. 4).

Ein Beispiel für eine positionsgestützte visuelle Regelung bietet der Artikel von [Wunsch96], in dem ein modellbasierter Ansatz vorgestellt wird, um die räumliche Lage eines bewegten Objektes aus einer Bildsequenz zu bestimmen. Hierbei werden aus den Bilddaten extrahierte Merkmale in räumliche Lagebedingungen umgerechnet, mit denen unter Ausnutzung einer geometrischen Modellbeschreibung die dreidimensionale Objektlage bestimmt wird. Mit Hilfe eines geeigneten Kalman-Filters wird danach aus der bis dahin ausgewerteten Bildsequenz die räumliche Objektbewegung geschätzt. Der Endeffektor, an dem eine Kamera angebracht ist, nimmt dann jeweils eine Referenzposition relativ zum bewegten

Objekt ein.

Eine Mischform der bildgestützten und der positionsgestützten Regelung beschreibt [Castaño94]. Hier wird ein Regler entworfen, der bildgestützte Regelung parallel zur Bildebene und positionsgestützte Regelung senkrecht zur Bildebene realisiert. Es wird hiermit eine Bewegung entlang eines Projektionstrahls einer Kamera realisiert, die fest positioniert ist und die Szene aus ca. 3 m Entfernung beobachtet. Die *Bildverarbeitung* beschränkt sich hierbei auf die Ermittlung des Schwerpunktes eines Lichtpunktes, der von einer Leuchtdiode stammt, die der Roboter in der Hand hält.

Eine Art der Bildverarbeitung, die auf Farbsegmentierung zurückgeht, ist in dem Artikel von [Harrell89] angegeben, wo die visuell basierte Regelung benutzt wird, um Früchte aufzusammeln. Die Farbe dient hier zur Unterscheidung zwischen Früchten und Blättern.

Einen anderen Ansatz der Bilderkennung für visuell basierte Regelung stellt [Allen93] vor. Das gestellte Ziel ist hierbei, einen auf einer Kreisbahn befindlichen Spielzeugzug mit der Roboterhand zu greifen. Beobachtet wird die Szene von einem statischen Kamerapaar. Durch Berechnung des optischen Flusses wird der Schwerpunkt der Bewegung des Objektes im linken und rechten Bild ermittelt als Eingabe für den Regler. Die Bewegung ist hierbei Voraussetzung für die Detektion des Objektes.

Um der komplexeren Szene in unserer Arbeit gerecht zu werden, wird hier die Erkennung des anzunähernden Objektes und des Lochbrettes mit Hilfe von *orientierungsselektiver Hough-Transformation* und *Ellipsenextraktion bzw. -approximation* durchgeführt mit anschließenden Gruppierungen der Berandungsegmente bzw. der Halbellipsen mittels Quasi-Invarianten ([Binford93] und [Mundy92]). Dies bedeutet für den Wahrnehmungs-Regelungs-Zyklus im Gegensatz zu den oben erwähnten Arbeiten mit nur einfacher Bilderkennung zwar eine gewisse zeitliche Verzögerung, der Anspruch der Bilderkennung für die Regelung ist dafür gestiegen. Die für die Erkennung dabei nötige perzeptuelle Gruppierung des 2D Bildes läßt sich nach [Sarkar93] in vier aufeinander aufbauende Klassen einteilen, je nach Betrachtung der Hauptmerkmale bei der Gruppierung. Es gibt hierbei erstens die Gruppierung auf der *Signalebene* ("signal level"), wo z.B. Berandungen als Verbindungen von Pixeln mit gleichen photometrischen Eigenschaften betrachtet werden, zweitens die abstraktere *Ebene der Primitiva* ("primitive level"), in der z.B. die aus der Signalebene extrahierten Merkmale zu eindimensionalen Kontursegmenten und Berandungsbögen approximiert werden, z.B. mit Messung der Krümmung der Kontur, drittens die *einfache Strukturierungsebene* ("structural level"), in der 2D Polygone und Ellipsen betrachtet werden, die aus Segmenten und Berandungsbögen zusammengesetzt sind und viertens die *komplexe Strukturierungsebene* ("assembly level"), wo die Polygone und Ellipsen zu Gruppen zusammengefaßt betrachtet werden. Die Kombination dieser Gruppierungsweisen ist eine Forderung im Ausblick des Artikels von [Sarkar93]: *"There is a need for research into frameworks for integration of various Gestaltic cues including nongeometric ones ..."* Unsere Suche nach den relevanten Berandungssegmenten ist eine Kombination der Betrachtungsweisen auf unterschiedlichen Ebenen. Die Hough-Transformation für die Kantenextraktion ist z.B. ein Verfahren für die Gruppierung auf der Ebene der Primitiva, die Erweiterung zur orientierungsselektiven Hough-Transformation fügt Elemente der Signalebene hinzu.

Die Erkennung mittels Symmetrieeigenschaften des Quaders und des Zylinders (vgl. Kap. 6.1.2 und Kap. 6.2.2) ist Teil der komplexen Strukturebene.

Im folgenden wird ein Überblick über die einzelnen Kapitel der Arbeit gegeben.

1.4 Aufbau der Arbeit

Kapitel zwei zeigt im ersten Teil die für die Versuche verwendeten technischen Systemkomponenten. Zu ihnen gehören der Roboterarm, der Kamerakopf und das Multithread-Prozessorsystem. Außerdem gehen wir auf den oben schon angedeuteten Versuchsaufbau bzw. -ablauf genauer ein. Kapitel drei beinhaltet im wesentlichen die Kalibrierung des Systems. Wir beschreiben die für die Umrechnung vom Weltkoordinatensystem[3] zum Bildkoordinatensystem nötigen Projektionsmatrizen und deren Ermittlung. Mit deren Hilfe läßt sich zum einen die Transformation beschreiben, die einen Punkt vom 3D Weltkoordinatensystem auf das 2D Bildkoordinatensystem projiziert. Zum anderen läßt sich die Transformation angeben, die mit Hilfe von zwei 2D Bildpunkten einen 3D Weltkoordinatenpunkt errechnet. Außerdem beschreibt dieses Kapitel die für den Regelalgorithmus nötige Jakobimatrix, die die *Änderungen* zwischen 2D Bildpunkten und 3D Weltkoordinatenpunkten ausdrückt. Mit Hilfe dieser Grundlagen läßt sich nun ein bestimmter Regelalgorithmus entwerfen. Dies geschieht in Kapitel vier. Hier wird zunächst eine Einführung in die Grundlagen der Regelung gegeben. Als Beispiele für mögliche Regelungsarten werden der Proportional-Regler (P-Regler), der Integral-Regler (I-Regler) und der Differential-Regler (D-Regler) beschrieben, sowie zwei Kombinationen der einzelnen Regler, resultierend in den Proportional-Integral-Regler (PI-Regler) und den Proportional-Integral-Differential-Regler (PID-Regler). Desweiteren wird eine Begründung für den schließlich von uns verwendeten Regler gegeben. Darauf folgt der Entwurf des grundlegenden von uns benutzten Regelalgorithmus. Als Beispiel für den grundlegenden Entwurf wird die Ansteuerung eines im Raum definierten Punktes mit Hilfe eines Lichtpunktes gezeigt, der durch eine Leuchtdiode erzeugt wird, die sich in der Roboterhand befindet. Die Wahl dieses Beispiels begründet sich in der Tatsache, daß sich auch in der späteren von uns gezeigten Annäherung die Regelung auf die Annäherung eines Punktes im Raum zu einem anderen reduzieren läßt. Als Abschluß des Kapitels folgen einige Experimente, die auf dem grundlegenden Regelalgorithmus beruhen. Mit diesen Experimenten soll der Frage nachgegangen werden, wie sich Kalibrierungsfehler, Veränderungen der Kameraposition nach der Kalibrierung und Veränderung der Regelkonstanten auf den Ablauf der Regelung auswirken. Die erste große Anwendung der Regelung, nämlich die Regelung für die Objektklassifizierung, folgt in dem sich anschließenden fünften Kapitel. Im ersten Teil wird die Objektausrichtung zu einer bestimmten Kamera beschrieben, so daß die Grundfläche im Bild auf eine bestimmte Größe skaliert zu sehen ist. Dieser Teil gliedert sich auf in die Ermittlung des Sehstrahls der linken Kamera, die Ausrichtung der Hand zum Sehstrahl, die Bewegung der Hand mit

[3] In dieser Arbeit entspricht das Weltkoordinatensystem dem Roboterbasiskoordinatensystem. Dies wird in Kapitel zwei noch genauer beschrieben.

dem Objekt entlang des Sehstrahls zur Kamera, bis die Hand in einer bestimmten Größe zu sehen ist und die Extraktion der Kontur der Grundfläche. Eine weitere Komponente des ersten Teils ist die Regelung der Vergenz der rechten Kamera. Sie geschieht nach der Ermittlung des Sehstrahls der linken Kamera und ist ein Beispiel für ein aktives Sehen. Der zweiten Teil des fünften Kapitels beinhaltet zum einen die Repräsentation der Grundfläche, die über Autoregression approximiert wird, und zum anderen die Erkennung der Grundfläche, d.h. die Bestimmung, um welche Grundfläche es sich handelt. Dies geschieht mit Hilfe der Berechnung eines bestimmten Abstandsmaßes in dem durch die Autoregression entstandenen Vektorraum. Die zweite große Anwendung der Regelung ist in Kapitel sechs zu finden. Hier wird die visuell basierte Regelung für das Heranführen des Objektes an das für das Objekt bestimmte Loch beschrieben. Diese Regelung dient also der Objektmanipulation. Hierzu wird zuerst auf die Objekterkennung hinsichtlich des Quaders eingegangen, nämlich auf eine Extraktion der den Quader berandenden Segmente mittels orientierungselektiver Hough-Transformation, und danach auf die Erkennung eines bestimmten Berandungssegmentes mittels Orientierung, Länge und Grauwertnachbarschaft der Segmente. Der Mittelpunkt des erkannten Segmentes ist der Istwert für die Regelung. Als zweites wird die Objekterkennung hinsichtlich des Zylinders beschrieben, die zum einen eine Ellipsenextraktion, und zum anderen die Erkennung einer bestimmten Halbellipse mittels Grauwertnachbarschaft beinhaltet. Die Eingabe für den Regler ist dann ein Punkt auf der Halbellipse, der eine bestimmte Tangentensteigung besitzt. Als drittes wird die Extraktion des Lochbrettes und der Löcher beschrieben, sowie deren Erkennung. Ein ausgezeichneter Punkt des Loches bestimmt den Sollwert für die Regelung. Der letzte Abschnitt beinhaltet die eigentliche Regelung bei der Annäherung. In Kapitel sieben werden die Ergebnisse noch einmal diskutiert und ein Ausblick auf mögliche Erweiterungen und Verbesserungen des Systems gegeben.

Kapitel 2

Die Systemumgebung

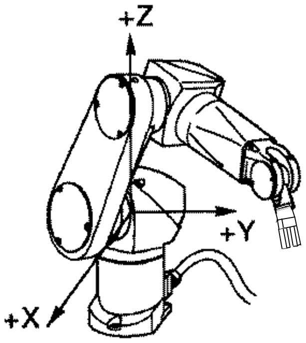
Dieses Kapitel soll sowohl einen Einblick in die technischen Systemkomponenten gewähren, die für diese Arbeit verwendet werden, als auch in den Versuchsaufbau. Die einzelnen Komponenten sind der Roboterarm, der die geregelte Annäherung des Objektes an das Ziel vornimmt, der Kamerakopf, der die Bildinformation liefert, und der Multithread-Prozessor, der die gegebenen Bildinformationen auswertet und die nächste Position an den Roboter liefert, zu der er sich bewegen soll. Die Aktionen von Bildaufnahme, Bildauswertung und Annäherung laufen hierbei zyklisch.

2.1 Der Roboterarm

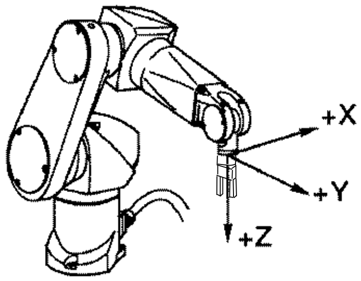
In dieser Arbeit wird der Roboterarm RX90 der Firma Stäubli verwendet. Er besteht aus sechs Gliedmaßen, die miteinander durch Gelenke verbunden sind. Jedes Gelenk stellt eine Achse dar, um die sich jeweils zwei Gliedmaße drehen. Die Greifhand ist an dem sechsten Gliedmaß befestigt. Sie besitzt 2 Finger,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

Die im Steuerschrank CS7 integrierte Software 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 zwei Koordinatensystemen, dem Basiskoordinatensystem W, das im folgenden unser Weltkoordinatensystem sein wird, und dem Werkzeugkoordinatensystem (Toolkoordinatensystem) T. Die Lage der Koordinatensysteme ist in Abbildung 2.1 dargestellt.

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 Werkzeugkoordinatensystems zum Roboterbasiskoordinatensystem in Millimeter (mm) und (yaw, $pitch$, $roll$) die *Orientierung* des Werkzeugkoordinatensystems relativ zum Roboterbasiskoordinatensystem in Grad. Diese Orientierung wird in der ZYZ-Eulernotation [Jain95] angegeben, wobei sich die Winkel nicht auf ein festes Koordinatensystem beziehen, denn die Drehung



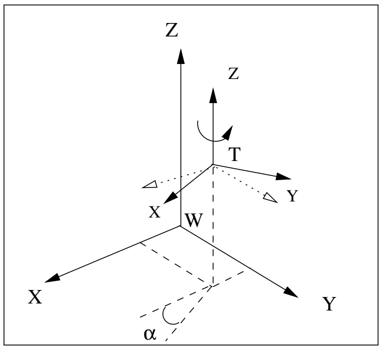
(a) Basissystem.



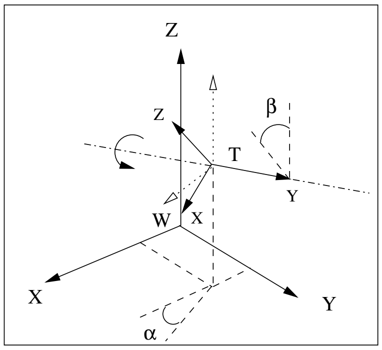
(b) Werkzeugkoordinatensystem.

Abbildung 2.1: Die Koordinatensysteme des Manipulators.

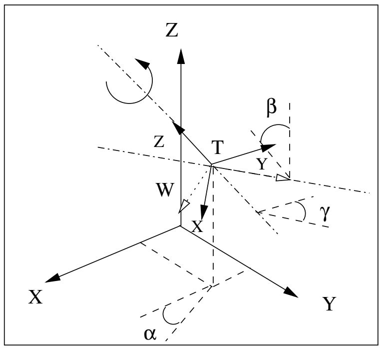
eines jeden Winkels wird bezüglich des durch die Drehung neu entstandenen Koordinatensystems ausgeführt. Abbildung 2.2 zeigt eine Drehung mit ZYZ- Eulerwinkeln.



(1.) Drehung um die Z-Achse des Toolkoordinatensystems um α (yaw).



(2.) Drehung um die neu entstandene Y-Achse des Toolkoordinatensystems um β (pitch).



(3.) Drehung um die neu entstandene Z-Achse des Toolkoordinatensystems um γ (roll).

Abbildung 2.2: Eine Drehung des Toolkoordinatensystems bezüglich des Basissystems in ZYZ- Eulerwinkeln $(yaw, pitch, roll)$.

Die Orientierung beträgt genau dann Null Grad in jeder Richtung, wenn die Hand senkrecht nach oben zeigt, und die Handfläche parallel zur YZ-Ebene des Basissystems ist (siehe Abb. 2.3). Der Ursprung des Werkzeugkoordinatensystems befindet sich üblicherweise in der Mitte des Werkzeugflansches, an dem die Greifhand befestigt ist. Die Schnittstelle wurde von uns aber so erweitert, daß sich aus dem Programm heraus auch ein anderer Ursprung, wie etwa der Mittelpunkt zwischen den Fingerspitzen, definieren läßt. Ebenso wurde die Schnittstelle derart erweitert, daß sich jetzt eine Translation und Rotation relativ zum

Werkzeugkoordinatensystem erreichen läßt.

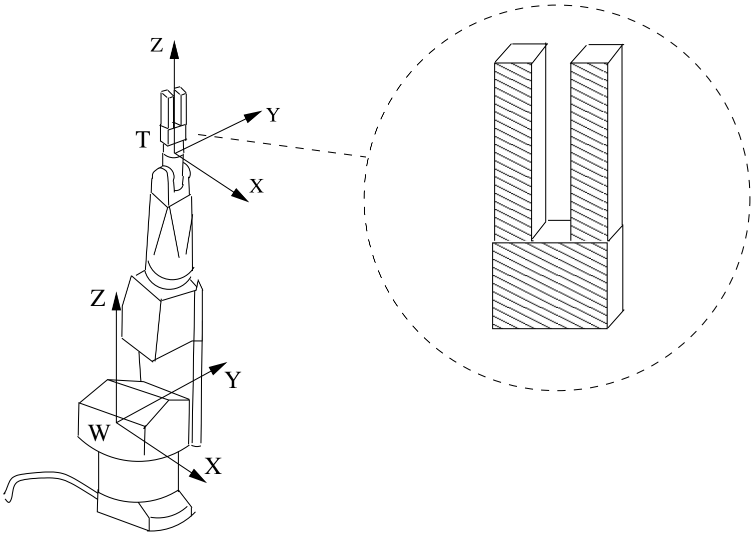


Abbildung 2.3: Eine Orientierung des Toolkoordinatensystems von Null Grad in jeder Richtung zum Basissystem. Die Hand zeigt senkrecht nach oben, und die Handfläche, die dem schraffierten Teil in der Vergrößerung entspricht, ist parallel zur YZ-Ebene des Basissystems.

2.2 Der Kamerakopf

Das bildgebende System ist eine binokulare Kameraplattform. Sie besitzt vier *mechanische* Freiheitsgrade, beschrieben durch den Schwenkwinkel χ ("pan") mit den maximalen Werten von $-160°$ bis $160°$, den Neigungswinkel ϕ ("tilt"), der sich von $-90°$ bis $90°$ einstellen läßt, und die Vergenzwinkel der Kameras θ_l und θ_r, die von $-45°$ bis $45°$ anzusteuern sind. Die einzelnen Freiheitsgrade zeigt noch einmal Abbildung 2.4. Als *optische* Freiheitsgrade sind der Zoom und der Fokus in einem geschlossenen Regelkreislauf variabel einstellbar. Der Zoom hat eine variable Brennweite von 11,5 mm bis 69 mm. Der Fokus ist einstellbar von 1 m bis ∞. Die Iris ist in einem offen Regelkreislauf veränderbar, d.h. es ist zwar kein bestimmter Wert für die Blende anlaufbar, sie kann sich aber relativ zum derzeitigen Wert weiter öffnen oder schließen lassen. Jede der beiden Kameras besitzt einen CCD-Chip, der Grauwertbilder der Größe 768x576 Pixel liefert. Der Abstand der optischen Zentren der beiden Kameras ist $\hat{b} = 25\ cm$. Gesteuert wird das ganze System durch die Motorensteuerkarte PMAC, die die Steuerbefehle der SUN-Workstation in direkte Steuerbefehle umsetzt. In unserer Anwendung dient der Kopf hauptsächlich als starre Kamera. Innerhalb der Kalibrierungsphase wird jedoch der rechte Vergenzwinkel aktiv geregelt, um die Objekterkennung für die Annäherung zu vereinfachen. Hierauf wird in Kapitel fünf näher eingegangen.

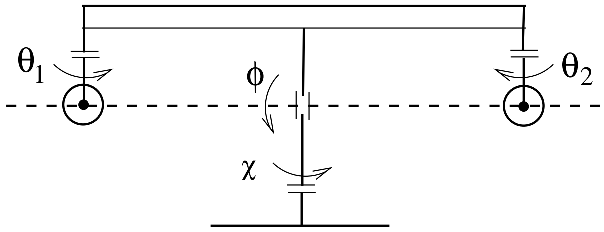


Abbildung 2.4: Die vier mechanischen Freiheitsgrade der Kameraplattform.

2.3 Das Multithread-Prozessorsystem

Als Schnittstelle zwischen Kamerakopf und Roboterarm dient eine Ultra Enterprise 4000 der Firma SUN, die mit 4 Prozessoren ausgestattet ist. Der Rechner erhält in jedem Annäherungszyklus die Bilder des Kamerakopfes, verarbeitet sie, berechnet den Positionierungsbefehl für den Roboter und leitet ihn dann über den Steuerschrank CS7 an den Roboterarm weiter, der ihn dann ausführt. Der Umstand, daß die Ultra Enterprise mit 4 Prozessoren ausgestattet ist, ermöglicht es uns, beide Bilder des Kamerakopfes parallel zu verarbeiten, um den Punkt zu bestimmen, den der Roboter anfahren soll. Dies bringt einen beachtlichen Zeitgewinn. Die von uns entwickelten Funktionen für die Bildverarbeitung sind eingebettet in das Bildverarbeitungssystem KHOROS, das auf dem Rechner installiert ist. Abbildung 2.5 zeigt ein Übersichtsschema über die Verbindung der einzelnen Einheiten.

Nach dieser Einführung in die verwendete Hardware folgt im nächsten Abschnitt eine ausführliche Beschreibung des Versuchsaufbaus bzw. -ablaufs, der im ersten Kapitel schon kurz skizziert wurde.

2.4 Der Versuchsaufbau bzw. -ablauf

Für die folgenden zeitlich aufeinanderfolgenden Phasen wird angenommen, daß der zum Ursprung des Roboterbasissystems erhöhte Kamerakopf die Szene aus ca. 1 m Entfernung beobachtet. Er hat zunächst fest vorgegebene Schwenk-, Neigungs- und Vergenzwinkel, um die gesamte Szene in beiden Kameras im Blickfeld zu haben. In der *Kalibrierungsphase* befindet sich eine Leuchtdiode in der Hand des Roboters. Im ersten Modul dieser Phase wird nun der Sehstrahl der linken Kamera über eine visuell basierte Regelung ermittelt. Dieses erreicht man, indem man den auf die Bildebene projizierten Lichtpunkt zur Bildmitte hin regelt. Der Roboter bewegt sich hierbei auf einer zur YZ-Ebene des Weltkoordinatensystems parallelen Ebene. Diese Regelung wird noch einmal wiederholt, wobei sich der Roboter auf einer zur ersten Ebene in X-Richtung verschobenen parallelen Ebene bewegt. Auf diese Weise erhält man zwei 3D Punkte, die auf dem Sehstrahl der linken Kamera liegen, so daß sich eine dem Sehstrahl kollineare Gerade beschreiben läßt.

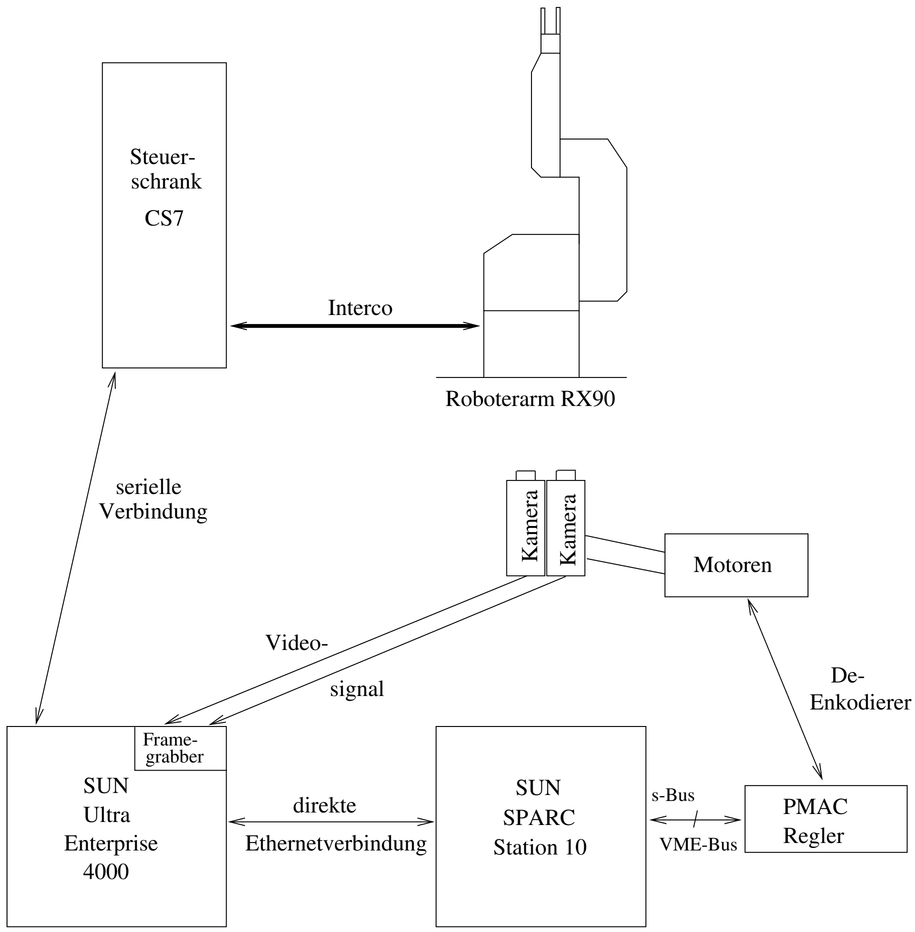
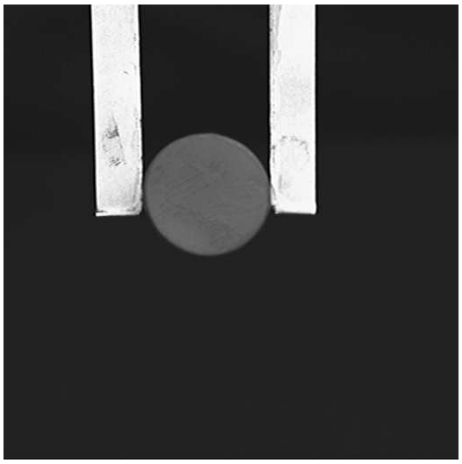


Abbildung 2.5: Übersichtsschema der Systemeinheiten und ihrer Verbindungen.

Mit Hilfe der Winkel, die diese Gerade gegenüber den Ebenen des Weltkoordinatensystems besitzt, läßt sich die Ausrichtung der Hand parallel zur Bildebene der linken Kamera ermitteln (vgl. Kap. 5.1). Das zweite Modul regelt die Vergenz der rechten Kamera, bis sich der auf die rechte Bildebene abgebildete Lichtpunkt im Mittelpunkt des Bildes befindet. Somit ist ein bestimmter Punkt im Weltkoordinatensystem durch die Kameras fixiert. Dies soll später der Startpunkt für die Annäherung des Objektes an das Lochbrett sein. Auf diese Weise läßt sich der Suchbereich für die erste Suche nach dem Objekt genau angeben. Durch das Abfahren des Roboterarmes von verschiedenen festgelegten Punkten wird im dritten Modul dieser Phase das Verhältnis zwischen dem von uns verwendeten Weltkoordinatensystem (hier das Roboterbasiskoordinatensystem) und dem Kamerakoordinatensystem gebildet (vgl. Kap. 3). Dies geschieht, indem man sich sowohl die abgefahrenen 3D Punkte, als auch die jeweils projizierten 2D Bildpunkte merkt und über eine Ausgleichsrechnung die Projektionsmatrizen ermittelt. Diese Projektionsmatrizen werden für die Regelung benötigt[1]. Für die nächste Phase, die *Klassifikationsphase*, wird angenommen, daß das Objekt schon gegriffen wurde. Das Greifen des Objektes wurde nicht noch zusätzlich verwirklicht, da dies den Rahmen dieser Diplomarbeit sprengen würde. Das Greifen läßt sich aber realisieren durch eine Modifikation der Diplomarbeit von [Tröster97] oder der Studienarbeit von [Kunze96]. Im Ausblick des siebten Kapitels wird außerdem ein Greifvorgang durch Regelung vorgestellt. Es befindet sich also eines der beiden Versuchsobjekte - d.h. entweder der Quader oder der Zylinder - in der Hand des Roboters, so daß die Grundfläche des Objektes parallel zur Handfläche des Roboters ausgerichtet ist. Ein Lochbrett liegt parallel zur XY-Ebene des Weltkoordinatensystems im Wirkungsbereich des Roboters vor einem homogenen Hintergrund. Die Hand befindet sich auf der zur Bildebene parallelen Position, die in der Kalibrierungsphase ermittelt wurde. Der Zoom der Kamera wird auf das Maximum gestellt, so daß nur die Hand mit dem Objekt in dem Sichtbereich der linken Kamera liegt. Um die Grundfläche des Objektes erkennen und klassifizieren zu können, müssen in dieser Phase die Finger im Bild gelöscht werden, damit nur noch die Grundfläche zu sehen ist. Hierzu lassen wir die Hand durch eine Regelung auf der zur Bildebene parallelen Ebene rotieren, bis die Finger der Hand senkrecht im Bild liegen. Die Hand mit dem Objekt wird entlang des Sehstrahls der linken Kamera zur selbigen hingeregelt, bis eine bestimmte Skalierung der Finger vorliegt (siehe Abb. 2.6).

(a) Annäherung des Roboterarmes an die linke Kamera.



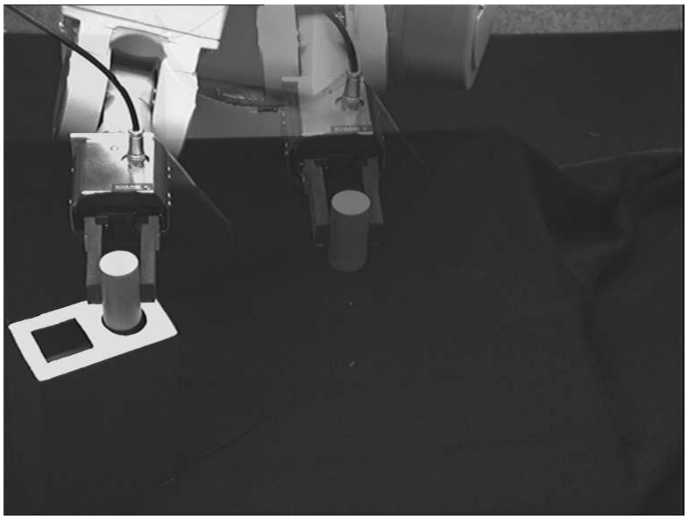
(b) Sicht der linken Kamera.

Abbildung 2.6: Teil der *Klassifikationsphase*. Links: Der Roboterarm nähert sich entlang des Sehstrahls der linken Kamera dem Kamerakopf an für eine bessere Objektansicht. Rechts: Sicht der linken Kamera auf die Grundfläche des Objektes nach der Annäherung.

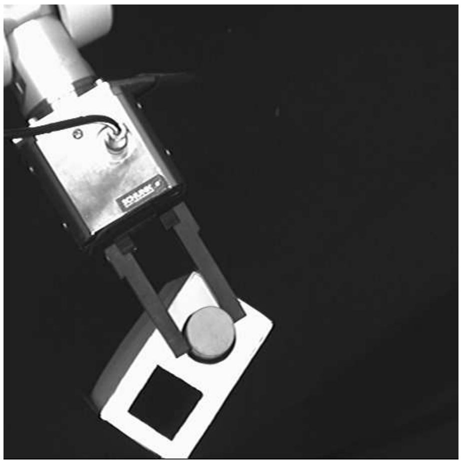
An dieser Stelle werden die Finger vom Objekt segmentiert, und das Objekt wird über die

[1]Auch das erste Modul dieser Phase benötigt schon Projektionsmatrizen. Die Regelung für die Ermittlung des Sehstrahls ist aber trotz sehr ungenauer Projektionsmatrizen so robust, daß hierfür eine beliebige Projektionsmatrix aus einer vorherigen Kalibrierung benutzt werden kann. Aus dieser Tatsache begründet sich die Anwendung der Regelung vor der eigentlichen Kalibrierung.

Grundfläche erkannt (vgl. Kap. 5.2). Für die *Annäherungsphase* zoomt die Kamera wieder weg, so daß die gesamte Szene wieder im Blickfeld des Kopfes ist. Für das erste Modul dieser Phase, die die Lochbretterkennung beinhaltet, fährt der Arm das Objekt aus dem Bild heraus, um die Erkennung nicht zu stören. Bei der Lochbretterkennung werden sowohl das Brett selber als auch die einzelnen Löcher erkannt. Zusätzlich wird der Zielpunkt am Loch bestimmt, zu dem der Arm mit dem Objekt hingeregelt werden soll. Anschließend fährt der Arm das Objekt wieder in das Blickfeld des Kopfes (vgl. Kap. 6.3). Das Objekt wird so gedreht, bis sich die Hauptachse des Objektes senkrecht zur XY-Ebene des Weltkoordinatensystems befindet, damit die Grundfläche des Objektes parallel zur Lochbrettfläche und damit auch zum Loch selber ist. In einem geschlossenen, sich immer wiederholenden Zyklus von Bildaufnahme, Merkmalsextraktion und Roboterbewegung wird das Objekt in dem anschließenden Modul zum Loch hingeregelt, wobei die Objekterkennung für die Merkmalsextraktion mit Hilfe der Hough-Transformation und Ellipsenapproximation geschieht. Zum Schluß befindet sich das Objekt direkt oberhalb des Loches, so daß es nur noch senkrecht nach unten eingeführt werden muß (siehe Abb. 2.7).



(a) Annäherung des Objektes an das Loch.



(b) Sicht von oben nach Annäherung.

Abbildung 2.7: *Annäherungsphase.* Links: Der Roboterarm nähert sich mit dem Objekt dem für das Objekt bestimmten Loch an (Sicht des linken Kamera, Überblendung vor und nach der Annäherung). Rechts: Sicht von oben nach der geregelten Annäherung. Das Objekt braucht zum Einfügen in das Loch lediglich senkrecht nach unten geführt werden.

Kapitel 3

Kalibrierung

3.1 Transformationen

Um das Verhältnis zwischen dem 3D Weltkoordinatensystem W und einem Bildkoordinatensystem K zu beschreiben, läßt sich folgende Projektionsgleichung angeben [Faugeras93]:

$$\begin{pmatrix} \lambda x^K \\ \lambda y^K \\ \lambda \end{pmatrix} = \begin{pmatrix} P_{11} & \cdots & P_{14} \\ P_{21} & \cdots & P_{24} \\ P_{31} & \cdots & P_{34} \end{pmatrix} \cdot \begin{pmatrix} X^W \\ Y^W \\ Z^W \\ 1 \end{pmatrix}. \tag{3.1}$$

Der Vektor $(\lambda x^K, \lambda y^K, \lambda)$ beschreibt hierbei den auf die Bildebene projizierten 2D Punkt in projektiven Koordinaten, und $(X^W, Y^W, Z^W, 1)$ den 3D Punkt im Weltkoordinatensystem in homogenen Koordinaten in mm, λ ist ein Skalar ungleich Null. Den 2D Bildpunkt (x^k, y^k) in Pixel erhält man entsprechend durch Division der ersten und zweiten Vektorkomponente mit der dritten Vektorkomponente λ.

Die Kamera vollführt also eine projektive Transformation vom projektiven Raum $\mathcal{P}^3$ in die projektive Ebene $\mathcal{P}^2$. Die Kamera wird hier als Lochkamera aufgefaßt. Die Matrix $\boldsymbol{P}$ beinhaltet implizit die extrinsischen und intrinsischen Parameter. Hierbei beschreiben die extrinsischen Parameter die Transformation des Punktes vom Weltkoordinatensystem zum Kamerakoordinatensystem, und die intrinsischen Parameter bestimmen die Projektion auf die Bildebene. Da wir die explizite Errechnung der Parameter in dieser Anwendung nicht benötigen, soll darauf nicht weiter eingegangen werden. Weil wir zwei Kameras benutzen und somit auch zwei verschiedene Bildkoordinatensysteme haben, benötigen wir zwei Projektionsmatrizen $\boldsymbol{P}^l$ bzw. $\boldsymbol{P}^r$, um das Verhältnis zwischen dem 3D Weltpunkt und dem 2D Punkt im linken Bildkoordinatensystem, bzw. dem 2D Punkt im rechten Bildkoordinatensystem zu erhalten. Ein von uns angewendetes Verfahren, um die Projektionsmatrizen zu erhalten, ist folgendes: In einem abgedunkelten Raum fährt der Roboterarm, der eine

Leuchtdiode in der Hand hält, n Punkte ab, die möglichst gut im Raum verteilt sind, und merkt sich die 3D Weltkoordinaten der Leuchtdiode. Gleichzeitig wird jeweils in beiden Bildern der mittlere hellste Punkt (also der aufs Bild projizierte Leuchtdiodenpunkt) durch geometrische Schwerpunktbildung aller hellsten Punkte ermittelt und auch gespeichert[1]. Nach der beendeten Prozedur hat man also n 3D Punkte (X_i^W, Y_i^W, Z_i^W) und die zugehörigen 2D Projektionen $(x_i^{K^l}, y_i^{K^l})$ auf die linke bzw. $(x_i^{K^r}, y_i^{K^r})$ auf die rechte Kamera. Im folgenden wird nur die Ermittlung der Projektionmatrix der linken Kamera $\boldsymbol{P^l}$ beschrieben. Die Ermittlung der Projektionsmatrix der rechten Kamera $\boldsymbol{P^r}$ ergibt sich genauso, man muß nur (x^{K^l}, y^{K^l}) jeweils durch (x^{K^r}, y^{K^r}) ersetzen. Setzt man jeden 3D Punkt (X_i^W, Y_i^W, Z_i^W) und den zugehörigen 2D Punkt $(x_i^{K^l}, y_i^{K^l})$ in das Gleichungssystem (3.1) ein und formt dieses um, erhält man n Gleichungen der Form:

$$\left(P_{11}^l P_{12}^l P_{13}^l\right) \cdot \begin{pmatrix} X_i^W \\ Y_i^W \\ Z_i^W \end{pmatrix} - x_i^{K^l} \left(P_{31}^l P_{32}^l P_{33}^l\right) \cdot \begin{pmatrix} X_i^W \\ Y_i^W \\ Z_i^W \end{pmatrix} + P_{14}^l - x^{K^l} P_{34}^l = 0, \quad i = 1, \cdots, n, \tag{3.2}$$

und n Gleichungen der Form:

$$\left(P_{21}^l P_{22}^l P_{23}^l\right) \cdot \begin{pmatrix} X_i^W \\ Y_i^W \\ Z_i^W \end{pmatrix} - y_i^{K^l} \left(P_{31}^l P_{32}^l P_{33}^l\right) \cdot \begin{pmatrix} X_i^W \\ Y_i^W \\ Z_i^W \end{pmatrix} + P_{24}^l - y^{K^l} P_{34}^l = 0, \quad i = 1, \cdots, n. \tag{3.3}$$

Da $\boldsymbol{P^l}$ nicht Null sein darf (dies wäre perspektivisch nicht sinnvoll), läßt sich folgende Einschränkung machen, um diese Bedingung zu erfüllen:

$$P_{34}^l = 1. \tag{3.4}$$

Unter dieser Annahme läßt sich aus den $2 \cdot n$ Gleichungen (3.2) bzw. (3.3) folgendes Gleichungsystem im Matrixform aufstellen:

$$\begin{pmatrix} X_1^W & Y_1^W & Z_1^W & 1 & 0 & 0 & 0 & 0 & -x_1^{K^l} X_1^W & -x_1^{K^l} Y_1^W & -x_1^{K^l} Z_1^W \\ 0 & 0 & 0 & 0 & X_1^W & Y_1^W & Z_1^W & 1 & -y_1^{K^l} X_1^W & -y_1^{K^l} Y_1^W & -y_1^{K^l} Z_1^W \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n^W & Y_n^W & Z_n^W & 1 & 0 & 0 & 0 & 0 & -x_n^{K^l} X_n^W & -x_n^{K^l} Y_n^W & -x_n^{K^l} Z_n^W \\ 0 & 0 & 0 & 0 & X_n^W & Y_n^W & Z_n^W & 1 & -y_n^{K^l} X_n^W & -y_n^{K^l} Y_n^W & -y_n^{K^l} Z_n^W \end{pmatrix} \cdot \begin{pmatrix} P_{11}^l \\ \vdots \\ P_{14}^l \\ \vdots \\ P_{32}^l \\ P_{33}^l \end{pmatrix} = \begin{pmatrix} x_1^{K^l} \\ y_1^{K^l} \\ \vdots \\ x_n^{K^l} \\ y_n^{K^l} \end{pmatrix}. \tag{3.5}$$

[1]Auf eine geeignete Anzahl n der abzufahrenden Punkte wird in Kapitel 4 eingegangen.

Unter Verwendung z.B. des Householderverfahrens läßt sich dieses Gleichungssystem lösen, und die Projektionsmatrix ermitteln. An dieser Stelle sei noch einmal darauf hingewiesen, daß sich die Projektionsmatrix $\boldsymbol{P^r}$ für die rechte Kamera auf dieselbe Weise errechnen läßt, indem man statt der n 2D Punkte $(x_i^{K^l}, y_i^{K^l}), i = 1, \cdots, n$ die zugehörigen n 2D Punkte $(x_i^{K^r}, y_i^{K^r}), i = 1, \cdots, n$ der rechten Kamera in das obige Gleichungssystem einsetzt.

Hat man beide Projektionsmatrizen errechnet, so läßt sich für jeden beliebigen 3D Punkt im Weltkoordinatensystem der zugehörige 2D Punkt im linken bzw. im rechten Bildkoordinatensystem errechnen. Will man aus den beiden gegebenen 2D Punkten der Bildkoordinatensysteme den zugehörigen 3D Weltkoordinatenpunkt erhalten, läßt sich unter Verwendung von $\boldsymbol{P^l}$ und $\boldsymbol{P^r}$ das Gleichungssystem (3.1) umformen zu:

$$\begin{pmatrix} -P_{14}^l \\ -P_{24}^l \\ -P_{34}^l \\ -P_{14}^r \\ -P_{24}^r \\ -P_{34}^r \end{pmatrix} = \begin{pmatrix} P_{11}^l & P_{12}^l & P_{13}^l & -x^{K^l} & 0 \\ P_{21}^l & P_{22}^l & P_{33}^l & -y^{K^l} & 0 \\ P_{31}^l & P_{32}^l & P_{33}^l & -1 & 0 \\ P_{11}^r & P_{12}^r & P_{13}^r & 0 & -x^{K^r} \\ P_{21}^r & P_{22}^r & P_{23}^r & 0 & -y^{K^r} \\ P_{31}^r & P_{32}^r & P_{33}^r & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} X^W \\ Y^W \\ Z^W \\ \lambda_1 \\ \lambda_2 \end{pmatrix}. \tag{3.6}$$

Durch Einsetzen der 2D Punkte (x^{K^l}, y^{K^l}) bzw. (x^{K^r}, y^{K^r}) und der Projektionsmatrizen $\boldsymbol{P^l}$ und $\boldsymbol{P^r}$ erhält man durch Lösung dieses Gleichungssystems den gesuchten 3D Weltkoordinatenpunkt.

3.2 Die Jakobimatrix

Durch die oben angegebenen Transformationen läßt sich nun aus einem gegebenen 3D Punkt $\boldsymbol{p^W} = (X^W, Y^W, Z^W)^T$ im Weltkoordinatensystem der jeweils zugehörige 2D Punkt im linken und rechten Bildkoordinatensystem angeben, und bei gegebenen 2D Punkten im linken und rechten Bildkoordinatensystem der zugehörige 3D Punkt. Wie aber läßt sich die resultierende *Änderung* eines 2D Punktes bei Änderung des 3D Weltkoordinatenpunktes ausdrücken? Die Antwort liefert die Jakobimatrix. Mit ihr läßt sich die gewünschte Änderung beschreiben. Es gilt nämlich:

$$\begin{pmatrix} dx^K \\ dy^K \end{pmatrix} = \boldsymbol{J_P}(\boldsymbol{p^W}) \cdot \begin{pmatrix} dX^W \\ dY^W \\ dZ^W \end{pmatrix}. \tag{3.7}$$

Hierbei beschreibt $(dx^K, dx^K)^T$ die Änderung in dem 2D Bildkoordinatensystem, und $(dX^W, dY^W, dZ^W)^T$ die Änderung im 3D Weltkoordinatensystem. Betrachtet man Gleichung (3.1) und formt diese um zu:

$$x^K = \frac{(P_{11}P_{12}P_{13}P_{14}) \cdot \begin{pmatrix} \boldsymbol{p^W} \\ 1 \end{pmatrix}}{(P_{31}P_{32}P_{33}P_{34}) \cdot \begin{pmatrix} \boldsymbol{p^W} \\ 1 \end{pmatrix}} \tag{3.8}$$

und zu:

$$y^K = \frac{(P_{21}P_{22}P_{23}P_{24}) \cdot \begin{pmatrix} \boldsymbol{p^W} \\ 1 \end{pmatrix}}{(P_{31}P_{32}P_{33}P_{34}) \cdot \begin{pmatrix} \boldsymbol{p^W} \\ 1 \end{pmatrix}}, \tag{3.9}$$

läßt sich die Jakobimatrix explizit über die Projektionsmatrizen angeben:

$$\begin{aligned} \boldsymbol{J_P}(\boldsymbol{p^W}) &= \begin{pmatrix} \frac{\partial x^K}{\partial X^W} & \frac{\partial x^K}{\partial Y^W} & \frac{\partial x^K}{\partial Z^W} \\ \frac{\partial y^K}{\partial X^W} & \frac{\partial y^K}{\partial Y^W} & \frac{\partial y^K}{\partial Z^W} \end{pmatrix} \\ &= \begin{pmatrix} \frac{P_{11} \cdot \left((P_{31}P_{32}P_{33}P_{34}) \cdot \begin{pmatrix} \boldsymbol{p^W} \\ 1 \end{pmatrix} \right) - P_{31} \cdot \left((P_{21}P_{22}P_{23}P_{24}) \cdot \begin{pmatrix} \boldsymbol{p^W} \\ 1 \end{pmatrix} \right)}{\left((P_{31}P_{32}P_{33}P_{34}) \cdot \begin{pmatrix} \boldsymbol{p^W} \\ 1 \end{pmatrix} \right)^2} & \cdots \\ \vdots & \ddots \end{pmatrix}. \end{aligned} \tag{3.10}$$

Da wir im folgenden einen 3D Punkt immer mit zwei Kameras beobachten werden, ist es sinnvoll, die Änderung im linken und rechten Bildkoordinatensystem als einen Vektor zu betrachten. Die Gleichung (3.7) läßt sich dann wie folgt erweitern:

$$\begin{pmatrix} dx^{K^l} \\ dy^{K^l} \\ dx^{K^r} \\ dy^{K^r} \end{pmatrix} = \boldsymbol{J_{P^{l,r}}}(\boldsymbol{p^W}) \cdot \begin{pmatrix} dX^W \\ dY^W \\ dZ^W \end{pmatrix}. \tag{3.11}$$

Die zugehörige Jakobimatrix ist jetzt eine 4x3 Matrix:

$$\boldsymbol{J_{P^{l,r}}}(\boldsymbol{p^W}) = \begin{pmatrix} \frac{\partial x^{K^l}}{\partial X^W} & \frac{\partial x^{K^l}}{\partial Y^W} & \frac{\partial x^{K^l}}{\partial Z^W} \\ \frac{\partial y^{K^l}}{\partial X^W} & \frac{\partial y^{K^l}}{\partial Y^W} & \frac{\partial y^{K^l}}{\partial Z^W} \\ \frac{\partial x^{K^r}}{\partial X^W} & \frac{\partial x^{K^r}}{\partial Y^W} & \frac{\partial x^{K^r}}{\partial Z^W} \\ \frac{\partial y^{K^r}}{\partial X^W} & \frac{\partial y^{K^r}}{\partial Y^W} & \frac{\partial y^{K^r}}{\partial Z^W} \end{pmatrix}. \tag{3.12}$$

Will man nun aus den gemessenen Veränderungen in dem linken und rechten Bildkoordinatensystem auf die Veränderung im Weltkoordinatensystem schließen, so läßt sich Gleichung

(3.11) folgendermaßen umformen:

$$\begin{pmatrix} dX^W \\ dY^W \\ dZ^W \end{pmatrix} = (\boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W})^T \cdot \boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W}))^{-1} \cdot \boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W})^T \cdot \begin{pmatrix} dx^{K^l} \\ dy^{K^l} \\ dx^{K^r} \\ dy^{K^r} \end{pmatrix}. \tag{3.13}$$

Da sich aus der 3x4 Matrix $\boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W})$ keine direkte inverse Matrix bilden läßt, benutzt man hierbei das Prinzip der Pseudoinversen [Stoer89]. Die 4x4 Matrix $(\boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W})^T \cdot \boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W}))^{-1} \cdot \boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W})^T$ ist hierbei die aus der 3x4 Matrix $\boldsymbol{J}_{\boldsymbol{P}^{l,r}}(\boldsymbol{p^W})$ gebildete Pseudoinverse. Mit Hilfe der in diesem Kapitel eingeführten Transformationen läßt sich nun ein Regelalgorithmus entwerfen, der auf Veränderungen in den zwei Bildebenen reagiert und einen Ausgleich zu dieser Veränderung durch eine geeignete Veränderung im Weltkoordinatensystem schafft. Der Entwurf des Reglers sowie das Prinzip des Reglers wird im folgenden Kapitel beschrieben.

Kapitel 4

Regelungstechnik

Dieses Kapitel soll sowohl die Regelungstechnik allgemein, als auch den von uns angewandten Regler beschreiben. Hierzu wird im ersten Abschnitt eine Einführung in die grundlegenden Begriffe der Regelungstechnik gegeben. Im zweiten Abschnitt werden verschiedene Regelungsarten angegeben, mit deren Hilfe wir dann im dritten Abschnitt einen Regler entwerfen, der das grundlegende Prinzip des von uns verwendeten Reglers beinhaltet.

4.1 Grundlegende Begriffe der Regelungstechnik

Das Regelungsmodell läßt sich einbetten in das übergeordnete Modell der *dynamischen Systeme* [Soltyasik89]. Ein dynamisches System ist zunächst einmal ein *Transformationssystem*, das einen Transformationsinput, wie Materie, Energie und Informationen aus der Umwelt, zu einen Transformationsoutput verarbeitet. Durch Störinputs wird der Prozeß im Transformationssystem beeinträchtigt. Damit ein gewünschter Output hervorgeht, wird der Transformationsprozeß durch ein *Führungssystem* zielorientiert beeinflußt. Die *Regelung* ist ein spezielles Führungssystem, das auf dem Prinzip der Rückkopplung basiert. Dies bedeutet, daß der Output einer Transformation wieder auf dessen Input wirkt. Dies ist die typische Situation eines geschlossenen Wirkungskreises. Das zum Führungssystem *Regler* zugehörige Transformationssystem wird *Regelstrecke* genannt. Die gewünschten Ausgabewerte heißen *Sollwerte*, und die tatsächlichen Ausgabewerte, also die Outputgrößen des Transformationssystems heißen *Istwerte*. Die Abbildung 4.1 zeigt ein Schema eines Regelungsmodells.

Da auch auf die Regelstrecke Störgrößen einwirken können, soll der Regler diesen Störgrößen entgegenwirken, indem er die Istwerte den Sollwerten fortwährend annähert. Dies geschieht durch den *Soll-Ist-Vergleich*. Durch ihn werden die Regelabweichungen ermittelt, und sogenannte *Stellgrößen* vom Regler vorgegeben. Der Regler sorgt also bei einer festgestellten Regelabweichung mit einer Veränderung der Stellgröße dafür, daß die Regelgrößen trotz einwirkender Störungen auf den Sollwert gebracht werden. Zur Einstellung der Regelgrößen

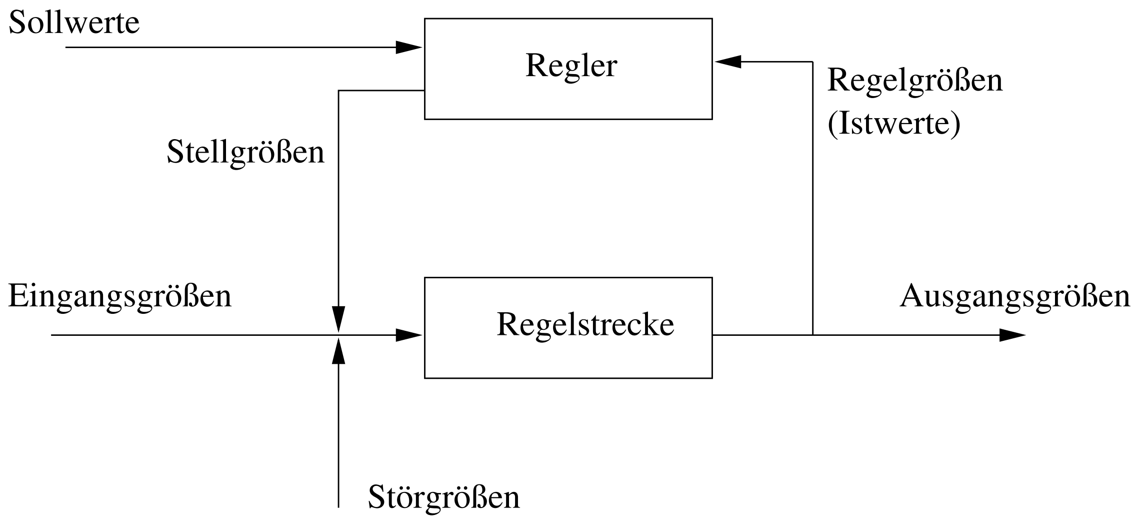


Abbildung 4.1: Schema eines Regelungsmodelles.

haben sich in der Praxis verschiedene Reglerarten bewährt, die im nächsten Abschnitt beschrieben werden.

4.2 Reglerarten

In diesem Abschnitt werden zunächst die drei Grundtypen, die sogenannten "reinen" Regler vorgestellt, nämlich der P-, der I- und der D-Regler. Aus diesen reinen Regler lassen sich Kombinationen bilden, die die positiven Eigenschaften der einzelnen reinen Regler hervorheben sollen. Um die Regler und ihre Eigenschaften zu erläutern, wird im folgenden ein System beschrieben, auf das der einzelne Regler einwirken soll.

Sei zunächst einmal das System im Ruhezustand, d.h der Sollwert SW ist gleich dem Istwert $q(t_0) = q_0$. Die eingestellte Stellgröße sei hierbei $u(t_0) = u_0$. Die Variable t_0 soll hierbei einen bestimmten Zeitpunkt andeuten. Durch eine Störung wird der Istwert q_0 zu $q(t) = q_a$ verändert. Gesucht ist nun eine geeignete Stellgröße u_a, um den aktuellen Istwert q_a wieder zum idealen Istwert q_0 - sprich dem Sollwert SW - zu verändern. Durch den Soll-Ist-Vergleich erhält man die *Regelabweichung*

$$e_q(t) = q_0 - q(t). \tag{4.1}$$

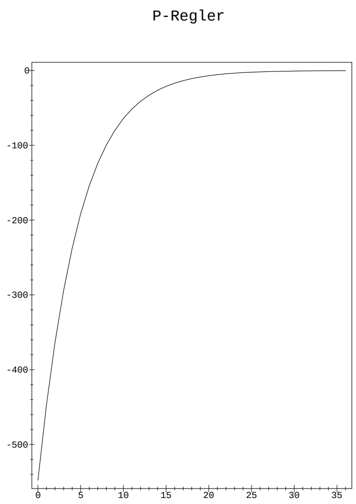
Die derzeitige Veränderung in der Stellgröße ist

$$e_u(t) = u_0 - u(t). \tag{4.2}$$

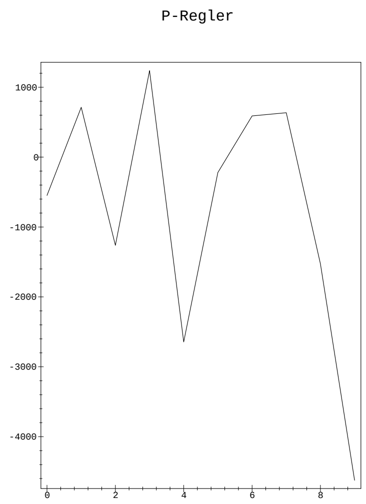
Die Regelungsvorschrift für den reinen Proportional-Regler (P-Regler) lautet

$$e_u^P(t) = \mathcal{K}_P \cdot e_q(t). \tag{4.3}$$

Er wird Proportionalregler genannt, weil die Veränderung der Stellgröße $(u_0 - u(t))$ proportional zur Regelabweichung $(q_0 - q(t))$ eingestellt wird. Der Proportionalitätsfaktor ist hierbei $\mathcal{K}_P$. Man bezeichnet $\mathcal{K}_P$ auch als Übertragungsfaktor des Reglers. $\mathcal{K}_P$ ist die bezeichnende Größe für den P-Regler. Bei kleiner $\mathcal{K}_P$ Konstante nähert sich der Fehler asymptotisch gegen Null (siehe Abb. 4.2(a)). Eine auftretende Störung kann um so schneller ausgeregelt werden, je höher der Faktor ist. Hierbei ist aber zu beachten, daß die Sollgröße nie ganz erreicht wird. Ein weiteres Problem ist, daß ab einer bestimmten Größe des Proportionalitätsfaktors Schwingungen entstehen, die sich aufschaukeln. Die Amplitude wächst über alle Grenzen. Die Regelung ist dann also instabil. Dies zeigt Abbildung 4.2(b).



(a) Ergebnis eines P-Reglers mit kleiner $\mathcal{K}_P$ Konstante.



(b) Ergebnis eines P-Reglers mit zu starker $\mathcal{K}_P$ Konstante.

Abbildung 4.2: Ergebnisse eines P-Regelalgorithmus. Die vertikale Achse beschreibt den auszuregelnden Fehler, die horizontale Achse den Zeitpunkt t. In Abb. (a) sieht man die asymptotische Annäherung des Fehlers an 0, in Abb. (b) das Aufschaukeln der Amplitude.

Ein Regler, der diese Probleme nicht besitzt ist der Integral-Regler (I-Regler). Seine Regelungsvorschrift lautet:

$$e_u^I(t) = \mathcal{K}_I \cdot \int_0^t e_q(t)dt. \tag{4.4}$$

Sein Name ist aus der Definition des Algorithmus abgeleitet. Der Proportionalitätsfaktor $\mathcal{K}_I$ wird als Integralanteil bezeichnet. Die Besonderheit beim I-Regler ist, daß er aufgrund des Integrales nicht nur den derzeitigen Wert der Regelgröße sondern auch alle bis zu diesem Zeitpunkt vorangegangenen Werte betrachtet. Dies läßt sich als eine Art von Speicherwirkung beschreiben, denn alle vorangegangenen Werte werden unter Berücksichtigung des

Vorzeichens aufintegriert bzw. - wenn man diskrete Zeitintervalle betrachtet - aufaddiert. Der I-Regler besitzt den Vorteil gegenüber dem P-Regler, daß er eine Regelabweichung vollständig beseitigen kann, er besitzt aber den Nachteil, daß er langsamer ist. Je nachdem wie sein Verhalten gegenüber der Regelstrecke ist, kann auch er zu Instabilitäten in der Regelung führen.

Der letzte reine Regler, der hier vorgestellt werden soll, ist der Differentialregler (D-Regler). Die Gleichung dieses Reglers ist:

$$e_u^D(t) = \mathcal{K}_D \cdot \frac{de_q(t)}{dt}. \tag{4.5}$$

Bei einem D-Regler wird die Änderungsgeschwindigkeit der Regelabweichung in eine dazu proportionale Strecke umgesetzt. Dies bedeutet, daß der Ausgang dieses Reglers gleich Null ist, wenn sich die Regelabweichung nicht ändert. Der D-Regler ist allein nicht in der Lage, den Regelkreis stabil zu halten. Deswegen tritt er in der Praxis immer in Kombination mit den anderen Reglern auf. Der PID-Regler ist eine Kombination der drei vorgestellten Regler. Er vereinigt die jeweiligen Eigenschaften der einzelnen Regler.

$$e_u^{PID}(t) = \mathcal{K}_P \cdot e_q(t) + \mathcal{K}_I \cdot \int_0^t e_q(t)dt + \mathcal{K}_D \cdot \frac{de_q(t)}{dt}. \tag{4.6}$$

In der Praxis reicht aber in den meisten Fällen ein PI-Regler. Daher haben auch wir einen PI-Regler in Betracht gezogen.

$$e_u^{PI}(t) = \mathcal{K}_P \cdot e_q(t) + \mathcal{K}_I \cdot \int_0^t e_q(t)dt. \tag{4.7}$$

Abbildung 4.3 zeigt das Ergebnis einer Regelung mit einen PI-Regler. Experimente zeigten jedoch[1], daß in unserem speziellen Fall der PI-Regler gegenüber dem reinen P-Regler keinen Vorteil bringt. Er bringt im Gegenteil sogar einen Nachteil. Dies liegt daran, daß wir uns in einem diskreten Raum befinden, und der P-Regler trotz der vermeintlich asymptotischen Annäherung den Sollwert genau erreicht, denn wir haben zum Messen nur Pixelgenauigkeit. Der Vorteil des PI-Reglers ist damit hinfällig, und der Nachteil des Überschwingens überwiegt. Daher wird letztendlich der reine P-Regler für die Annäherung angewandt. Da der PI-Regler aber eine allgemeinere Variante ist als der P-Regler, und dieser bei anderen Regelungsaufgaben Vorteile gegenüber dem P-Regler besitzten kann, folgt trotzdem im weiteren der Entwurf dieses Reglers. Der nachfolgende Abschnitt soll zeigen, wie der Regler bei uns zum Einsatz kommt. Dieser Abschnitt beinhaltet also das grundlegende Prinzip der von uns angewendeten geregelten Annäherung eines Objektes an das Loch.

4.3 Der Regelungsalgorithmus

Gegeben sei zum Zeitpunkt t ein Punkt $\boldsymbol{p}_s^{\boldsymbol{W}}(t) = (X_s^W(t), Y_s^W(t), Z_s^W(t))^T$ im Weltkoordinatensystem. In der späteren in Kapitel 6 beschriebenen Regelung wird dies ein markanter

[1] siehe Abschnitt 4.4.

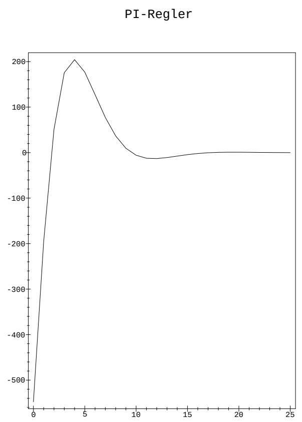


Abbildung 4.3: Ergebnis eines PI-Regelalgorithmus. Die vertikale Achse beschreibt den auszuregelnden Fehler, die horizontale Achse Zeitpunkt t. Der Sollwert wird exakt erreicht.

Punkt des Objektes in der Roboterhand sein. Diesen Punkt wollen wir zu einem im Weltkoordinatensystem nicht explizit beschriebenen Zielpunkt $\boldsymbol{p_z^W}$, der in der späteren Regelung ein markanter Punkt der Lochkontur sein wird, geregelt hinbewegen. Der Punkt $\boldsymbol{p_z^W}$ wird hier mit Absicht nicht in Abhängigkeit von t gesetzt, da der Punkt stationär ist, sich also während der gesamten Annäherung nicht verändern wird. Da zwei Kameras die Punkte beobachten, haben wir beide Weltpunkte im linken und rechten Bildkoordinatensystem beschrieben. Wir haben also zwei mal 2D Punkte im jeweiligen Bildkoordinatensystem, nämlich $\boldsymbol{p_s^{K^l}}(t)$ und $\boldsymbol{p_z^{K^l}}$, bzw. $\boldsymbol{p_s^{K^r}}(t)$ und $\boldsymbol{p_z^{K^r}}$ (siehe Abb. 4.4).

Unser Ziel ist es nun, den Punkt $\boldsymbol{p_s^W}(t)$ geeignet zu verändern, bis $\boldsymbol{p_z^W}$ erreicht ist. Dies kann aber immer nur mit den jeweiligen Informationen in den Kamerabildern geschehen.

Fassen wir den jeweiligen gemessenen Fehler (Soll-Ist-Vergleich) im rechten und linken Bild als einen Vektor zusammen, so gilt:

$$\boldsymbol{e_{2D}}(t) = \begin{pmatrix} x_z^{K^l} - x_s^{K^l}(t) \\ y_z^{K^l} - y_s^{K^l}(t) \\ x_z^{K^r} - x_s^{K^r}(t) \\ y_z^{K^r} - y_s^{K^r}(t) \end{pmatrix}. \tag{4.8}$$

Für die Ermittlung der Veränderung $\boldsymbol{u}(t)$ von $\boldsymbol{p_s^W}(t)$ zu $\boldsymbol{p_z^W}$ läßt sich nun mit Hilfe von Gleichung (3.13) und Gleichung (4.7) folgender Regelalgorithmus formulieren:

$$\boldsymbol{u}(t) = (\boldsymbol{J_{P^{l,r}}}(\boldsymbol{p^W})^T \cdot \boldsymbol{J_{P^{l,r}}}(\boldsymbol{p^W}))^{-1} \cdot \boldsymbol{J_{P^{l,r}}}(\boldsymbol{p^W})^T \cdot \left(\mathcal{K}_P\, \boldsymbol{e_{2D}}(t) + \mathcal{K}_I \int_0^t \boldsymbol{e_{2D}}(t)dt\right). \tag{4.9}$$

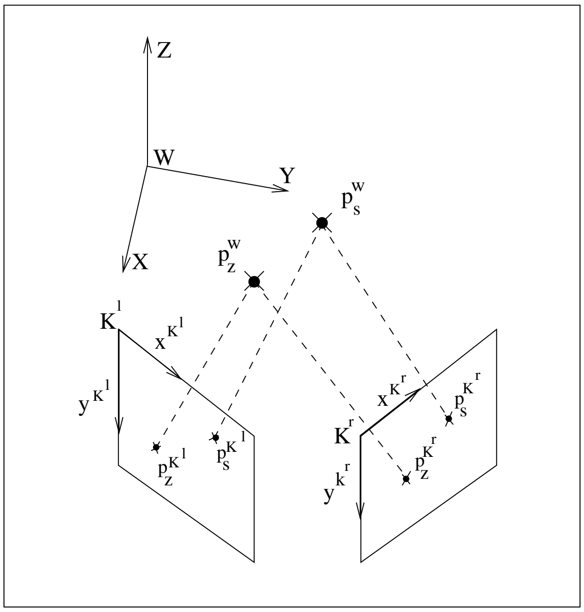


Abbildung 4.4: Projektion zweier Raumpunkte auf das linke und rechte Bildkoordinatensystem.

Diese Regelungsvorschrift ist dem Artikel von [Hager95] angelehnt. Für die Ermittlung der Jacobimatrix wurden dort jedoch explizite Kameraparameter benötigt, deren Angabe wir mit einer partiellen Ableitung mit direkter Hilfe der Projektionsmatrizen umgehen. Da die Ermittlung solcher Kameraparameter immer mit Fehlern behaftet ist, ist dies ein großer Vorteil gegenüber dem System von [Hager95].

Für die Annäherung des Punktes $\boldsymbol{p_s^W}(t)$ an den Zielpunkt $\boldsymbol{p_z^W}$ läßt sich nun mit Hilfe der obigen Gleichungen folgender Ablauf zum Zeitpunkt t beschreiben:

1. Bildaufnahme
2. Ermittlung der markanten Punkte $\boldsymbol{p_s^{K^l}}(t)$ und $\boldsymbol{p_z^{K^l}}$ im linken bzw. $\boldsymbol{p_s^{K^r}}(t)$ und $\boldsymbol{p_z^{K^r}}$ im rechten Bild
3. Ermittlung von $\boldsymbol{e_{2D}}(t)$ unter Verwendung von Gleichung (4.8)
4. Ermittlung von $\boldsymbol{u}(t)$ mit Hilfe der Gleichung (4.9)
5. Addition von $\boldsymbol{u}(t)$ auf $\boldsymbol{p_s^W}(t)$: $\boldsymbol{p_{s_{neu}}^W}(t) \;=\; \boldsymbol{p_{s_{alt}}^W}(t) + \boldsymbol{u}(t)$
6. Bewegen des Roboters zum ermittelten Punkt $\boldsymbol{p_{s_{neu}}^W}(t)$
7. $\boldsymbol{e_{2D}^{sum}}(t) = \boldsymbol{e_{2D}^{sum}}(t) + \boldsymbol{e_{2D}}(t)$
8. t=t+1

Dieser Ablauf geschieht solange, bis der Fehler $\boldsymbol{e_{2D}}(t)$ gleich Null, bzw. bis eine vom Benutzer angegebene Grenze unterschritten ist (z.B. 2 Pixel in jeder Vektorkomponente).

Bemerkungen:

1. Falls sich die Position des Ziels während einer Annäherung nicht ändert, müssen folgerichtig die Punkte $\boldsymbol{p_z^{K^l}}$ und $\boldsymbol{p_z^{K^r}}$ nur einmal am Anfang ermittelt werden und nicht bei jedem Schritt immer wieder. Dieser gesparte Objekterkennungsschritt bringt einen großen Zeitgewinn, gerade wenn die Erkennung des Zielpunktes nicht nur aus einem einfachen Extrahieren eines hellen Punktes aus dem Bild, sondern aus einem komplizierten Algorithmus besteht (vgl. Kap. 6.3).

2. Mit Zeitpunkt t ist der Zeitpunkt nach einem Annäherungsschritt gemeint, wir befinden uns also in einem diskreten Zeitraum, so daß die Ermittlung des Integrales für den I-Anteil des PI-Reglers durch eine Aufsummierung der Fehler vom Ausgangsfehler bis zu diesem Annäherungsschritt approximiert wird:

$$\int_0^t \boldsymbol{e_{2D}}(t)dt \approx \sum_0^t \boldsymbol{e_{2D}}(t). \tag{4.10}$$

Dies begründet den siebten Schritt im Ablauf.

4.4 Experimente

Um den entwickelten Regler zu testen und seine Robustheit gegenüber Kalibrierungsfehlern zu zeigen, wurden von uns nachstehende Experimente durchgeführt. Hierbei standen folgende Fragen im Vordergrund, über die die Experimente Aufschluß geben sollten.

- Wie viele Kalibrierungspunkte sind für die Erstellung der Projektionsmatrizen nötig, um noch eine gutes Regelergebnis zu erzielen ?
- Wie sind die Konstanten $\mathcal{K}_p$ und $\mathcal{K}_I$ zu wählen, um eine bestmögliche Regelung zu erzielen ?
- Wie reagiert der Regelalgorithmus auf Kalibrierungsfehler ?

Für die Beantwortung der Fragen wurde folgender Aufbau angewendet. Die Kamera beobachtete im abgedunkelten Raum von ca. 1400 mm Entfernung mit einer eingestellten Brennweite von ca. 17 mm den Manipulator, der als Arbeitsbereich einen virtuellen Quader von 200x400x200 mm^3 besaß und eine Lichtquelle in der Hand hielt. Dies gestattete eine möglichst schnelle und einfache Erkennung im Bild, um eventuell auftretende Fehler der Bilderkennung zu vermeiden. Innerhalb dieses Arbeitsbereiches wurde eine wie in Kapitel 3 beschriebene Kalibrierung mit drei in der Anzahl unterschiedlichen Mengen von 3D Referenzpunkten vorgenommen, um die jeweiligen Projektionsmatrizen für die rechte und linke Kamera zu erhalten. In der ersten Kalibrierung wurden 225 3D Referenzpunkte angefahren, in der zweiten 45 und in der dritten nur 8 Punkte, die genau den Eckpunkten des Arbeitsbereiches entsprachen. Um den für die Regelung nötigen Sollwertvektor zu erhalten, wurde ein definierter 3D Punkt $\boldsymbol{p^d}$ im Arbeitsbereich angefahren und mittels Schwerpunktbestimmung des auf die Kameraebenen projizierten Lichtpunktes die zugehörigen 2D Punkte ermittelt. Dann wurde der Manipulator an eine beliebige Stelle im Arbeitsbereich gefahren und mittels der oben beschriebenen Regelung zum Sollwert hingeregelt. Die Regelung für das erste Experiment beruht auf einen reinen P-Regler, die Konstante $\mathcal{K}_I$ ist also Null. Als Ergebnis wurden sowohl die Anzahl der Durchgänge bis zum Erreichen des Sollwertes, als auch die Abweichung des erreichten 3D Punktes zum definierten Punkt $\boldsymbol{p^d}$ in X-,Y- und Z-Richtung gemessen. Tabelle 4.1 zeigt die Ergebnisse.

Anzahl der Kalibrierungspunkte	$\mathcal{K}_P$	$\mathcal{K}_I$	Anzahl der Durchgänge	Abweichung in mm (X, Y, Z)
225	0.5	0.0	11	(0.7, 0.09, 0.4)
45	0.5	0.0	11	(0.1, 0.2, 0.03)
8	0.5	0.0	11	(1.3, 0.3, 0.5)

Tabelle 4.1: Testergebnisse mit verschiedenen Kalibrierungen.

Die Ergebnisse zeigen deutlich, daß für die Regelung eine vorherige Kalibrierung mit nur 8 Punkten, nämlich die Ecken des Arbeitsbereiches, völlig ausreicht. Denn die Anzahl der Durchgänge bis zum Erreichen des Sollwertes ist bei jeder Anwendung der durch die verschiedenen Kalibrierungspunkte ermittelten Projektionsmatrizen gleich, und die Größe des gemessenen Abstandsvektors zum tatsächlichen Punkt $\boldsymbol{p^d}$ ist selbst bei 8 Punkten mit 1,4 mm noch vernachlässigbar gering. Die Kalibrierung mit nur 8 Punkten bringt einen enormen Zeitvorteil, da jetzt nur noch 8 mal der Bildpunkt in einem Bild gesucht werden muß[2].

Das nächste Experiment behandelte die Frage, wie sich eine Veränderung der Proportional- und Integralkonstante $\mathcal{K}_p$ und $\mathcal{K}_I$ sowohl auf die Anzahl der Durchgänge bis zum Erreichen des Sollwertes, als auch auf die Abweichung des erreichten 3D Punktes zum definierten Punkt $\boldsymbol{p^d}$ in X-, Y- und Z-Richtung auswirkt. Tabelle 4.2 zeigt die Hauptergebnisse zusammengefaßt.

Die Ergebnisse zeigen, daß je größer die $\mathcal{K}_P$-Konstante wird, desto weniger Durchgänge benötigt werden. Bei einer bestimmten Größe kommt es aber zu einer Überschwingung, d.h. der Manipulator geht erst am Ziel vorbei, und nähert sich dann von der anderen Seite an. Das gleiche passiert, wenn man die $\mathcal{K}_I$-Konstante größer als Null setzt. Auch die Durchgänge bis zum Erreichen des Zieles nehmen wegen des Hin- und Herschwingens

[2] Die Suche im linken und rechten Bild läuft aufgrund der vorhandenen Rechnerarchitektur und Implementation parallel.

Anzahl der Kalibrierungspunkte	$\mathcal{K}_P$	$\mathcal{K}_I$	Anzahl der Durchgänge	Abweichung in mm (X, Y, Z)	Überschwingung des Reglers
45	0.5	0.0	11	(0.1, 0.2, 0.03)	nein
45	0.7	0.0	8	(0.5, 0.7, 0.4)	nein
45	1.0	0.0	3	(2.0, 0.1, 1.4)	ja
45	0.5	0.07	26	(0.25, 0.17, 0.5)	ja

Tabelle 4.2: Testergebnisse mit verschiedenen Reglerkonstanten $\mathcal{K}_P$ und $\mathcal{K}_I$.

hier stark zu. In unserem Falle bedeutet das zum einen, daß wir die $\mathcal{K}_P$-Konstante nicht zu hoch setzen dürfen, da sonst bei unserer späteren Anwendung der Objektannäherung zum Lochbrett der Manipulator in das Lochbrett hineingeregelt wird und alles zerstören würde. Zum anderen dürfen wir aus eben diesem Grund der Überschwingung auch keinen PI-Regler verwenden, der die Annäherung außerdem noch verzögert, sondern nur einen reinen P-Regler.

Für das letzte Experiment wurde die Kalibrierungmatrix für die 45 Punkte genommen, und zusätzlich die Kamerakopfstellung um eine Fehlergröße verändert, bevor der Sollwert in den projizierten Bildern ermittelt wurde. Mit dieser zur Kamerakalibrierung abweichenden Kopfstellung wurde wieder die Regelung zum Sollwert, der Projektion von $\boldsymbol{p^d}$, vorgenommen, und die Anzahl der Durchgänge bis zum Erreichen des Sollwertes sowie die Abweichung des erreichten 3D Punktes zum definierten Punkt $\boldsymbol{p^d}$ in X-,Y- und Z-Richtung gemessen. Tabelle 4.3 zeigt die dabei erreichten Ergebnisse mit einem eingestellten Proportionalitätsfaktor von $\mathcal{K}_P = 0.5$ und und einer Integralkonstanten von $\mathcal{K}_I = 0$.

Veränderte Kameraparameter			Anzahl der	Abweichung in mm
Schwenkwinkel	Neigewinkel	Brennweite	Durchgänge	(X, Y, Z)
5°	0°	0.0 mm	9	(2.5, 1.8, 2.8)
10°	0°	0.0 mm	11	(0.7, 0.2, 1.0)
10°	5°	0.0 mm	20	(0.8, 0.3, 1.3)
0°	0°	-5.75 mm	20	(2.0, 0.1, 2.8)
10°	5°	-5.75 mm	20	(2.0, 0.08, 3.0)
25°	5°	-5.75 mm	22	(1.3, 0.5, 3.2)

Tabelle 4.3: Testergebnisse mit veränderten Kameraparametern nach der Kalibrierung.

Die Tabelle zeigt deutlich, daß selbst bei einer stark gestörten Kalibrierung noch gute Ergebnisse erzielt werden können. Die Anzahl der Annäherungsschritte nimmt zwar zu, aber die Abweichung vom Originalwert ist weiterhin sehr gering. Dies ist ein enormer Vorteil gegenüber der Ein-Schritt-Methode. Sie würde eine direkte Umrechnung der gefundenen 2D Punkte mit Hilfe der Projektionsmatrizen benutzen und völlig versagen. Die direkte Umrechnung ohne die Regelung bei gegebenen veränderten Kameraparametern wie in der vorletzten Zeile der Tabelle 4.3 würde eine Abweichung in X-Richtung um 3 mm, in Y-Richtung um 159 mm und in Z-Richtung um 91 mm liefern. Der durch die Ein-Schritt-Methode verursachte Fehler (euklidischer Abstand) im Weltkoordinatensystem von 183.2 mm ist mit Hilfe der Regelung also auf nur 3.6 mm reduziert worden. Dies ist ein wichtiger Pluspunkt für die Regelung. Die letzte Zeile der Tabelle zeigt eine Kamerakonfiguration, bei der die Kamera so verdreht wurde, daß der Startpunkt gerade noch im Bild zu sehen war. Da auch hier die Regelung erfolgreich war, läßt sich über den Algorithmus sagen, daß er erst versagt, wenn der Istwert (bzw. der Sollwert) nicht mehr im linken oder rechten Bild zu sehen ist.

Im folgenden Kapitel wird der grundlegende Regelungsalgorithmus für die erste große Anwendung genutzt. Er dient zur Unterstützung der Objektklassifizierung. Mit Hilfe des Algorithmus wird nämlich der Sehstrahl der Kamera ermittelt, an dem das Objekt zur Kamera zwecks Erkennung der Grundfläche hinbewegt werden soll. Der Regelung liefert hier also einen Beitrag zur aktiven Wahrnehmung ("active vision").

Kapitel 5

Visuell basierte Regelung für die Objektklassifizierung

In diesem Kapitel findet die in Kapitel 4 vorgestellte Regelung eine erste Anwendung. Sie dient dazu, den Sehstrahl der linken Kamera zu ermitteln. Mit dessen Hilfe läßt sich die Grundfläche des Objektes, das ja anhand seiner Grundfläche erkannt werden soll, parallel zur Bildebene ausrichten und entlang des Sehstrahls zur Kamera hinbewegen, bis sie im Bild eine bestimmte Größe erreicht hat. Diese Ausrichtung geschieht, um die Finger aus dem Bild zu extrahieren und zu löschen, damit nur noch die Grundfläche im Bild zu sehen ist. Die Regelung wird hier also eingesetzt, um das an sich schwierige Problem der Extraktion der Finger aus dem Bild durch eine rotations- und skalierungsinvariante Erkennung der Finger zu reduzieren, indem man mit Hilfe der Regelung eine "normierte" Situation schafft, so daß eine einfache Erkennung der Finger möglich ist. Nach Vorstellung dieser Anwendung beschreibt das Kapitel weiterhin die Extraktion der Kontur der Grundfläche und die Grundflächenrepräsentation, bei der die Kontur mit Hilfe einer Autoregression durch einen Vektor approximiert wird. Zuletzt beschreiben wir die Grundflächenverifikation, die mit Hilfe eines Abstandsmaßes im durch Autoregression entstandenen Musterraum geschieht.

5.1 Objektausrichtung zur Kamera

Damit man die Grundfläche des Objektes und die Finger ohne perspektivische Verzerrung mit einer Kamera (in unserem Falle mit der linken) sehen kann, muß sie parallel zur Kameraebene ausgerichtet sein und den Sehstrahl der Kamera schneiden. Da das Objekt von Anfang an so in der Roboterhand liegt, daß die Oberseite der Finger und die Grundfläche des Werkstückes parallel sind, läßt sich die gewünschte Ausrichtung der Grundfläche erreichen, indem man die Roboterhand am Sehstrahl parallel zu Kameraebene ausrichtet (siehe Abb. 5.1).

Der nächste Abschnitt zeigt, wie sich dies erreichen läßt.

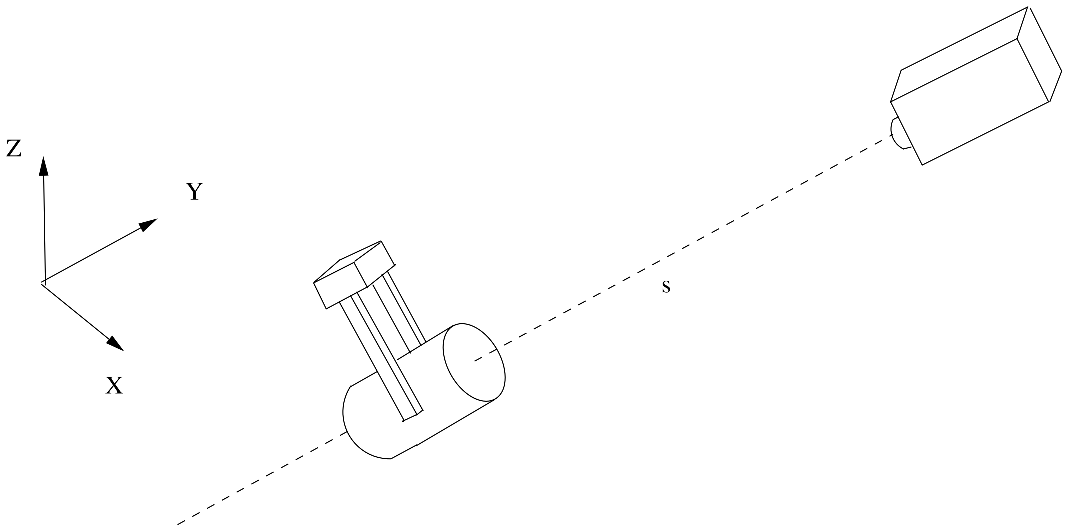


Abbildung 5.1: Ausrichtung der Grundfläche des Werkstückes parallel zur Kameraebene am Sehstrahl.

5.1.1 Ermittlung des Sehstrahls

Um den Sehstrahl der Kamera zu ermitteln, läßt sich folgende Regelungsprozedur anwenden, die auf den Ergebnissen aus Kapitel 4 gestützt ist. Am Anfang ist die Roboterhand parallel zur YZ-Ebene des Weltkoordinatensystems ausgerichtet und hat eine Leuchtdiode in der Hand, damit sich eine möglichst schnelle Extraktion der Position des relevanten Punktes im Bild anwenden läßt (vgl. Kap. 3). Die Kamera beobachtet die gesamte Szene. Die Hand wird nun an einer zur YZ-Ebene des Weltkoordinatensystems parallelen Ebene E_1 zum Mittelpunkt des Bildes geregelt hingeführt, was gleichbedeutend ist mit einer geregelten Hinführung zu dem Schnittpunkt von $E1$ mit dem Sehstrahl s der Kamera. Dieser Schnittpunkt sei mit $\boldsymbol{SP_1}$ bezeichnet. Im nächsten Schritt wird die Hand wieder entlang einer zur YZ-Ebene des Weltkoordinatensystems parallelen Ebene E_2 zum Schnittpunkt der Ebene mit dem Sehstrahl geführt. Die Ebene E_2 ist allerdings um ein Inkrement positiv in X-Richtung verschoben. Der neue Schnittpunkt sei mit $\boldsymbol{SP_2}$ bezeichnet (siehe Abb. 5.2). Die Punkte $\boldsymbol{SP_1}$ und $\boldsymbol{SP_2}$ bilden eine Gerade, die kollinear zum Sehstrahl der Kamera ist. Mit Hilfe dieser beiden Punkte lassen sich die benötigten Winkel errechnen, um die Hand so zu rotieren, daß sie parallel zur Kameraebene ist. Bevor der Algorithmus hierfür erklärt wird, wollen wir aber noch vorher näher auf den Regelalgorithmus eingehen.

Die in dieser Prozedur angewendete Regelung ist ein Spezialfall der für die Annäherung des Objektes zum Lochbrett angewendeten Regelung. Die Vorschrift für diese allgemeine Regelung wurde in Kapitel 4 gezeigt. Da die hier angewendete Regelung jeweils an einer zur YZ-Ebene des Weltkoordinatensystems parallelen Ebene erfolgt, bleibt der X-Wert der zu errechnenden 3D Koordinate immer gleich, so daß man nur eine 2D Bild - 2D Welt-Transformation benötigt. Sie sieht im einzelnen wie folgt aus:

Mit Hilfe der aus der Kalibrierung gewonnenen Projektionsmatrizen lassen sich durch An-

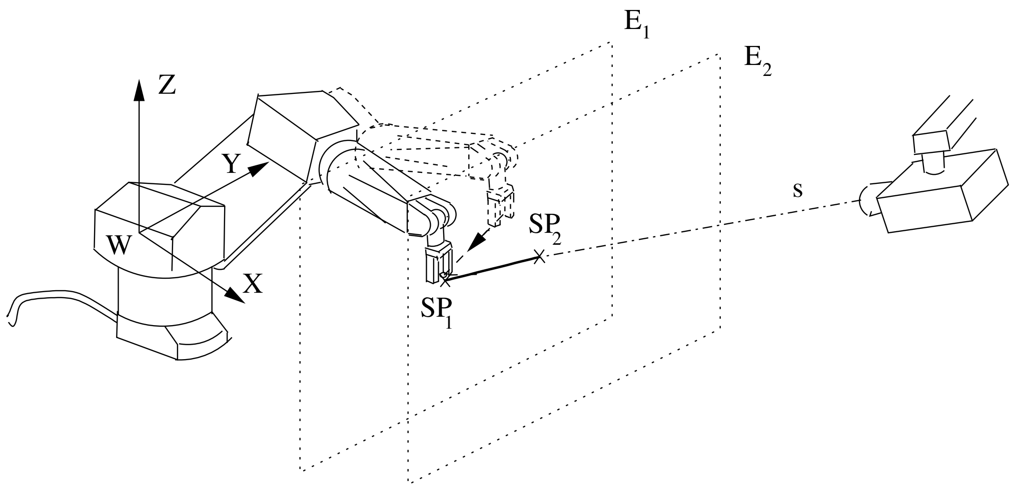


Abbildung 5.2: Bildung der Geraden $\boldsymbol{SP_1SP_2}$ aus Schnittpunkten der Ebenen E_1 und E_2 mit dem Sehstrahl s der Kamera.

wendung von Gleichung (3.1) die 3D Punkte auf 2D Bildpunkte transformieren. Aufgrund der Parallelität zur YZ-Ebene des Weltkoordinatensystems läßt sich Gleichung (3.1) in diesem Fall durch Weglassen der ersten Spalte in der Projektionsmatrix und Entfernung der X-Komponente des Weltkoordinatenvektors in homogenen Koordinaten vereinfachen:

$$\begin{pmatrix} \lambda_1 x^{K^l} \\ \lambda_1 y^{K^l} \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} P_{12}{}^l & \cdots & P_{14}{}^l \\ P_{22}{}^l & \cdots & P_{24}{}^l \\ P_{32}{}^l & \cdots & P_{34}{}^l \end{pmatrix} \cdot \begin{pmatrix} Y^W \\ Z^W \\ 1 \end{pmatrix}. \tag{5.1}$$

Für die Transformation des Bild-Punktes auf die Ebene E_i ergibt sich somit folgende Gleichung:

$$\begin{pmatrix} -P_{14}{}^l \\ -P_{24}{}^l \\ -P_{34}{}^l \end{pmatrix} = \begin{pmatrix} P_{12}{}^l & P_{13}{}^l & -x^{K^l} \\ P_{22}{}^l & P_{24}{}^l & -y^{K^l} \\ P_{32}{}^l & P_{34}{}^l & -1 \end{pmatrix} \cdot \begin{pmatrix} Y^W \\ Z^W \\ \lambda_1 \end{pmatrix}. \tag{5.2}$$

Um nun einen Zusammenhang zwischen Veränderungen im Weltkoordinatensystem und im Bildkoordinatensystem herzustellen, benötigt man die zugehörige Jakobimatrix, die sich gegenüber Gleichung (3.7) entsprechend vereinfacht:

$$\boldsymbol{J_{P^l}} \begin{pmatrix} Y^W \\ Z^W \end{pmatrix} = \begin{pmatrix} \frac{\partial x^{K^l}}{\partial Y^W} & \frac{\partial x^{K^l}}{\partial Z^W} \\ \frac{\partial y^{K^l}}{\partial Y^W} & \frac{\partial y^{K^l}}{\partial Z^W} \end{pmatrix}. \tag{5.3}$$

Der Fehler, den es geregelt zu minimieren gilt, ist der Abstand des im Bild gefunden Lichtpunktes zum Mittelpunkt des Bildes $(x^M, y^M)^T$:

$$\boldsymbol{e_{2D}}(t) = \begin{pmatrix} x^M - x^{K^l}(t) \\ y^M - y^{K^l}(t) \end{pmatrix}. \tag{5.4}$$

Für die Ermittlung der Veränderung der 2D Weltkoordinaten läßt sich folgender P-Regelalgorithmus aufstellen[1].

$$\boldsymbol{u}(t) = \begin{pmatrix} dY^W(t) \\ dZ^W(t) \end{pmatrix} = \left(\boldsymbol{J_{P^l}}\begin{pmatrix} Y^W \\ Z^W \end{pmatrix}\right)^{-1} \cdot \mathcal{K}_P \cdot \boldsymbol{e_{2D}}(t). \tag{5.5}$$

Zu bemerken ist hier noch, daß keine Pseudoinverse zu der Jacobimatrix gebildet werden muß (vgl. Kap. 3.2). Es kann direkt die inverse Matrix gebildet werden, da es sich in diesem Falle um eine 2x2 Matrix handelt. Die zu jedem Schritt t ermittelte Änderung $\boldsymbol{u}(t)$ wird zu der jeweiligen aktuellen Roboterkoordinate hinzuaddiert bis $\boldsymbol{e_{2D}}(t)$ gleich Null ist. Wir können hier trotz Anwendung eines P-Reglers eine Minimierung des Fehlers bis auf Null voraussetzen, da wir uns im diskreten Raum befinden und die unterste Fehlergrenze von einem Pixel als Abstand trotz asymptotischer Annäherung des Fehlers unterschritten wird.

Nachdem man also den Regelalgorithmus für beide Ebenen E_1 und E_2 angewendet hat, erhält man zwei Punkte im Weltkoordinatensystem, die eine Gerade bilden, die dem Sehstrahl der Kamera entspricht. Um die Roboterhand an dem Sehstrahl auszurichten, benötigt man nun zum einen die drei Winkel, um die sich die Hand um die X-,Y- und Z-Achse des Weltkoordinatensystems drehen muß. Zum anderen benötigt man die Angabe dieser Winkel in ZYZ-Euler-Form, da die Rotation des Toolkoordinatensystems und somit auch der Roboterhand in den Eulerwinkeln $(yaw, pitch, roll)$ angegeben wird[2]. Die Ausrichtung wird im nächsten Abschnitt beschrieben.

5.1.2 Ausrichtung zum Sehstrahl

Mit Hilfe der beiden ermittelten Punkte $\boldsymbol{SP_1} = (X^{SP_1}, Y^{SP_1}, Z^{SP_1})^T$ bzw. $\boldsymbol{SP_2} = (X^{SP_2}, Y^{SP_2}, Z^{SP_2})^T$ lassen sich zunächst folgende Winkel bezüglich des festen Weltkoordinatensystems angeben:

Projiziert man die aus $\boldsymbol{SP_1}$ und $\boldsymbol{SP_2}$ gebildete Gerade auf die YZ-Ebene, ergibt sich folgender Winkel zwischen der Y-Achse und der projizierten Gerade:

$$\tan(\gamma) = \frac{Z^{SP_2} - Z^{SP_1}}{Y^{SP_2} - Y^{SP_1}}. \tag{5.6}$$

Dies entspricht einer später gewünschten Rotation der Hand um die X-Achse des Weltkoordinatensystems.

---

[1] Wir haben bewußt nur einen P-Regelalgorithmus gewählt, da er schneller zum Ziel kommt als der PI-Regelalgorithmus, der sich erst überschwingt und langsam ausschwingt.

[2] siehe Kapitel 2.

Wird die aus $\boldsymbol{SP_1}$ und $\boldsymbol{SP_2}$ gebildete Gerade auf die XZ-Ebene projiziert, ergibt sich zwischen der X-Achse und der projizierten Gerade der Winkel

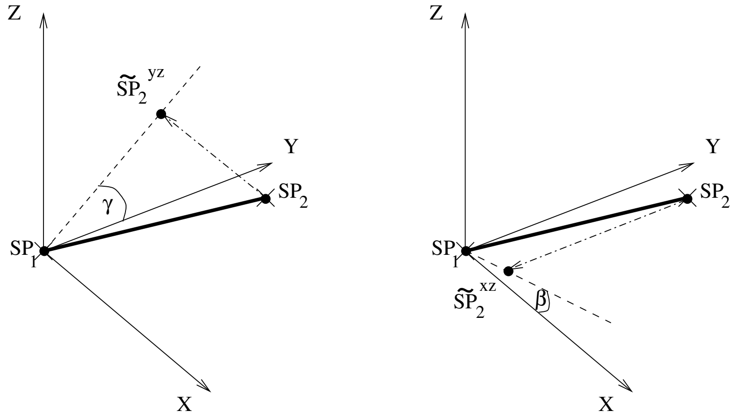
$$\tan(\beta') \;=\; \frac{Z^{SP_2} - Z^{SP_1}}{X^{SP_2} - X^{SP_1}}. \tag{5.7}$$

Dies entspricht einer Rotation um die Y-Achse. Zu beachten ist hierbei, daß die ermittelte Rotation um β' gegen die eigentliche Drehrichtung einer Drehung um die Y-Achse läuft. Außerdem soll die Hand nach unten gerichtet sein, d.h man muß vorher die Hand um 180 Grad in Drehrichtung der Y-Achse drehen. Somit ist der richtige Winkel, um den man drehen muß: $\beta = 180 - \beta'$.

Den Winkel für die Drehung um die Z-Achse erhält man, indem man den Winkel betrachtet, der zwischen der X-Achse und der Geraden gebildet wird, die durch Projektion der Geraden $\overline{\boldsymbol{SP_1SP_2}}$ auf die XY-Ebene entstanden ist:

$$\tan(\alpha) \;=\; \frac{Y^{SP_2} - Y^{SP_1}}{X^{SP_2} - X^{SP_1}}. \tag{5.8}$$

Abbildung 5.3 soll noch einmal die verschiedenen Projektionen auf die Ebenen und die entstehenden Winkel veranschaulichen.



(a) Entstehender Winkel für Rotation um die X-Achse.

(b) Entstehender Winkel für Rotation um die Y-Achse.

(c) Entstehender Winkel für Rotation um die Z-Achse.

Abbildung 5.3: Resultierende Drehwinkel für die Roboterhand um die einzelnen Achsen aus Projektion der Geraden $\overline{\boldsymbol{SP_1SP_2}}$ auf die jeweiligen Ebenen.

Für die eigentliche Ausrichtung der Hand zum Sehstrahl der Kamera benötigen wir die Winkel noch in Eulerwinkeln $(yaw, pitch, roll)$. Die Umrechnung geschieht mit nachfolgend beschriebenen Gleichungen[Craig89].

Man ermittelt die Rotationsmatrix $\boldsymbol{R^f}$ von α, β und γ, die die Drehung des Toolkoordinatensystems T in Bezug auf ein *festes* Referenzkoordinatensystem - nämlich das Roboterkoordinatensystem W - bewirkt. Das heißt, man ermittelt die Rotationsmatrix $\boldsymbol{R^f}$, die folgendes aussagt:

Rotiere T um die X-Achse von W um den Winkel γ, dann rotiere T um die Y-Achse von W um den Winkel β und rotiere zuletzt T um die Z-Achse von W um den Winkel α.

$$
{}^{\boldsymbol{W}}_{\boldsymbol{T}}\boldsymbol{R}^{\boldsymbol{f}}_{\boldsymbol{XYZ}}(\gamma,\beta,\alpha) = \boldsymbol{R}^{\boldsymbol{f}}_{\boldsymbol{z}}(\alpha)\cdot \boldsymbol{R}^{\boldsymbol{f}}_{\boldsymbol{y}}(\beta)\cdot \boldsymbol{R}^{\boldsymbol{f}}_{\boldsymbol{x}}(\gamma)
$$

$$
= \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix}
$$

$$
= \begin{pmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta)\cos(\gamma)+\sin(\alpha)\sin(\gamma) \\ \sin(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta)\cos(\gamma)-\cos(\alpha)\sin(\gamma) \\ -\sin(\beta) & \cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{pmatrix}. \tag{5.9}
$$

Eine Rotationsmatrix $\boldsymbol{R}^e$, die Drehungen mit ZYZ-Eulerwinkeln bewirkt, d.h die zuerst die Z-Achse um den Winkel α^e (yaw) in Bezug zum Weltkoordinatensystem dreht, dann die nach der Drehung *neu entstandene* Y-Achse um den Winkel β^e ($pitch$) und um die wiederum neu entstandene Z-Achse um den Winkel γ^e ($roll$) dreht, hat folgendes Aussehen[3]:

$$
{}^{\boldsymbol{W}}_{\boldsymbol{T}}\boldsymbol{R}^{e}_{\boldsymbol{Z'Y'Z'}}(\alpha^e,\beta^e,\gamma^e) = \boldsymbol{R}^{e}_{\boldsymbol{z}}(\alpha^e)\cdot \boldsymbol{R}^{e}_{\boldsymbol{y}}(\beta^e)\cdot \boldsymbol{R}^{e}_{\boldsymbol{z}}(\gamma^e) =
$$

$$
\begin{pmatrix} \cos(\alpha^e)\cos(\beta^e)\cos(\gamma^e)-\sin(\alpha^e)\sin(\gamma^e) & -\cos(\alpha^e)\cos(\beta^e)\sin(\gamma^e)-\sin(\alpha^e)\cos(\gamma^e) & \cos(\alpha^e)\sin(\beta^e) \\ \sin(\alpha^e)\cos(\beta^e)\cos(\gamma^e)+\cos(\alpha^e)\sin(\gamma^e) & -\sin(\alpha^e)\cos(\beta^e)\sin(\gamma^e)+\cos(\alpha^e)\cos(\gamma^e) & \sin(\alpha^e)\sin(\beta^e) \\ -\sin(\beta^e)\cos(\gamma^e) & \sin(\beta^e)\sin(\gamma^e) & \cos(\beta^e) \end{pmatrix}. \tag{5.10}
$$

Aus der Rotationsmatrix $\boldsymbol{R}^{\boldsymbol{f}}$ lassen sich implizit über $\boldsymbol{R}^e$ die benötigten Eulerwinkel extrahieren unter der Annahme, daß $\boldsymbol{R}^{\boldsymbol{f}}$ gebildet wurde wie $\boldsymbol{R}^e$. Man definiert also $\boldsymbol{R}^e := \boldsymbol{R}^{\boldsymbol{f}}$. Nun lassen sich die benötigten Winkel aus $\boldsymbol{R}^e$ (bzw. $\boldsymbol{R}^{\boldsymbol{f}}$) extrahieren.

Falls β^e ungleich Null ist, gilt:

$$
\begin{aligned}
\beta^e &= atan2(\sqrt{(R^e_{31})^2+(R^e_{32})^2}, R^e_{33}), \\
\alpha^e &= atan2(R^e_{23}/sin(\beta^e), R^e_{13}/sin(\beta^e)), \\
\gamma^e &= atan2(R^e_{32}/sin(\beta^e), -R^e_{31}/sin(\beta^e)).
\end{aligned} \tag{5.11}
$$

Die positive Wurzel hält β^e in den Grenzen: $0° \le \beta^e \le 180°$.

Spezialfall: Da bei $\beta^e = 0°$ und $\beta^e = 180°$ die Gleichung für α^e und γ^e undefiniert ist, läßt sich die Konvention einführen, α^e gleich Null zu setzen. Somit ergibt sich folgende spezielle Lösung für $\beta^e = 0°$:

$$
\begin{aligned}
\beta^e &= 0°, \\
\alpha^e &= 0°, \\
\gamma^e &= atan2(-R^e_{12}, R^e_{11}).
\end{aligned} \tag{5.12}
$$

[3]vgl. Kap. 2.

Für $\beta^e = 180°$ ergibt sich:

$$\begin{aligned} \beta^e &= 180°, \\ \alpha^e &= 0°, \\ \gamma^e &= atan2(R^e_{12}, -R^e_{11}). \end{aligned} \tag{5.13}$$

Die Funktion $atan2(v_1, v_2)$ entspricht hierbei der Funktion $atan(v_1/v_2)$, jedoch mit Berücksichtigung des jeweiligen Quadranten, in dem man sich befindet:

$$atan2(v_1, v_2) = \begin{cases} atan(v_1/v_2), & \text{falls } (v_1 > 0) \wedge (v_2 > 0), \\ atan(v_1/v_2) + 90°, & \text{falls } (v_1 < 0) \wedge (v_2 > 0), \\ atan(v_1/v_2) + 180°, & \text{falls } (v_1 < 0) \wedge (v_2 < 0), \\ atan(v_1/v_2) + 270°, & \text{falls } (v_1 > 0) \wedge (v_2 < 0), \\ 0°, & \text{falls } (v_1 >= 0) \wedge (v_2 = 0), \\ 180°, & \text{falls } (v_1 < 0) \wedge (v_2 = 0), \\ 90°, & \text{falls } (v_1 = 0) \wedge (v_2 > 0), \\ 270°, & \text{falls } (v_1 = 0) \wedge (v_2 < 0). \end{cases} \tag{5.14}$$

Nun läßt sich also die Roboterhandfläche parallel zur Bildebene der linken Kamera ausrichten. Der nächste Abschnitt beinhaltet eine Komponente, die die visuelle Aufmerksamkeit für die Objektsuche auf einen Teil des Bildes beschränken soll, es ist also ein Modul zur Steuerung der aktiven visuellen Wahrnehmung.

5.1.3 Vergenzregelung der Kamera

Um die Suche des Objektes für den Beginn der in Kapitel 6 beschriebenen geregelten Annäherung zu vereinfachen, lassen wir an dieser Stelle der Kalibrierung die rechte Kamera so vergieren, daß auch sie den auf ihre Bildebene projizierten Lichtpunkt in der Mitte des Bildes hat. Diese Fixierung eines 3D-Punktes hat den Vorteil, daß man schon einen bestimmten Bereich für die später noch beschriebene Suche des Objektes vor dem ersten Annäherungsschritt hat. Dieser Suchbereich befindet sich also im linken und im rechten Bild um den Mittelpunkt des jeweiligen Bildes. Der Startpunkt der Annäherung ist der 3D Punkt, auf den die Kameras fixiert sind. Die Vergenzregelung erfolgt mit Hilfe eines einfaches P-Reglers. Der Vergenzwinkel $\theta_r(t)$, um die sich die rechte Kamera drehen soll, wird jeweils aus dem Soll-Ist-Vergleich zwischen de x-Koordinate x^M der Bildmitte und der x-Koordinate des aktuell ermittelten Lichtpunktes $x^{K^r}(t)$ multipliziert mit einem Proportionalitätsfaktor $\mathcal{K}_{P_{\theta_r}}$ ermittelt:

$$\theta_r(t) = K_{P_{\theta_r}} \cdot (x^M - x^{K^r}(t)). \tag{5.15}$$

Nachdem sich also die Hand zu der Bildebene der linken Kamera parallel ausgerichtet hat, und man sich die Position merkt, läßt sich, nachdem das Objekt gegriffen wurde, die Grundfläche des Objektes parallel zur Bildebene ausrichten. Vorausgesetzt wird hier, daß das Objekt während des Greifvorganges stand und von der Seite gegriffen wurde. Das Greifen ist in dieser Arbeit zwar nicht implementiert, da es sonst den Umfang der Arbeit sprengen würde, im Ausblick in Kapitel 7 wird aber trotzdem eine Verwirklichung des Greifvorganges beschrieben. Nach dem Greifen liegt das Objekt so in der Hand, daß sich die Grundfläche parallel zur Handfläche befindet. Zur Erinnerung der derzeitigen Konstellation sei noch einmal auf Abbildung 5.1 verwiesen.

Ein Problem für die Grundflächenextraktion sind die Roboterfinger, die sich auch im Bild befinden. Es muß nun nach einer Möglichkeit gesucht werden, diese im Bild zu segmentieren und zu löschen. Neben der Implementation eines aufwendigen rotations- und skalierungsinvarianten Erkennungsalgorithmus bietet sich hier die Möglichkeit an, eine geeignete Transformation durch eine entsprechende Roboterbewegung zu finden, so daß die Finger am Ende der Transformation für jeden Erkennungsdurchgang das gleiche Aussehen haben, und somit leicht lokalisiert und gelöscht werden können. Die Transformation besteht aus einer Drehung der Hand, bis die Finger senkrecht im Bild sind und einer Bewegung entlang des Sehstrahls der Kamera, bis die Finger mit einer bestimmten Breite im Bild zu sehen sind. Auch diese Module für die Normierung der Sicht lassen sich als Module für das Konzept der aktiven visuellen Wahrnehmung ("active vision") bezeichnen. Hier wird jedoch nicht die Kamera zur visuellen Aufmerksamkeit gesteuert, sondern der Roboterarm steuert das Objekt so, daß das Objekt in einer normierten Sicht in einen bestimmten fixierten Bereich des Gesichtsfeldes der Kamera kommt. Die Transformation für diese normierte Sicht und das Löschen der Finger aus dem Bild mit anschließender Konturextraktion der Grundfläche des Objektes soll der nächste Abschnitt beschreiben.

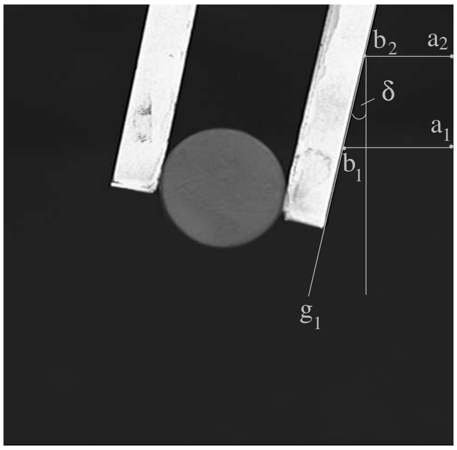
5.1.4 Fingersegmentierung und Konturextraktion

Um die Finger und die Objektgrundfläche möglichst groß im Bild zu haben, wird zuerst der Zoom der Kamera maximal eingestellt. Dann wird die Hand so gedreht, daß die Finger senkrecht im Bild erscheinen. Hierzu wird im Bild der Winkel gemessen, um den sich die Hand drehen soll. Dies geschieht, indem man an zwei verschiedenen Stellen a_1,a_2 von der rechten Bildkante kommend solange in Richtung der linken Bildseite geht, bis der erste starke Grauwertunterschied zu messen ist. Diese Punkte b_1, b_2 bilden eine Gerade g_1, die mit der Senkrechten im Bild einen Winkel δ bildet, um den sich die Hand drehen muß.

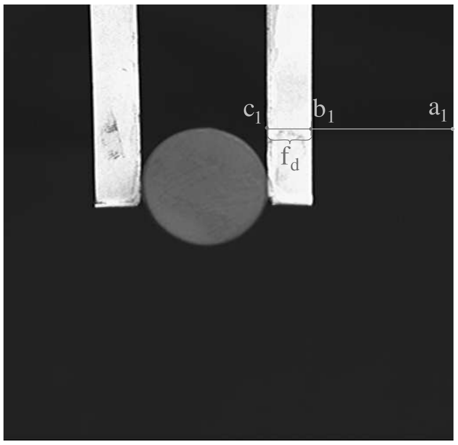
$$\delta = 90° - atan\left(\frac{b_2^y - b_1^y}{b_2^x - b_1^x}\right). \tag{5.16}$$

Die Abbildung 5.4(a) soll dies noch einmal an einem realen Beispiel verdeutlichen. Die Hand muß sich nun an der X-Achse des Toolkoordinatensystems um den Winkel δ drehen, um die gewünschte Ausrichtung zu erhalten. Da die Angabe in Eulerwinkeln gemacht werden muß, ergeben sich aus den Gleichungssystemen (5.10) und (5.12) die Winkel $(yaw = -90°, pitch = \delta°, roll = 90°)$. Der Kreislauf von Messung und Drehung wird so oft wiederholt, bis der gemessene Winkel δ gleich Null ist (Sollwert).

Nachdem die Finger senkrecht im Bild sind, muß nun noch skaliert werden, um eine einheitliche Größe der Finger zu gewähren. Dies geschieht, indem man die Roboterhand entlang des Sehstrahls der Kamera fahren läßt, bis eine bestimmte Dicke des Fingers in Pixeln erreicht ist. Die Dicke f_d des Fingers mißt man, ähnlich wie beim Drehwinkel, indem man ausgehend von Punkt a_1 den Punkt des ersten Grauwertanstieges b_1 (Fingeranfang) und den des nachfolgend ersten Grauwertabstieges c_1 mißt. Die Abbildung 5.4(b) zeigt eine solche Messung. Ist die vorgegebene Solldicke größer als die gemessene Ist-Dicke wird die Hand in Richtung Kamera um ein Inkrement translatiert. Dies geschieht solange, bis die aktuell gemessene Dicke des Fingers der gewünschten Solldicke entspricht (siehe auch Abb. 2.6).



(a) Winkelmessung.



(a) Messung der Fingerdicke.

Abbildung 5.4: (a)Bestimmung des Drehwinkels δ als Schnittwinkel zwischen Bildsenkrechten und der Geraden g_1. (b) Bestimmung der Fingerdicke nach der Drehung für die Objektannäherung.

Die Finger lassen sich nun mit visueller Demonstration aus dem Bild folgendermaßen extrahieren. Man macht zuerst eine aktuelle Aufnahme des Hand, merkt sich die aktuelle Position und läßt sie ganz aus dem Bild fahren, um den Hintergrund aufzunehmen. Dann wird das Hintergrundbild vom vorher aufgenommenen Bild subtrahiert. An dieser Stelle hat man nun die Finger zusammen mit dem Objekt vom Hintergrund extrahiert. Um die Finger zu extrahieren, sucht man im Bild die linke untere Ecke des linken Fingers und die rechte untere Ecke des rechten Fingers. An diese Ecken legt man jeweils eine vorher erstellte Binärmaske, die die jeweilige Fingergestalt hat, und löscht entsprechend der Maske die Pixel im Bild. Die Maske erstellt man in einer frühen Kalibrierungsphase, indem man die Finger ohne Objekt in der Hand in den gleichen normierten Bedingungen mit offenen und geschlossenen Fingern aufnimmt, aus dem mit geeigneten Schwellwerten binarisierten Subtraktionsbild den rechten und linken äußeren Finger mit einen Kontursucher umläuft und die entstandenen Masken abspeichert. Dies muß nur ein einziges mal geschehen. Die Masken können dann immer wieder verwendet werden, da immer wieder die gleichen Bedingungen erzwungen werden können. Die Abbildung 5.5 zeigt den Ablauf der Erstellung der Masken.

(a) Finger offen. (b) Finger geschlossen. (c) binarisiertes Subtraktionsbild mit extrahierten Masken.

Abbildung 5.5: Ablauf der Erstellung der binarisierten Fingermasken für das Löschen der Finger aus dem Bild. Bild (c) entsteht durch Subtraktion des Bildes (b) von Bild (a) und anschließender Binarisierung. Ein Kontursucher geht von der rechten (bzw. linken) Seite zur Bildmitte bis zur ersten Kante und umläuft die Fingerkontur. Die entstandenen und extrahierten Masken sind neben dem Bild zu sehen.

Die Abbildung 5.6 soll noch einmal die Entstehung vom Ausgangsbild zur extrahierten Objektgrundfläche verdeutlichen.

(a) Ausgangsbild. (b) Nach Hintergrundbildsubtraktion und Binarisierung. (c) Nach Löschen der Finger mit Hilfe der Masken.

Abbildung 5.6: Das Löschen der Finger aus dem Bild.

Die eigentliche Kontur läßt sich nun extrahieren, indem man einen von uns implementierten Kontursucher aus dem Mittelpunkt des Bildes in eine Richtung fahren läßt bis zur Grauwertkante, diese Kante entlanggeht und sich jeden Konturpunkt merkt, bis man wieder am Startpunkt ist. Der nächste Abschnitt beschreibt ein Verfahren, wie man die extrahierten Konturen voneinander unterscheiden kann.

5.2 Objektklassifikation

Nachdem die Kontur extrahiert ist, gilt es nun, die Kontur zu verifizieren. Das heißt in unserem Fall, wir wollen wissen, ob es die Grundfläche des Zylinders oder des Quaders ist. Um die Flächen zu unterscheiden, muß zuerst eine geeignete einheitliche Repräsentationsart für die jeweilige Grundfläche erreicht werden. Es gibt viele Möglichkeiten, eine Grundfläche zu repräsentieren. Klassische Beschreibungen sind z.B. die Beschreibung durch Fourierdeskriptoren [Zahn72], durch Momente [Hu62], durch Ketten-Codes (chain-codes) [Freeman77] und durch stochastische Modelle [Kashyap81], die alle gegenüber den anderen Verfahren gewisse Vor- und Nachteile haben. Wir haben uns für eine Repräsentation durch ein stochastisches Modell entschieden, da stochastische Modelle ein gewisses Maß an Robustheit garantieren. Die Vorteile der im folgenden vorgestellten Repräsentationsart sind weiterhin, daß nur ein kleine Menge an Parametern gebraucht wird, um ein komplexes Muster eindeutig zu repräsentieren, und somit auch nur eine kleine Menge an Speicherplatz benötigt wird. Außerdem ist diese Methode sehr schnell.

5.2.1 Grundflächenrepräsentation

Die Grundflächenrepräsentation geschieht in unserem Fall mit Hilfe eines *Autoregressionsmodells* [Dubois86]. Ein Autoregressionsmodell ist eine parametrische Gleichung, die jeden Wert aus einer geordneten Menge von Datenwerten als eine Linearkombination von einer bestimmten Anzahl von vorher erhaltenen Datenwerten plus einem Fehlerterm ausdrückt. In diesem Falle dient das Modell dazu, eine Approximation einer geschlossenen Begrenzung eines 2D-Objektes zu erhalten. Die Begrenzung wird approximiert durch eine geordnete Menge von N Radiusvektorlängen, wobei die Radiusvektoren vom geometrischen Schwerpunkt ausgehend immer im gleichen Winkelabstand $\Delta\phi$ in Richtung Konturbegrenzung gehen. Die Abbildung 5.7 zeigt eine solche Approximation einer Kontur. Die Kontur läßt sich umso genauer darstellen, je größer die Anzahl der Radien wird. Die Länge r des Radiusvektors ist eine Funktion des Projektionswinkels $\phi = t2\pi/N$, mit $t = 1, \cdots, N$, und $r(\phi)$ beschreibt eine eindimensionale Konturapproximation. Die Länge r kann ebenso als eine "Zeitreihe" $r(t)$ ausgedrückt werden, wobei t die Position des Radiusvektors in äquidistanten Winkelinkrementen vom Startpunkt aus beschreibt, z.B. ist $r(1) = r(2\pi/N)$ und $r(N) = r(N \cdot 2\pi/N) = r(2\pi)$. Damit die Funktion an der Stelle (t_i) auch wirklich nur einen Wert liefert, ist diese Darstellungsart nur für konvexe Konturen möglich, d.h. der Radiusvektor darf die Kontur nur an einer Stelle schneiden. [Dubois86] beschreibt noch eine mögliche Erweiterung für Konturen mit Konkavitäten, auf die wir aber nicht weiter eingehen, da es sich bei unseren Grundflächen nur um konvexe Flächen handelt. Mit der obigen Beschreibung läßt sich das Autoregressionsmodell erstellen. Da die Begrenzung zirkulär ermittelt wird, ist die für das Autoregressionsmodell benötigte "Zeitreihe" periodisch, und es gilt $r(t) = r(N + t)$ für $t = 1, \cdots, N$. Sei im folgenden $r(t) = r_t$. Das Modell läßt sich

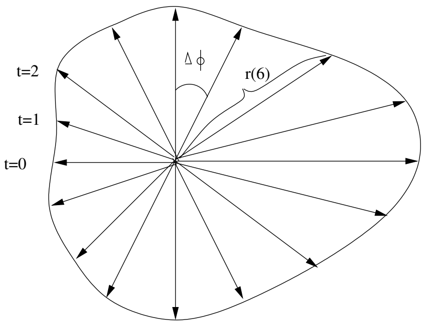


Abbildung 5.7: Approximation der Begrenzung (nach [Dubois86]).

wie folgt beschreiben [Kashyap81]:

$$r_t \;=\; \eta + \sum_{j=1}^{m} \zeta_j r_{t-j} + \sqrt{\rho} w_t, \quad t = 1, \cdots, N. \tag{5.17}$$

Hierbei ist

- r_t die Radiusvektorlänge zum "Zeitpunkt" t, bzw. an der Stelle t,
- r_{t-j} die Radiusvektorlänge die j Radiusvektoren vor r_t gemessen wurde,
- $\zeta_1, \cdots, \zeta_m$ die unbekannten Verzögerungskoeffizienten, die von der beobachteten Zeitreihe errechnet werden sollen, sie bestimmen das Aussehen der Kontur.
- m die Ordnung des Modells, eine geeignete Wahl ist z.B. $m = 3$,
- $\sqrt{\rho} w_t$ der derzeitige Restfehler,
- η eine zu errechnende Unbekannte,
- $\{w_t\}$ eine Menge von unabhängigen Meßwerten mit Varianz gleich eins:
  $E(w_i) = 0, E(w_i w_j) = \rho \delta_{ij};\quad \delta_{ij} = \begin{cases} 1, \text{ falls } & i = j, \\ 0 \text{ sonst.} \end{cases}$

Da die Varianz von w_t eins ist, transformiert der Faktor $\sqrt{\rho}$ die mit der Varianz eins versehene Zufallsvariable w_t zu einer Zufallsvariablen mit der Varianz ρ. Dann kann ρ, die Restvarianz, beschrieben werden als:

$$\rho \;=\; \frac{1}{N} \sum_{t=1}^{N} \left(r_t - \eta - \sum_{j=1}^{m} \zeta_j r_{t-j} \right)^2 . \tag{5.18}$$

Der Parameter η läßt sich folgendermaßen beschrieben:

$$\eta = \bar{r}\left[1 - \sum_{j=1}^{m} \zeta_j\right]. \tag{5.19}$$

Gleichung (5.19) zeigt, daß η proportional zur Durchschnittsvektorlänge $\bar{r}$ ist. Die Koeffizienten $\{\zeta_1, \cdots, \zeta_m\}$ beschreiben die korrelierten Gestaltsvariationen und bestimmen somit, wie oben schon erwähnt, das Aussehen der Kontur.

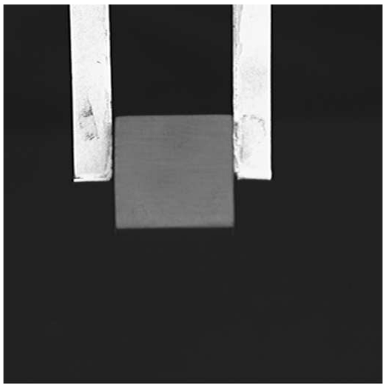
Die Modellparameter werden errechnet, indem man das Modell der beobachteten Zeitreihe $\{r_1, \cdots, r_N\}$ angleicht. Mit Hilfe der Methode der kleinsten Quadrate lassen sich die Modellparameter finden, die den erwartenden Wert des quadratischen Fehlers ρ minimieren sollen.

Die Lösung ist[4]

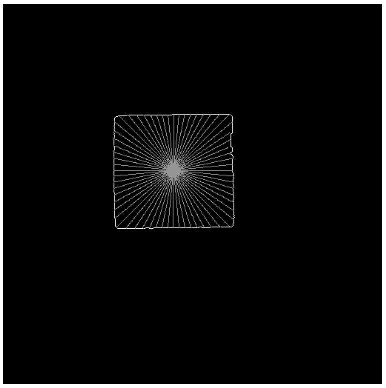
$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_m \\ \eta \end{bmatrix} = \begin{bmatrix} \sum_{t=1}^{N} {r^2}_{t-1} & \cdots & \sum_{t=1}^{N} r_{t-1} r_{t-m} & \sum_{t=1}^{N} r_{t-1} \\ \sum_{t=1}^{N} r_{t-2} r_{t-1} & \cdots & \sum_{t=1}^{N} r_{t-2} r_{t-m} & \sum_{t=1}^{N} r_{t-2} \\ \vdots & \ddots & \vdots & \vdots \\ \sum_{t=1}^{N} r_{t-m} r_{t-1} & \cdots & \sum_{t=1}^{N} {r^2}_{t-m} & \sum_{t=1}^{N} r_{t-m} \\ \sum_{t=1}^{N} r_{t-1} & \cdots & \sum_{t=1}^{N} r_{t-m} & N \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum_{t=1}^{N} r_{t-1} r_t \\ \sum_{t=1}^{N} r_{t-2} r_t \\ \vdots \\ \sum_{t=1}^{N} r_{t-m} r_t \\ \sum_{t=1}^{N} r_t \end{bmatrix}. \tag{5.20}$$

Somit erhält man also für jede Kontur den die Kontur bestimmenden Vektor. Dieses Verfahren ist vergleichbar mit dem *Kalman-Filter*, das ebenso konstruiert ist, daß es Schätzungen des Signals (der Kontur) berechnet und die Abweichungen zwischen Signal und Schätzungen minimiert. Abbildung 5.8 zeigt noch einmal an einem Realbild die entstehenden Konturapproximationen durch die Radienlängen als Eingabewerte für die Autoregression.

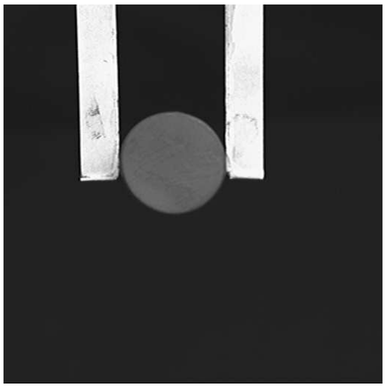
[4] Eine ausführliche Herleitung ist in Anhang A zu finden.



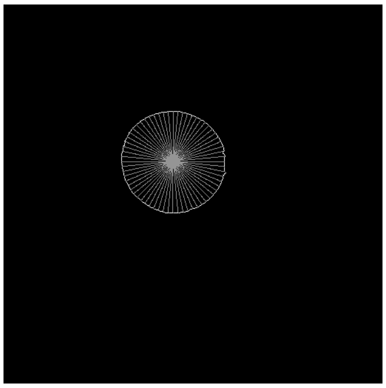
(a) Quader von Vorne.



(b) Konturapproximation der Grundfläche.



(c) Zylinder von Vorne.



(d) Konturapproximation der Grundfläche.

Abbildung 5.8: Originalbilder der Grundfläche des Quaders und des Zylinders und die zugehörigen Konturapproximationen durch Radienlängen (N=64).

Von jeder Objektgrundfläche wird eine bestimmte Menge von Aufnahmen gemacht und die durch Autoregression erstellten Vektoren in einer Datei abgespeichert, so daß jedes Objekt durch eine Menge von Trainingsvektoren repräsentiert wird. Im nächsten Abschnitt wird nun gezeigt, wie ein durch eine neue Aufnahme mit anschließender Autoregression entstandener Vektor mit den Trainingsdaten verglichen werden kann, um ihn als Grundfläche des Quaders oder des Zylinders zu verifizieren.

5.2.2 Grundflächenerkennung

Um die einzelnen Grundflächen zu unterscheiden, benötigt man ein mathematisches Werkzeug, um die jeweiligen Ähnlichkeiten zwischen den Parametern zu messen, die die Grundfläche charakterisieren. Mit Hilfe dieses mathematischen Werkzeuges läßt sich dann ein automatischer Klassifikationsalgorithmus entwickeln.

Die Menge der durch Autoregression entstandenen Modellparameter läßt sich auffassen als

Vektor im sogenannten *Musterraum*. Jeder einzelne Parameter korrespondiert mit einer bestimmten Dimension im Musterraum. Durch die Invarianzeigenschaften des Autoregressionsmodells gegenüber Translation und Rotation der Kontur läßt sich über die Parametervektoren $[\boldsymbol{\zeta}, \eta/\sqrt{\rho}]$, die ähnliche Muster repräsentieren, sagen, daß sie im Musterraum ein eng zusammenliegendes Cluster bilden. Der Abstand eines unbekannten Beispielvektors $\boldsymbol{v}$ und der Trainingsmenge $\{\boldsymbol{q_k}\}$ läßt sich über die Summe der gewichteten euklidischen Abstände des Vektors und der Menge beschreiben. Die Gewichte werden gewählt, um gemeinsame Eigenschaften hervorzuheben und verschiedene Eigenschaften zu unterdrücken. Das k-te optimale Gewicht ist dann klein, wenn die Varianz in dieser korrespondierenden Richtung groß ist, und umgekehrt. Mit Hilfe einer Hauptachsentransformation lassen sich die Ähnlichkeiten der Parameter einer Klasse weiter hervorheben. Die optimal rotierten Koordinatenachsen für eine bestimmte Trainingsmenge zeigen in die Richtung der orthogonalen Eigenvektoren der Beispiel-Kovarianzmatrix für diese Klasse. Die korrespondierenden Eigenwerte sind die Varianzen der Trainingsmenge dieser Klasse in Richtung der neuen Koordinatenachsen. Der Abstand eines unbekannten Beispielvektors $\boldsymbol{v}$ und der Trainingsmenge $\{\boldsymbol{q_k}\}$ der Klasse Q in dem transformierten Musterraum (Eigenraum) der Klasse Q ist somit nach [Dubois86]

$$S(\boldsymbol{v}, \{\boldsymbol{q_k}\}) = \left(\prod_{j=1}^{m+1} \sqrt{\mu_j}\right)^{2/(m+1)} \cdot \left[\left\{\sum_{i=1}^{m+1} \frac{[(\boldsymbol{v} - \bar{\boldsymbol{q}}) \cdot \boldsymbol{e_i}]^2}{\mu_i}\right\} + m + 1\right]. \tag{5.21}$$

Hierbei ist:

- $\bar{\boldsymbol{q}}$ der Durchschnittsvektor der Trainingsmenge $\{\boldsymbol{q_k}\}$,
- μ_i der i-te Eigenwert der Kovarianzmatrix der Trainingsmenge,
- $\boldsymbol{e_i}$ der zugehörige Eigenvektor zu μ_i.

Eine Kontur mit konstantem Abstand zur Klasse Q ist eine Ellipse mit Mittelpunkt $\bar{\boldsymbol{q}}'$, was dem Durchschnittsvektor $\bar{\boldsymbol{q}}$ nach der Transformation entspricht. Die Längen der Hauptachsen der Ellipse sind proportional zu den Eigenwerten der Kovarianzmatrix der Trainingsmenge, und sie sind in die Richtung der Eigenvektoren der Trainingsmengenkovarianzmatrix orientiert. Jeder Eigenwert μ_i ist die Varianz der Beispielpunkte in Richtung des i-ten Eigenvektors $\boldsymbol{e_i}$. Der kleinste Eigenwert korrespondiert mit dem Eigenvektor, der in die Richtung der kleinsten Beispielvarianz zeigt. Die Projektion der Beispielpunkte auf diesen Eigenvektor läßt sich als ein neues Merkmal interpretieren, das das am meisten invariante Merkmal der Beispiele einer Klasse ist.

Die Zugehörigkeit des Beispielvektors zu einer Trainingsmenge wird über den kleinsten Abstand $S(\boldsymbol{v}, \{\boldsymbol{q_k}\})_{min}$ ermittelt. Das heißt also, man mißt von dem unbekannten Vektor $\boldsymbol{v}$ den Abstand $S(\boldsymbol{v}, \{\boldsymbol{q_k}\})$ zur Trainingsmenge $\{\boldsymbol{q_k}\}$ in dem zu der Trainingsmenge $\{\boldsymbol{q_k}\}$ gehörenden Eigenraum. Der Vektor $\boldsymbol{v}$ gehört dann zu derjenigen Trainingsmenge, zu der er den geringsten Abstand hat, in unserem Fall also entweder zur Trainingsmenge der

Grundflächenansicht des Zylinders oder zur Trainingsmenge der Grundflächenansicht des Quaders.

Nach der Erkennung der Grundfläche rotiert die Hand so, daß sie wieder senkrecht zur XY-Ebene des Weltkoordinatensystems steht. Aus der Lage des Objektes in der Hand ergibt sich, daß sich die Hauptachse des Objektes, die durch den Mittelpunkt der Grund- und der Deckfläche geht, senkrecht zur XY-Ebene befindet. Der Kamerakopf zoomt wieder weg, um die gesamte Szene im Auge zu haben. Das nächste Kapitel beschreibt, wie die Annäherung des erkannten Objektes zu seinem Loch verwirklicht wird.

Kapitel 6

Visuell basierte Regelung für die Annäherung

Dieses Kapitel beschreibt nun die zweite von uns genutzte Anwendung des im vierten Kapitel beschriebenen Regelalgorithmus. In dieser Anwendung wird das durch den in Kapitel fünf beschriebenen Algorithmus erkannte Objekt dem für das Objekt bestimmten Loch angenähert. Die hierfür zuerst benötigte Erkennung des Lochbretts und der Löcher wird in Abschnitt 6.3 beschrieben[1].

Um das Objekt an das Loch anzunähern, benötigt man vor jedem Annäherungsschritt den Istwert im Bild. Der Istwert für die Regelung wird beschrieben durch einen markanten Punkt am Objekt. Die Objekterkennung ist also reduziert auf die Bestimmung eines ausgezeichneten Punktes im rechten und linken Bild. Dies hat den Vorteil, ein rechenaufwendiges und zeitraubendes Stereomatchingverfahren ausschließen zu können. Im ersten Abschnitt dieses Kapitels wird beschrieben, wie man diesen markanten Punkt am Quader im Bild vor jedem Annäherungschritt wiederfindet[2]. Der zweite Abschnitt zeigt die Extraktion des jeweiligen Istwertes für den Zylinder. Der dritte Abschnitt beschreibt, wie oben schon erwähnt, die Extraktion des Lochbrettes und der Löcher und eines markanten Punktes am Loch, also implizit damit auch den Sollwert für die Regelung, auf die im vierten Abschnitt noch einmal eingegangen wird.

[1]Obwohl die Erkennung der Löcher zeitlich vor der Erkennung des sich an das Loch annähernden Objektes stattfindet, wird sie erst später beschrieben, da es so inhaltlich besser in das Gesamtkonzept der Arbeit paßt, denn es werden hierfür Algorithmen aus den Bereichen Kantenextraktion und Ellipsenextraktion angewendet, die sowohl in dem Abschnitt der Istwertbestimmung beim Quader, als auch in dem Abschnitt der Istwertbestimmung beim Zylinder vorkommen.

[2]Da bei dem Quader für eine paßgerechte Annäherung das zusätzliche Problem vorliegt, daß eine Verdrehung des Quaders gegenüber der Lochkontur möglich ist, muß sich neben dem markanten Punkt als Istwert noch der Istwert der möglichen Verdrehung des Objektes gemerkt werden, so daß man insgesamt von einem Istzustand sprechen kann. Eine Lösung des Problems der Verdrehung wird in Abschnitt 6.3 beschrieben.

6.1 Istwertbestimmung beim Quader

Dieser Abschnitt zeigt, wie man einen markanten Punkt des Quaders, also den Istwert für die Regelung, aus dem Bild extrahiert. Als markanter Punkt am Quader eignet sich sehr gut der Mittelpunkt des vorderen Berandungssegmentes der Grundfläche, da wir dieses Segment immer in beiden Bildern sehen können, und damit natürlich auch den Mittelpunkt. Die Abbildung 6.1 soll die Lage des Segmentes und des Punktes noch einmal veranschaulichen.

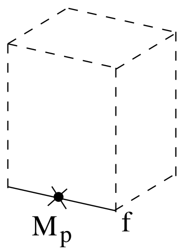


Abbildung 6.1: Zu suchendes Segment f mit Mittelpunkt M_p als Istwert für die Regelung.

Um diesen markanten Punkt des Quaders zu extrahieren, benötigt man zwei Dinge. Zum einen braucht man allgemein die Extraktion von Berandungssegmenten aus dem Bild und zum anderen die Erkennung des Berandungssegmentes, deren Mittelpunkt unser gesuchter markanter Punkt ist. Der nächste Abschnitt soll zunächst zeigen, wie die Berandungssegmente aus einem Bild extrahiert werden können.

6.1.1 Kantenextraktion

Dieser Abschnitt gliedert sich in vier Teile. Zunächst wird ein allgemeines Verfahren zur Extraktion von Geraden, nämlich die *Standard-Hough-Transformation* [Hough62] beschrieben. Im zweiten Teil wird dieses Verfahren verbessert zur *orientierungselektiven Hough-Transformation*. Im dritten Teil zeigen wir eine geeignete Extraktion der Maxima, die bei der Hough-Transformation entstehen. Sie beinhalten die Parameter der Geraden. Im vierten Teil dieses Abschnittes beschreiben wir schließlich, wie man aus den gefundenen Geraden Berandungssegmente im Bild erhält. Zunächst wird kurz eine Einführung in die Hough-Transformation für Geraden gegeben. Für eine weiterführende Diskussion, welche Hough-Transformation unter welchen Bedingungen die sinnvollste ist, sei hiermit auf [Leavers92] verwiesen, der in seinem Artikel auch eine gute Literaturliste aufführt. Es sei eine Menge von Bildpunkten $(x_1, y_1)...(x_n, y_n)$ betrachtet, die auf einer Geraden mit Steigung a' und Achsenabschnitt b' liegen und somit die Geradengleichung $y = a'x + b'$ erfüllen. Angenommen sei nun, daß man die Parameter (a', b') nicht weiß. Formt man eine Geradengleichung um zu $b = -xa + y$ und läßt a laufen, erhält man im nun betrachteten Parameterraum (man betrachtet die ab Ebene) für jeden Punkt (x_i, y_i), $i = 1, ..., n$ aus der

Punktemenge eine Gerade. Es entsteht eine Geradenschar, deren gemeinsamer Schnittpunkt genau (a', b') ist. Die Abbildung 6.2 soll dieses Konzept noch einmal an zwei aus der Punktemenge gewählten Punkten verdeutlichen.

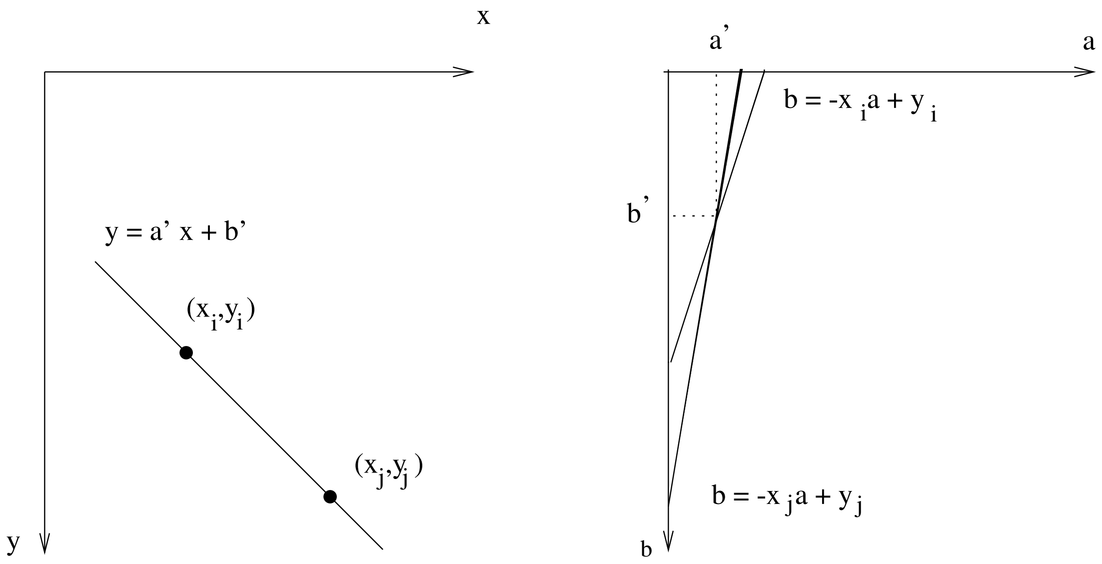


Abbildung 6.2: Betrachtung zweier ausgewählter Punkte einer Geraden im Bildraum (links) und im (a,b)-Parameterraum (rechts).

Für die Hough-Transformation wird der Parameterraum in sogenannte *Akkumulatorzellen* unterteilt, beginnend bei (a_{min}, b_{min}) und endend bei (a_{max}, b_{max}), den zu erwartenden Bereich für Steigung und Achsenabschnitt. Jede Zelle (i, j) korrespondiert mit den Parameterraumkoordinaten (a_i, b_j), und erhält anfangs den Wert $A(i, j) = 0$. Für jeden Punkt (x_i, y_i) im Bildraum wird nun bei laufendem a der Parameter b gemäß der Gleichung $b = -x_i a + y_i$ ermittelt und die zugehörige Akkumulatorzelle jeweils um 1 erhöht. Das Maximum aller Akkumulatoren beinhaltet am Schluß die gesuchten Parameter der Gerade.

Ein Problem tritt aber auf, wenn die Gerade im Bildraum z.B. parallel zur vertikalen y-Achse läuft und somit eine unendliche Steigung besitzt. Der Parameterraum müßte hierbei auch unendlich gewählt werden, um den gesuchten Schnittpunkt zu ermitteln, was nicht praktikabel ist. Um dieses Problem zu umgehen kann man die Geradengleichung in Polarform verwenden:

$$x \ \cos\theta + y \ \cos\theta \ = \ r. \tag{6.1}$$

Die beiden Parameter r und θ repräsentieren zum einen den Abstand der Geraden zum Mittelpunkt des Bildes gemessen an der Normalen zur Geraden und zum anderen den Winkel dieser Normalen zur x-Achse (siehe Abb. 6.3).

Das Verfahren, das für diese Geradenrepräsentation angewendet wird, ist das gleiche, wie für die Steigung-Achsenabschnittsform, nur daß hier anstelle von Geraden im (θ, r)-Parameterraum sinusoidale Kurven entstehen (siehe Abb. 6.4). Der Parameter r hat hier

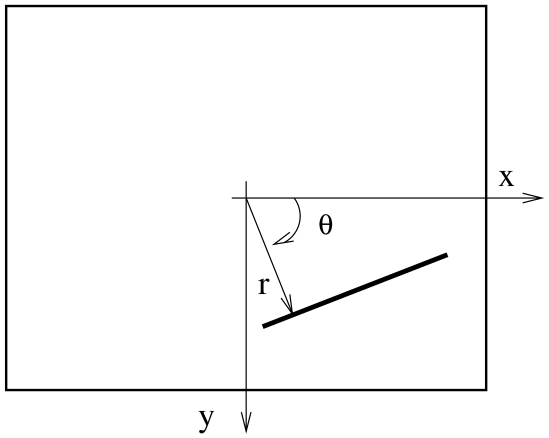
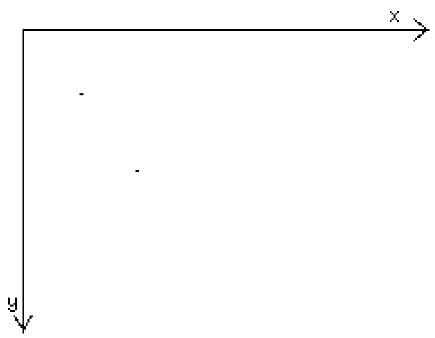
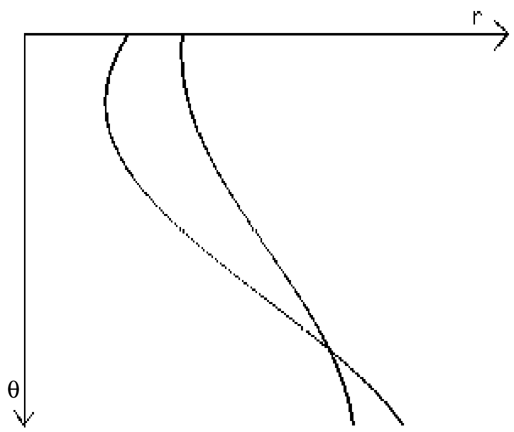


Abbildung 6.3: Gerade in Polarform.

die Größe von $-r_{max}$ bis r_{max}, wobei r_{max} die Hälfte der Bilddiagonalen ist. Der Parameter θ läuft von 0 bis 180 Grad. Nach vollständig vollzogener Standard-Hough-Transformation beschreiben die Koordinaten der Akkumulatorzelle mit dem höchsten Wert die zur Gerade zugehörigen Parameter r und θ.



(a) Bildraum.



(b) (r, θ)-Parameterraum.

Abbildung 6.4: Betrachtung zweier Punkte im Bildraum (links) und die zugehörigen Sinusoidalkurven im (r, θ)-Parameterraum (rechts). Um nur positive Werte zu erhalten, sind die Kurven um r_{max} nach rechts verschoben.

Der Vorteil der Hough-Transformation ist, daß eine Gerade (bzw. ein *Berandungssegment* eines Objektes) aus einem Bild extrahiert werden kann, obwohl die Kontur der Gerade (bzw. des Segmentes) im Kantenbild nicht durchgehend sein muß. Außerdem müssen die Segmente, wie bei anderen klassischen Verfahren, nicht vorher voneinander segmentiert werden (vgl. Kap. 6.2.1). Die Probleme, die bei der Extraktion von Berandungssegmenten durch die Hough-Transformation auftreten, werden im folgenden beschrieben. Außerdem wird eine Lösung für jedes Problem vorgestellt.

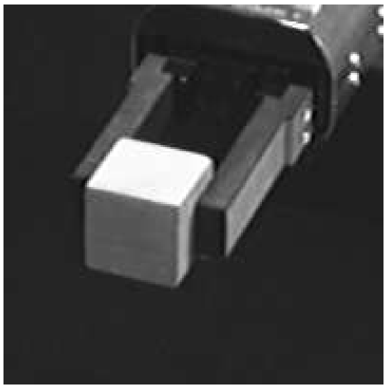
Da eine Hough-Transformation für Geraden auf ein Kantenbild vorgenommen wird, muß erst einmal ein solches aus dem von der Kamera aufgenommenen Grauwertbild erstellt werden. Ein mögliches Verfahren ist die Erstellung eines durch einen geeigneten Schwellwert binarisierten Gradientenbetragsbildes. Hierbei wird die Bildfunktion $b(x, y)$ durch einen Ableitungsoperator, wie z.B. den Sobeloperator, nach x bzw. nach y abgeleitet. Der Sobeloperator hat folgendes Aussehen

$$S_x = \frac{1}{8}\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad S_y = \frac{1}{8}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

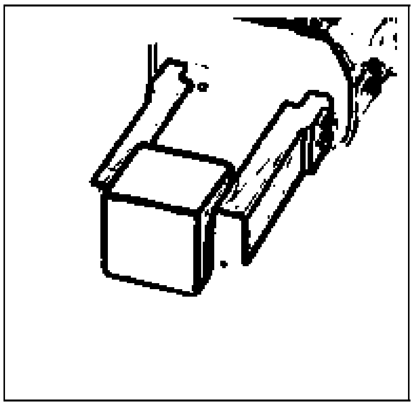
Nach der horizontalen Ableitung mit S_x entsteht Δb_x und nach der vertikalen Ableitung mit S_y entsteht Δb_y. Der Betrag des Gradienten $g(x, y)$ läßt sich wie folgt ermitteln

$$g(x,y) \;=\; \sqrt{\Delta {b_x}^2 + \Delta {b_y}^2}. \tag{6.2}$$

Mit Hilfe eines Schwellwertes läßt sich dieses Bild nun binarisieren (siehe Abb. 6.5).



(a) Originalbild.



(b) Binarisiertes Gradientenbetragsbild.

Abbildung 6.5: Beispielbild mit zugehörigem Gradientenbetragsbild.

Ein Problem hierbei ist die mögliche Entstehung breiter Kanten, die sich unter Umständen fatal auf die Hough-Transformation auswirken, da hier "falsche" Kanten die gleiche Summe im Akkumulatorfeld erwirken können, wie die "gewünschte" Kante. Dies soll Abbildung 6.6 an einem vergrößerten Ausschnitt eines Kantenbildes verdeutlichen.

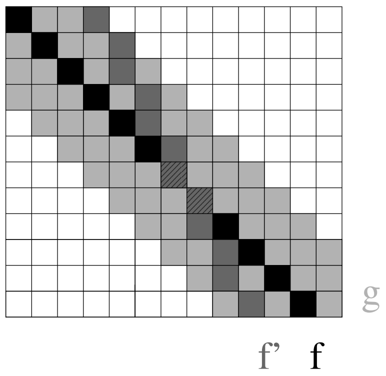


Abbildung 6.6: Problem der Hough-Transformation bei breiten Kanten (hier der Deutlichkeit halber übertrieben): Nach der Hough-Transformation aller Pixel von der breiten Kante g hat die Akkumulatorzelle für die Parameter von f' dasselbe Maximum wie die Zelle für die gewünschte Gerade f, da gleich viele Pixel für f' votiert haben. Es wird nach der Maximaextraktion der Pixel von der breiten Kante g sowohl die Gerade f, als auch die Gerade f' rückprojiziert.

Eine Abhilfe schafft eine geeignete *Ausdünnung* des unbinarisierten Gradientenbetragsbildes. Dies geschieht mit Hilfe der Unterdrückung von Nichtmaxima [Jain95]. Dieses Verfahren dünnt den entstandenen Hügel, den eine Kante im Gradientenbetragsbild bildet, so aus, daß nur noch das Maximum eines Hügels übrigbleibt. Hierfür muß zusätzlich zum Gradientenbetragsbild noch das Gradientenorientierungsbild ermittelt werden:

$$\Theta(x, y) \;=\; \arctan\left(\frac{\Delta b_x}{\Delta b_y}\right). \tag{6.3}$$

Abbildung 6.8(a) zeigt ein solches Gradientenorientierungsbild. Um nur Werte von 0 bis π zu erhalten, wird auf negative Werte π addiert. Jedes Pixel wird dann je nach Gradientenwinkel in einem von vier möglichen Sektoren unterteilt (siehe Abb. 6.7).

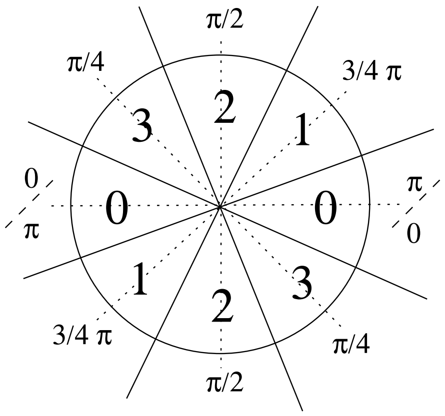
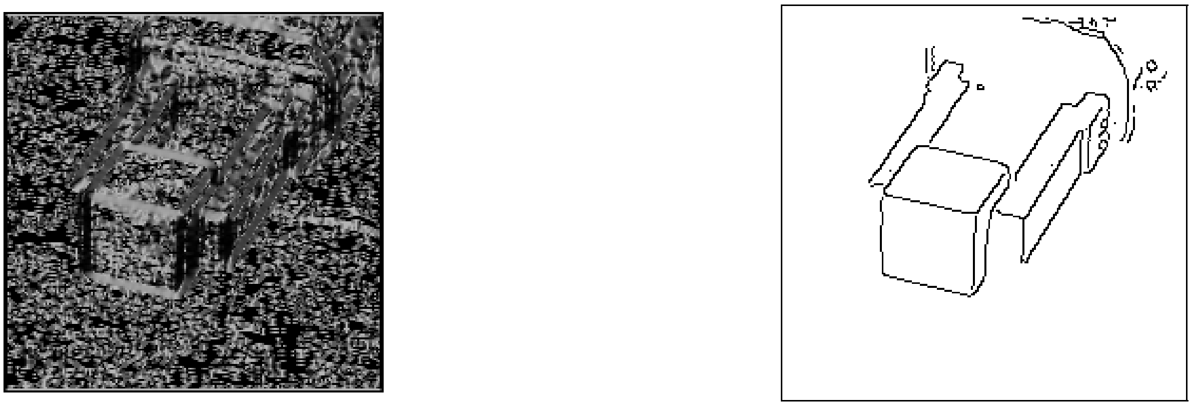


Abbildung 6.7: Einteilung eines Pixels im Gradientenorientierungsbild in einem von vier Sektoren (nach [Jain95]).

Für die Ausdünnung läuft man durch das Gradientenbetragsbild mit einer virtuellen 3x3 Nachbarschaftsmaske und vergleicht den Aufpunkt (das Zentrum) mit den sich aus den Sektorwert des Aufpunktes ergebenden Nachbarn. Hat der Aufpunkt z.B. eine Gradientenorientierung von $\pi/4$, also den Sektorwert 3, wird er mit dem linken oberen und dem rechten unteren Nachbarn verglichen. Falls der Aufpunkt keinen größeren Gradientenbetrag als beide Nachbarn hat, wird er gleich Null gesetzt. Dieser Prozeß garantiert das Ausdünnen des breiten Hügels im Gradientenbetragsbild auf eine Breite von einem Pixel, so daß nach der Binarisierung durch einen geeigneten Schwellwert ein ausgedünntes Kantenbild entsteht (siehe Abb. 6.8(b)).

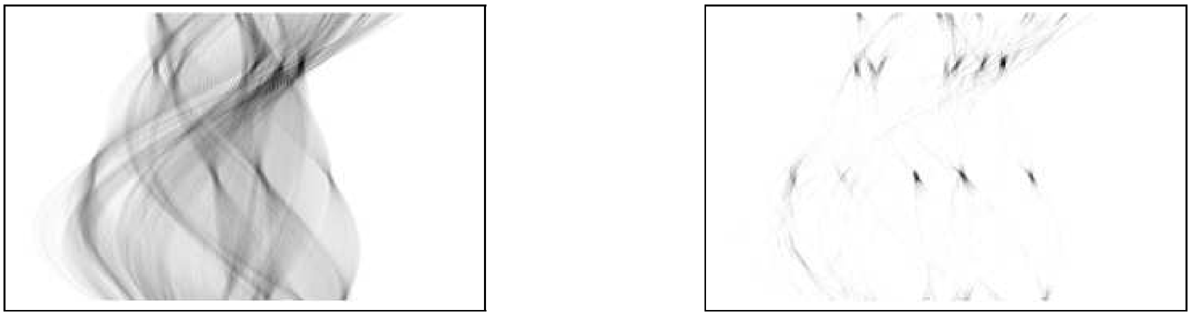


(a) Gradientenorientierungsbild.

(b) Ausgedünntes Gradientenbetragsbild.

Abbildung 6.8: Das Gradientenorientierungsbild und das ausgedünnte Gradientenbetragsbild zum Beispielbild 6.5(a). Zur Veranschaulichung wurde das Gradientenbetragsbild hier derart verändert, daß die Winkel $0-\pi/2$ durch die Grauwerte $0-180$ repräsentiert werden, und die Winkel $\pi/2-\pi$ wieder absteigend von den Grauwerten $180-0$ repräsentiert werden, sonst würde man senkrechte Kanten schlecht sehen.

Die Errechnung der Kantenorientierung birgt gegenüber der Hough-Transformation einen weiteren Vorteil, der darauf beruht, daß ein Punkt einer Grauwertkante etwa dieselbe errechnete Gradientenorientierung hat, wie die Kante selber. Mit dieser Eigenschaft läßt sich die Standard-Hough-Transformation in eine *orientierungsselektive Hough-Transformation* [Pauli97] umwandeln. Hierbei werden für einen Pixel (x_i, y_i) nicht mehr alle Akkumulatorzellen erhöht, die die entstehende Sinusoidalkurve im Parameterraum durchläuft, sondern nur noch die, dessen vertikaler Index der Gradientenorientierung dieses Pixels entspricht. Da aber bei der Berechnung der Gradientenorientierung Ungenauigkeiten entstehen, ist es sinnvoll, einen Toleranzbereich δt einzuführen, so daß alle Akkumulatorzellen im Bereich von $\Theta(x_i, y_i) - \delta t$ bis $\Theta(x_i, y_i) + \delta t$ erhöht werden. Hierbei entstehen schärfere Peaks im Houghbild. Dies soll Abbildung 6.9 verdeutlichen.



(a) Standardhoughtransformation zum Beispielbild.

(b) Orientierungsselektive Hough-Transformation.

Abbildung 6.9: Vergleich zwischen Standard- und orientierungsselektiver Hough-Transformation.

Ein großes Problem der Hough-Transformation ist die geeignete *Extraktion der Maxima* im Houghbild, da aufgrund von Diskretisierungsfehlern trotz der orientierungsselektiven Hough-Transformationen keine scharf abgegrenzten Spitzen entstehen[Veen80]. Es ensteht vielmehr ein Hügel im Houghbild pro Gerade mit einem globalen Maximum und lokalen Maxima in der Nachbarschaft. Es wäre nun falsch bei einer zu erwartenden Geradenanzahl von n Geraden - wenn man einmal davon absieht, daß man die zu erwartende Geradenanzahl sowieso nicht weiß - n Maxima aus dem Houghbild zu extrahieren. Denn lokale Maxima eines Hügels können größer sein als ein globales Maximum eines anderen Hügels, diese würden dann also gegenüber der gewünschten Gerade bevorzugt werden und das Ergebnis verfälschen. Um diesem Problem entgegenzuwirken, geben wir jedem Hügel im Houghbild, d.h. jeder Zusammenhangskomponente, die größer als ein bestimmter Schwellwert ist, ein anderes Label (siehe Abb. 6.10) und extrahieren nur ein Maximum pro Label.

Bemerkung:

Bei der Benennung der Hügel ist zu beachten, daß der obere und untere Rand des Houghbildes aufgrund der sinusoidalen Eigenschaft der Kurven zyklisch gespiegelt weitergeht (ähnlich einem Möbiusband), so daß dementsprechend auch die Benennung der Hügel gleichbleiben muß.

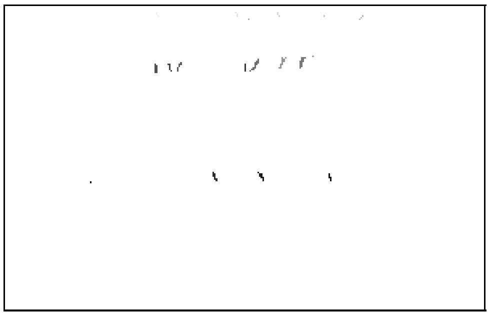
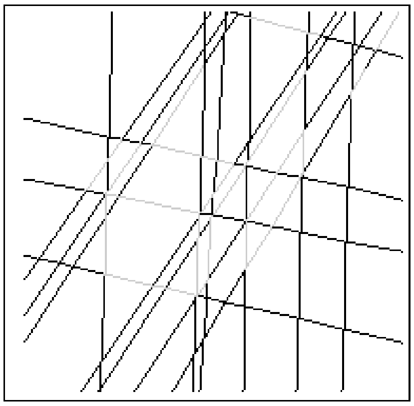
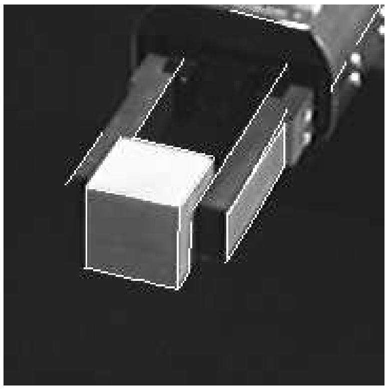


Abbildung 6.10: Gelabeltes Houghbild: jede Zusammenhangskomponente ist mit einem anderen Grauwert kodiert.

Nach der Extraktion der Maxima erhält man für jedes Maximum die Parameter einer (virtuellen) Gerade, die nach der Rückprojektion in den Bildraum (bzw. auf das Kantenbild) über das gesamte Bild geht. Uns interessiert aber nur das jeweilige Geradensegment, das unser Objekt berandet. Zwar beschreibt z.B. [Atiquzzaman94] ein Verfahren, das den Anfangs- und Endpunkt eines Segmentes und deren Länge *direkt* aus dem Houghbild ermittelt, leider ist es ungeeignet für die gleichzeitige Ermittlung mehrerer Geraden. Wir benutzen daher eine andere Vorgehensweise. Dafür werden zunächst für jede aufgrund des Houghbildes ermittelte Gerade die Schnittpunkte mit allen anderen gefundenen Geraden und dem Bildrand ermittelt, so daß eine Gerade in n Segmente unterteilt wird. Wenn man nun innerhalb eines Segmentes von einem Schnittpunkt zum nächsten läuft, läßt sich wieder die Eigenschaft ausnutzen, daß ein Punkt einer Grauwertkante dieselbe errechnete Gradientenorientierung hat, wie die Kante selber. Jedes Pixel des Segmentes, das aufgrund der Maximafindung eine bestimmte Gradientenorientierung hat, läßt sich mit dem jeweiligen Pixel des Gradientenorientierungsbildes vergleichen. Darüberhinaus läßt es sich auch mit dem jeweiligen Pixel des Kantenbildes vergleichen. Nur wenn ein bestimmter Prozentsatz an Übereinstimmungen sowohl mit den jeweiligen Pixeln des Gradientenorientierungsbildes, als auch mit den Pixeln des Kantenbildes überschritten wird, existiert tatsächlich ein Berandungssegment im Bild. Falls zwei oder mehr verbundene Berandungssegmente hintereinander auf einer Geraden gefunden werden, werden sie zu einem Segment verschmolzen. Die Abbildung 6.11 zeigt die rücktransformierten Houghgeraden, und die gefundenen Segmente im Originalbild.



(a) Rückprojizierte Houghgeraden.



(b) Gefundene Kantensegmente.

Abbildung 6.11: (a) Rückprojizierte Houghgeraden mit gefundenen Segmenten, (b) gefundene Segmente im Beispielbild.

Zusammenfassend läßt sich folgende Vorgehensweise für unsere Kantenextraktion skizzieren:

Schema der Kantenextraktion

- Ableitung des Bildes in x- und y- Richtung
- Ermittlung des Gradientenbetrags- und Gradientenorientierungsbildes
- Ausdünnen des Gradientenbetragsbildes mittels des kodierten Gradientenwinkelbildes
- Erstellung eines Kantenbildes mittels Binarisierung durch eine Schwellwertoperation
- Orientierungselektive Hough-Transformation
- Labeln des Hough-Transformationsbildes über Zusammenhangskomponenten, die bei Setzung einer Schwelle entstehen
- Ermittlung der Maxima im Houghbild mit Hilfe des gelabelten Houghbildes
- Rückprojektion in den Bildraum
- Ermittlung aller Schnittpunkte der rückprojizierten Geraden, um Segmente zu erhalten
- Verifikation des Segmente mittels Kantenbild und Gradientenorientierungsbild

Der nächste Abschnitt beschreibt, wie wir aus den extrahierten Berandungssegmenten dasjenige herausfinden, das den gesuchten markanten Punkt als Mittelpunkt hat.

6.1.2 Erkennung von Berandungssegmenten

Die einzelnen Geradensegmente, die man aus dem Bild extrahiert hat, müssen nun noch über eine geeignete Methode in Verbindung zueinander gebracht werden, um nur relevante Berandungssegmente, also Segmente des Quaders, zu erhalten. Aus diesen Segmenten wollen wir dann unser gesuchtes Segment extrahieren, dessen Mittelpunkt Mp unser gesuchter Istwert für die Regelung ist. Da wir ein kleines Objekt von einem relativ großen Kameraabstand betrachten, lassen sich bei der Projektion des Quaders auf die Bildebene einige geometrische Quasi-Invarianzen ([Binford93] und [Mundy92]) der Berandungssegmente gegenüber der Projektion ermitteln, die wir ausnutzen wollen. Segmente, die im 3D Koordinatensystem die gleiche Länge und Orientierung haben, haben auch in der Projektion (unter den uns gegebenen Umständen) eine ähnliche Orientierung und Länge (siehe Abb. 6.12).

Geht man von der Annahme aus, daß der Quader homogene Seitenflächen hat und sich die Beleuchtung während der Annäherung nicht ändert, läßt sich eine weitere Quasi-Invariante

$$|f_1| = |f_2| = |f_3|$$
$$O(f_1) = O(f_2) = O(f_3)$$
$$|g_1| = |g_2| = |g_3|$$
$$O(g_1) = O(g_2) = O(g_3)$$
$$|h_1| = |h_2| = |h_3|$$
$$O(h_1) = O(h_2) = O(h_3)$$

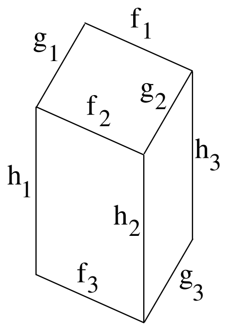


Abbildung 6.12: Quasi-Invarianzen der Berandungssegmente gegenüber Drehung und Translation des Quaders: die Länge und Orientierungen der Kanten f_i,g_i und h_i untereinander sind sehr ähnlich.

erkennen, die von Aufnahme zu Aufnahme gilt. Die Grauwertstruktur der Flächen innerhalb des Quaders ändern sich nur innerhalb einer geringen Toleranzgrenze, so daß man für jedes Berandungssegment einen durchschnittlichen benachbarten Grauwert betrachten kann, falls die benachbarte Seite des Segmentes sich im Inneren des Quaders befindet. In der Abbildung 6.12 bleibt z.B. der untere Grauwert von f_1 von Aufnahme zu Aufnahme unverändert. Ebenso der untere und obere von f_2 usw. Für ein Berandungssegment s_i , das aus N zusammenhängenden Konturpunkten (p_j^x, p_j^y) besteht, läßt sich folgender oberer benachbarter Durchschnittsgrauwert GW^o bzw. unterer benachbarter Durchschnittsgrauwert GW^u aus dem Grauwertbild ermitteln:

$$\begin{aligned} GW^o(s_i) &= \frac{1}{N}\sum_{j=1}^{N} gw(p_j^x, p_j^y - 1), \\ GW^u(s_i) &= \frac{1}{N}\sum_{j=1}^{N} gw(p_j^x, p_j^y + 1). \end{aligned} \tag{6.4}$$

Die Abbildung 6.13 zeigt an einem Beispiel die oberen und unteren Grauwerte des Berandungssegmentes f_2, die zur Ermittlung von $GW^o(f_2)$ und $GW^u(f_2)$ aus dem Grauwertbild herangezogen werden.

Macht man nun initial einmal eine Aufnahme von dem Quader alleine mit homogenem Hintergrund, lassen sich aufgrund dieser *visuellen Demonstration* über die benachbarten Durchschnittsgrauwerte der extrahierten Kanten dieser Aufnahme der Grauwert der Deckfläche des Quaders, der im folgenden mit gw^{DF} bezeichnet wird, und der Grauwert der vorderen Fläche gw^{VF} bestimmen, deren Werte wir uns für den Regelungsvorgang merken. Damit sich die Grauwerte der Deckfläche und der Vorderfläche nicht zu stark ändern muß die Annahme gemacht werden, daß sich die Beleuchtungsverhältnisse während der Aufnahmen nur in einem geringen Toleranzbereich ändern.

Da die Deckfläche des Quaders immer zu sehen ist, bestimmen wir bei jedem Annäherungsschritt zuerst das vordere und hintere Berandungssegment der Deckfläche. Das sind die Segmente f_1 und f_2 der Abbildung 6.12. Um nun die beiden Berandungssegmente (f_1, f_2)

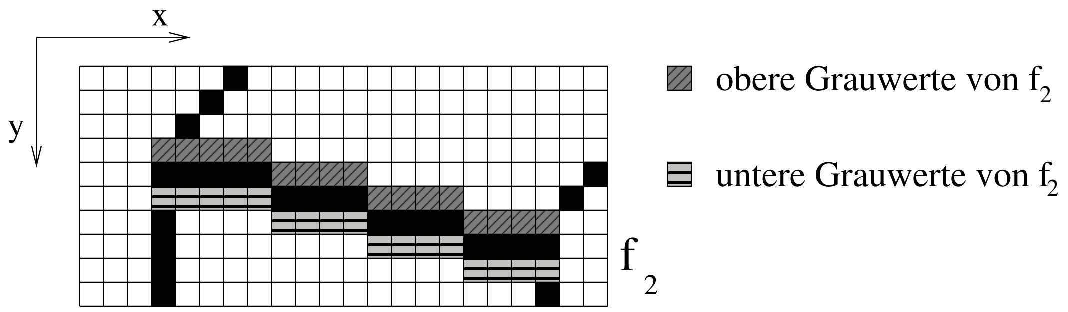


Abbildung 6.13: Obere und untere Grauwerte des Berandungssegmentes f_2.

aus der Menge aller extrahierten Segmente, die im folgenden die Größe M haben soll, zu verifizieren, läßt sich folgende Gleichung anwenden. Hierbei seien $O(s)$ und $L(s)$ die Orientierung und Länge von Segment s. Die Variable $Mp^y(s)$ beschreibt die y-Komponente des Mittelpunktes vom Segment s.

$$\begin{aligned}(f_1, f_2) := \arg \big\{ s_i, s_j \big| \quad & (i \neq j) \wedge (O(s_i) \approx O(s_j)) \wedge (L(s_i) \approx L(s_j)) \wedge (GW^u(s_i) \approx gw^{DF}) \\ & \wedge (GW^o(s_j) \approx gw^{DF}) \wedge (GW^u(s_j) \approx gw^{VF}) \\ & \wedge (Mp^y(s_i) < Mp^y(s_j));\ i, j, = 1, ..., R \big\}. \end{aligned} \tag{6.5}$$

Das Zeichen "$\approx$" soll hier die Gleichheit bis auf eine vom Benutzer jeweilig anzugebende Fehlerschranke δ_i anzeigen.

Nachdem man die beiden Berandungssegmente f_1 und f_2 hat, läßt sich unser gesuchtes Berandungssegment f_3, über dessen Mittelpunkt wir unseren gesuchten markanten Punkt erhalten, aus der Menge aller extrahierten Segmente wie folgt bestimmen. Wir unterscheiden hierbei die Ermittlung des Segmentes bei der ersten Aufnahme von der Ermittlung bei allen anderen Aufnahmen. Bei der ersten Aufnahme wird f_3 mit Hilfe von f_2 ermittelt durch:

$$\begin{aligned} f_3 := \arg \big\{ s_k \big| \quad & (s_k \neq f_2) \wedge (O(s_k) \approx O(f_2)) \wedge (L(s_k) \approx L(f_2)) \wedge (GW^o(s_k) \approx gw^{VF}) \\ & \wedge (Mp^y(s_k) < Mp^y(f_2)); k, = 1, ..., R \big\}. \end{aligned} \tag{6.6}$$

Außerdem wird danach der euklidische Abstand $\boldsymbol{abst} = (abst^x, abst^y)$ zwischen den Mittelpunkten von f_2 und f_3 ermittelt.

Bei jeder weiteren Aufnahme dient dieser Abstand als Suchbereichseinschränkung. Wir suchen bei den weiteren Aufnahmen nach dem Segment f_3 also nur in Abstand $abst$ von f_2. Dafür erweitern wir nach der ersten Aufnahme Gleichung (6.6) noch um folgende Bedingung:

$$\begin{aligned} f_3 \quad := \quad & \arg \big\{ s_k \big| \ldots \wedge ((Mp^y(f_2) + abst^y \approx Mp^y(s_k)) \wedge ((Mp^x(f_2) + abst^x \approx Mp^x(s_k)); \\ & k = 1, ..., R \big\}. \end{aligned} \tag{6.7}$$

Falls diese Bedingung erfüllt ist, wird für die nächste Aufnahme wieder der aktuelle Abstand zwischen f_2 und f_3 bestimmt. Falls diese Bedingung nicht erfüllt ist[3], läßt sich die Symmetrieeigenschaft des Quaders nutzen, und man nimmt das um *abst* verschobene Segment f_2 als neues f_3. Somit erhält man auch einen markanten Punkt, ohne das untere Segment gefunden zu haben. Falls sich schon die Segmente f_1 und f_2 selber nicht ermitteln ließen, wie z.B bei einer temporären Störung während der Aufnahme, wird die gesamte Aufnahme wiederholt. Die Abbildung 6.14 zeigt noch einmal das Beispielbild mit den gefundenen Kantensegmenten und den durch die oben beschriebenen Gleichungen gefundenen markanten Punkt.

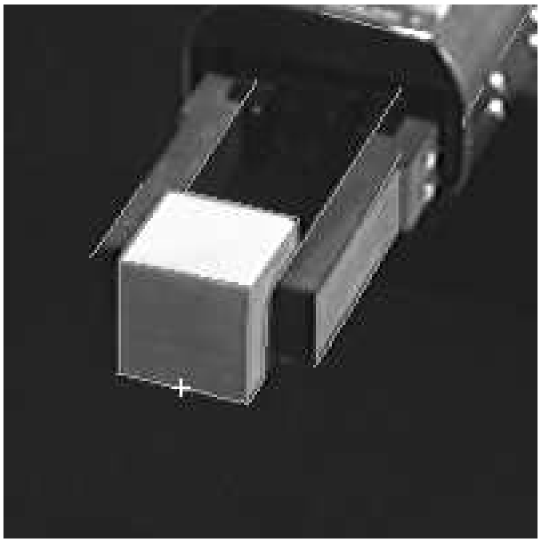


Abbildung 6.14: Beispielbild mit gefundenen Segmenten und erkanntem markanten Punkt.

Der nächste Abschnitt zeigt, wie man diesen markanten Punkt beim Zylinder finden kann.

6.2 Istwertbestimmung beim Zylinder

Wie auch bei dem Quader besteht die Bestimmung des Istwertes des Zylinders aus zwei Teilen. Da wir nämlich wieder einen markanten Punkt der Grundfläche des Objektes als Istwert nehmen wollen, müssen wir zum einen Ellipsen aus den Bildern extrahieren und zum anderen die relevante Halbellipse verifizieren, die man als Teil der Grundfläche sieht. Für den dann zu extrahierenden markanten Punkt ist zu beachten, daß sich der Konturmittelpunkt der sichtbaren Halbellipse des Zylinders in unseren Fall nicht eignet. Wir betrachten den Zylinder nämlich mit beiden Kameras aus zwei verschiedenen Blickwinkeln, so daß der jeweilige Mittelpunkt der sichtbaren Halbellipse von einem anderen Punkt des

[3] Dies kann z.B. bei einer Überdeckung unseres gesuchten unteren Berandungssegmentes durch das Lochbrett sein.

Zylinders stammt. Um trotzdem einen eindeutigen markanten 3D Punkt im rechten und und linken Bild zu verifizieren, nehmen wir denjenigen Konturpunkt, der dieselbe Tangenstensteigung hat wie das vordere Berandungssegment des Lochbrettes. Dies wird im Abschnitt 6.2.2 genau beschrieben. Die Abbildung 6.15 soll die Lage der Halbellipse mit dem markanten Punkt noch einmal veranschaulichen.

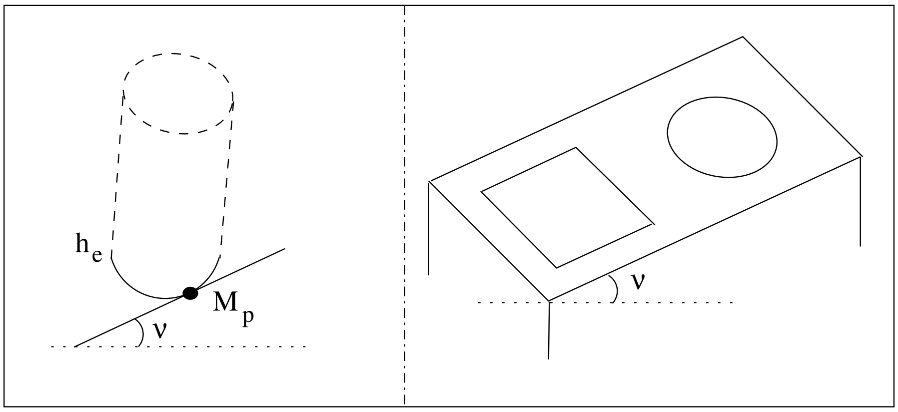


Abbildung 6.15: Zu suchende Halbellipse h_e mit markantem Punkt M_p dessen Tangentensteigung dieselbe ist wie die Orientierung des vorderen Berandungssegmentes des Lochbrettes.

Zunächst zeigen wir im folgenden Abschnitt den ersten Teil für die Suche nach dem markanten Punkt des Zylinders, die Ellipsenextraktion.

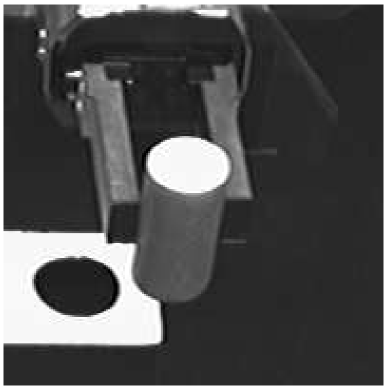
6.2.1 Ellipsenextraktion und -approximation

Auch für die Ellipsenextraktion existiert eine Hough-Transformation, die aber nicht auf einen zweidimensionalen Parameterraum (θ, r) beschränkt ist, sondern eine Maximumsuche in einem fünfdimensionalen Parameterraum erfordert, denn eine Ellipse wird durch ihren Mittelpunkt (x_0, y_0), ihre Achsen (a, b) und ihren Rotationswinkel θ gegenüber der x-Achse des Koordinatensystems beschrieben. Eine Suche in einem fünfdimensionalen Parameterraum ist sehr zeitaufwendig, so daß diese Art von Ellipsenextraktion für unsere Anwendung nicht in Betracht kommt. Es existieren aber gute Ansätze, den hochdimensionalen Unterraum in mehrere kleindimensionierte Unterräume zu unterteilen (siehe hierzu z.B. [Ho95] und [Nair95]). Das Problem bei der Unterteilung der Unterräume ist aber, daß immer die (Symmetrie-)Eigenschaft der *gesamten* Ellipse ausgenutzt wird, so daß sich im Bild Konturpunkte der ganzen Ellipse befinden müssen. Bei unserer Anwendung hingegen ist der markante Punkt für die spätere Regelung, den es implizit über die extrahierte Ellipse zu finden gilt, auf der unteren Kontur des einzufügenden Zylinders, die nur eine Halbellipse repräsentiert. Daher sind auch Algorithmen dieser Art nicht von uns nutzbar. Also müssen wir auf eine andere (klassische) Methode zurückgreifen. In diesem Abschnitt wird nun beschrieben, wie untere Halbellipsen aus einem Bild extrahiert werden können.

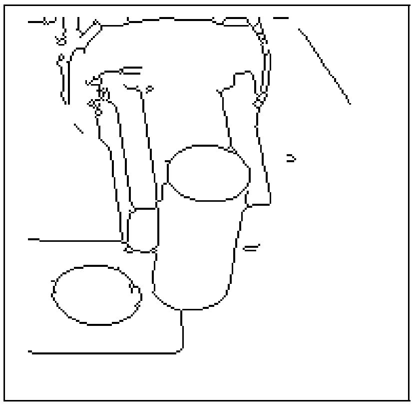
Im folgenden wird zunächst der Ablauf grob skizziert, um auf die Probleme aufmerksam zu machen, die bei diesem Verfahren auftreten können und behoben werden müssen.

Das Verfahren beruht auf einer *Segmentierung* des Kantenbildes in untere Ellipsenbögen und andere Kontursegmente, die keine Ellipsenbögen sind. An jedem segmentierten Konturstück wird mit Hilfe der Konturpixel eine Halbellipsenapproximation gemacht. Die errechnete untere Halbellipse wird auf das Kantenbild zurückprojiziert und pixelweise mit diesem Konturstück verglichen. Nur wenn eine bestimmte Anzahl von errechneten Pixeln mit den tatsächlichen Konturpixeln übereinstimmen, gilt eine untere Halbellipse als gefunden. Die Schwierigkeit bei diesem Verfahren liegt in der Segmentierung der durchgehenden Konturabschnitte in Abschnitte, die nur zu einer unteren Halbellipse gehören. Man weiß nämlich nicht, wann eine untere Halbellipse aufhört und ein gerades Berandungssegment anfängt. In Abbildung 6.15 wäre dies z.B. der Übergang der durchgezogenen Halbellipse zu den gestrichelt gezeichneten seitlichen Berandungssegmenten des Zylinders. Ein weiteres Problem tritt bei inhomogenem Hintergrund auf, wenn z.B. der Zylinder halb über dem Lochbrett ist. In Abbildung 6.16 sieht man z.B. ein Berandungssegment des Lochbrettes, das eine Verbindung mit der gesuchten Halbellipse besitzt. Im folgenden wird beschrieben, wie wir diese Segmentierungsprobleme lösen[4].

Aus dem Bild wird zuerst ein ausgedünntes Kantenbild, wie im vorigen Abschnitt beschrieben, erstellt, so daß jede Kontur nur noch ein Pixel Breite hat. Lücken mit einer Größe kleiner δ_a Pixeln (z.B. 2 Pixel) werden zwischen jedem Konturstückchen geschlossen. Jedes Pixel einer Kontur hat jetzt nur noch höchstens zwei Nachbarn, es sei denn, es treffen zwei Konturen aufeinander. Dieses Pixel wird im folgenden als Verzweigungspixel ("junction") bezeichnet. Die Abbildung 6.16 zeigt das Originalbild und das ausgedünnte Kantenbild des Beispielbildes.



(a) Originalbild.

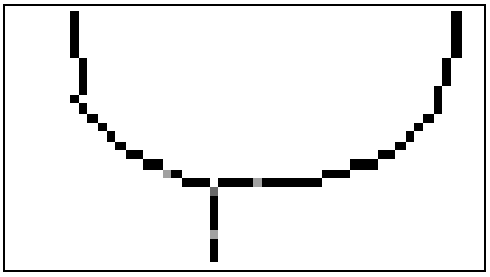


(b) Gradientenbetragsbild.

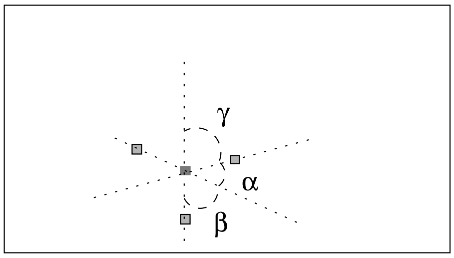
Abbildung 6.16: Beispielbild mit zugehörigem Gradientenbetragsbild.

[4] Hier zeigt sich deutlich der Vorteil einer Hough-Transformation gegenüber diesem klassischen Verfahren. Bei der Hough-Transformation ist eine solche Segmentierung nicht nötig und es können die Segmente sogar trotz Lücken in der Kontur gefunden werden.

An einer solchen Verzweigung wird jedes abgehende Konturstück ausgehend vom Verzweigungspixel bis hin zu fünf Pixeln Entfernung oder - falls dies vorher eintritt - bis zum Ende der Kontur als ein Geradensegment approximiert. Jetzt wird der Krümmungswinkel der Geradensegmente untereinander gemessen. Es entsteht ein kleiner Krümmungswinkel, nämlich derjenige, der zwischen den Geradensegmenten der unteren Halbellipsenkontur gemessen wird. Außerdem entstehen zwei hohe Krümmungswinkel, die jeweils den Winkel zwischen einem der Halbellipse gehörendem Geradensegment und einem auf die Halbellipse zulaufenden Geradensegment repräsentieren. Dasjenige Segment, das jeweils einen hohen Krümmungswinkel mit dem anderen Segment aufweist, wird segmentiert, indem das Pixel, das das Segment und das Verzweigungspixel verbindet, gelöscht wird. Im Beispielbild (6.17), das einen vergrößerten Ausschnitt vom unteren Teil des Zylinders im Kantenbild zeigt, ist es das Geradensegment, das mit den anderen Segmenten den Winkel β und γ bildet.



(a) vergrößerter Ausschnitt.



(b) Winkelmessung an den approximierten Geraden.

Abbildung 6.17: Ein Beispielausschnitt mit Winkelmessung der approximierten Geradensegmente. Links: Ausschnitt mit grau markiertem Verzweigungspixel und davon ausgehend die Endpixel für die Geradenapproximation in 5 Pixel Entfernung. Rechts: Eingezeichnete Winkel, die die Geraden miteinander bilden.

Um die unteren Halbellipsen von den seitlichen Berandungssegmenten zu trennen, wird jede Kante aus dem Bild entfernt, die mit dem im vorigen Abschnitt beschriebenen Verfahren der Kantenextraktion ermittelt wurde. Die Abbildung 6.19(a) zeigt die extrahierten Geradensegmente des Beispielbildes, die im Kantenbild gelöscht werden. Um eine untere Halbellipse von einer oberen Halbellipse zu trennen, und andere "Konturreste", die nicht zu der unteren Halbellipse gehören, zu entfernen, wird zuletzt noch jede neu entstandene Teilkontur alle fünf Pixel durch ein Geradensegment approximiert und das Steigungsverhalten der Geradensegmente zueinander getestet. Entspricht das Steigungsverhalten der Geradensegmente nicht einer unteren Halbellipse, werden die Geradensegmente voneinander getrennt, indem das verbindende Pixel gelöscht wird. Sei hierzu seg_1 das linke approximierte Segment einer Teilkontur mit der Steigung m_{seg_1}, und seg_2 das rechte mit der Steigung m_{seg_2}. Sei weiterhin der von seg_1 und seg_2 eingeschlossene Winkel bezeichnet mit $\angle(seg_1 seg_2)$. Der Verbindungspunkt VP wird dann nach folgendem Schema gelöscht:

$$VP = \begin{cases} 0, \quad \text{falls} & \begin{cases} (m_{seg_1} > m_{seg_2}) \wedge (m_{seg_1} \geq 0), \text{ oder} \\ (m_{seg_1} > m_{seg_2}) \wedge (m_{seg_1} \leq 0) \wedge (m_{seg_2} \leq 0), \text{ oder} \\ \angle(seg_1 seg_2) < 90° \end{cases} \\ 1 \quad \text{sonst.} \end{cases} \tag{6.8}$$

Die Abbildung 6.18 zeigt drei Beispiele, wo das Verbindungspixel jeweils gelöscht werden muß. Die Abbildung 6.18(a) ist ein Beispiel für den ersten Fall aus Schema (6.8), Abbildung 6.18(b) ein Beispiel für den zweiten, und Abbildung 6.18(c) ein Beispiel für den dritten Fall.

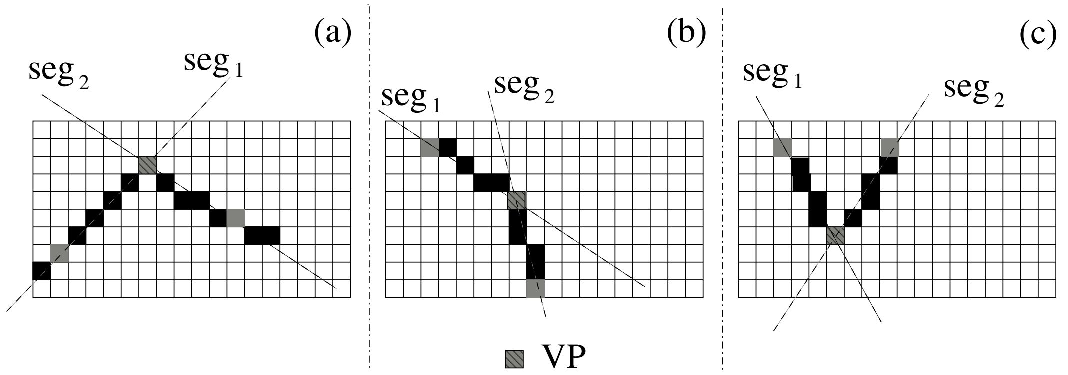
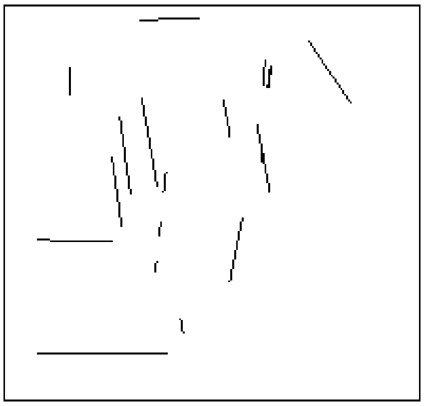
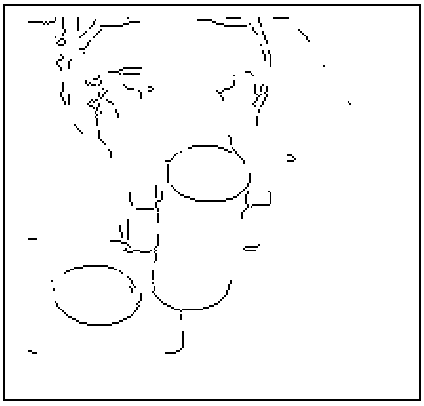


Abbildung 6.18: Drei Beispiele für das Löschen des Verbindungspixels aufgrund des Steigungsverhaltens der approximierten Geradensegmente.

Die Abbildung 6.19(b) zeigt das vollständig segmentierte Beispielbild.



(a) Gefundene Kanten zum Bild.



(b) Vollständig segmentiertes Bild.

Abbildung 6.19: Links: gefundene Kanten im Beispielbild. Rechts: Bild mit entfernten Kanten und nach Winkelberechnung.

An jedem übriggebliebenden größeren Konturstück wird nun eine Ellipsenapproximation durchgeführt. Um die einzelnen Konturstücke voneinander zu unterscheiden, erhält nun jede Zusammenhangskomponente im Sinne der 8-er Nachbarschaft einen anderen Grauwert bzw. ein sie auszeichnendes Label. Über ein Histogramm werden die größeren Zusammenhangskomponenten herausgefunden, bei jeder einzelnen größeren Zusammenhangskomponente wird eine Ellipsenapproximation durchgeführt:

Für die Ellipsenapproximation wird die Methode der *direkten kleinsten Quadrate* verwendet. Hierfür wurden die Ergebnisse des Artikels von [Fitzgibbon96] verwendet. Die Methode hat den Vorteil, daß sie sowohl effizient, als auch ellipsenspezifisch ist und somit als Ergebnis wirklich nur eine Ellipse liefert und nicht, wie bei anderen Methoden, eine Hyperbel oder Parabel. Weiterhin ist die Methode robust gegenüber Okklusion, was ebenso wichtig ist, denn der Dateninput für die Approximation ist nur eine Halbellipse.

Um die Methode zu beschreiben, wird erst einmal das Problem allgemein eingeführt:

Ein allgemeiner Kegelschnitt läßt sich durch ein implizites Polynom zweiten Grades folgendermaßen beschreiben:

$$F(\boldsymbol{a}, \boldsymbol{x}) = \boldsymbol{a} \cdot \boldsymbol{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0. \tag{6.9}$$

Hierbei ist $\boldsymbol{a} = [a\ b\ c\ d\ e\ f]^T$ und $\boldsymbol{x} = [x^2\ xy\ y^2\ x\ y\ 1]^T$. Die Funktion $F(\boldsymbol{a}, \boldsymbol{x_i}) = \mathrm{d}'$ wird der algebraische Abstand eines Punktes $\boldsymbol{x_i} = (x_i, y_i)$ zu dem Kegelschnitt $F(\boldsymbol{a}; \boldsymbol{x}) = 0$ genannt. Eine Ellipsenapproximation der Kurve zu N gegebenen Bildpunkten $\boldsymbol{x_i}$ kann erreicht werden, indem man die Summe der quadrierten algebraischen Abstände der Kurve zu den N Punkten minimiert:

$$\hat{\boldsymbol{a}} \;=\; \arg \min_{\boldsymbol{a}} \left\{ \sum_{i=1}^{N} F(\boldsymbol{a}, \boldsymbol{x_i})^2 \right\}. \tag{6.10}$$

Um die triviale Lösung $\boldsymbol{a} = [0, 0, 0, 0, 0, 0]$ zu vermeiden, muß der Parametervektor $\boldsymbol{a}$ noch eine zusätzliche Nebenbedingung erfüllen. Hierzu gibt es viele verschiedene Ansätze. Einige Beispiele seien hier genannt:

* $\|\boldsymbol{a}\|^2 = 1$. [Fitzgibbon95]
* $f = 1$. [Rosin93] [Zhang95]
* $a + c = 1$. [Zhang95] [Rosin93] [Gander94]
* $a^2 + \frac{1}{2}b^2 + c^2 = 1$. [Bookstein79]

All diese Einschränkungen sind entweder *linear*, mit dem 1x6 *Einschränkungsvektor* $\boldsymbol{c}$ ergibt sich also die Form $\boldsymbol{c} \cdot \boldsymbol{a} = 1$, oder sie sind *quadratisch*, wobei sich mit der 6x6 *Einschränkungsmatrix* $\boldsymbol{C}$ die Form $\boldsymbol{a}^T \boldsymbol{C} \boldsymbol{a} = 1$ ergibt.

In unserem Fall muß der Parametervektor $\boldsymbol{a}$ so eingeschränkt werden, daß der Kegelschnitt, den er repräsentiert, wirklich nur eine Ellipse ergibt (siehe Abb. 6.20).

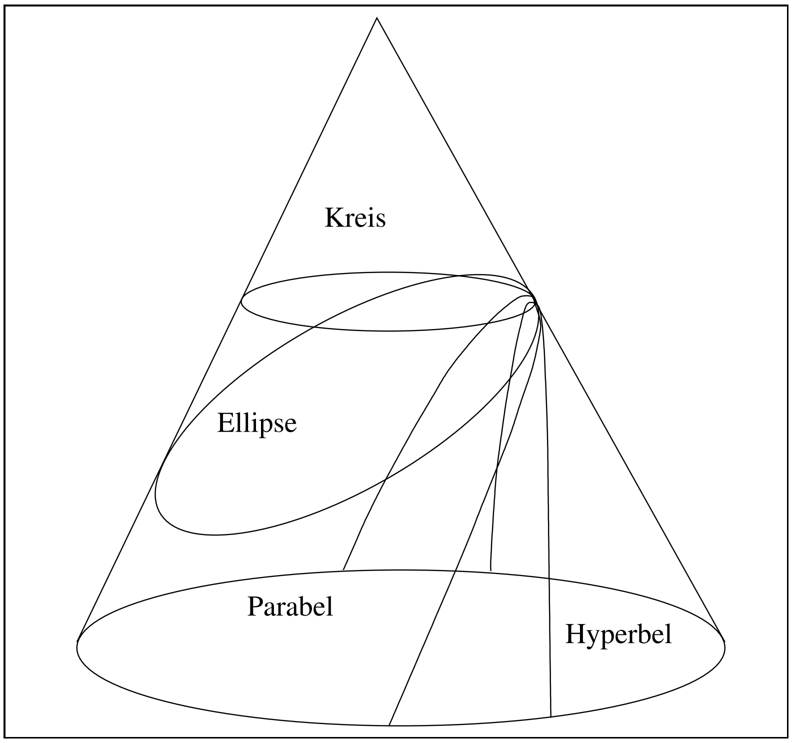


Abbildung 6.20: Mögliche Kegelschnitte.

Die gewünschte Einschränkung lautet allgemein, daß die *Diskriminante* $b^2 - 4ac$ negativ sein muß. Diese *Ungleichung* ist im allgemeinen schwierig zu verwirklichen. In [Fitzgibbon96] wurde jedoch gezeigt, daß es möglich ist, die Parameter so zu skalieren, daß man mit Hilfe der Skalierung eine einfache *Gleichung* als Nebenbedingung einführen kann, nämlich $4ac - b^2 = 1$, die genau zu einer Lösung führt. Dies ist eine quadratische Einschränkung, die in der Matrixschreibweise $\boldsymbol{a}^T \boldsymbol{C} \boldsymbol{a} = 1$ folgendermaßen lautet:

$$\boldsymbol{a}^T \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{a} = 1. \tag{6.11}$$

Die vollständige Beschreibung des zu lösenden Problems lautet:

$$\text{Minimiere } E = \|\boldsymbol{M}\boldsymbol{a}\|^2 \text{ unter Beachtung der Nebenbedingung } \boldsymbol{a}^T \boldsymbol{C} \boldsymbol{a} = 1. \tag{6.12}$$

$\boldsymbol{M}$ ist hierbei die nx6 *Mustermatrix*

$$\boldsymbol{M} = \begin{bmatrix} {x_1}^2 & x_1 y_1 & {y_1}^2 & x_1 & y_1 & 1 \\ & \vdots & & & \vdots & \\ {x_n}^2 & x_n y_n & {y_n}^2 & x_n & y_n & 1 \end{bmatrix}.$$

Unter Einführung des Lagrangemultiplikators λ und nach Differentiation erhält man folgendes Gleichungssystem

$$\begin{aligned} 2\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{a} - 2\lambda\boldsymbol{C}\boldsymbol{a} &= 0, \\ \boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} &= 1. \end{aligned} \tag{6.13}$$

Dieses System läßt sich umformen zu

$$\boldsymbol{S}\boldsymbol{a} = \lambda\boldsymbol{C}\boldsymbol{a}, \tag{6.14}$$

$$\boldsymbol{a}^T\boldsymbol{C}\boldsymbol{a} = 1. \tag{6.15}$$

Hierbei ist S die *Streuungsmatrix* $\boldsymbol{M}^T\boldsymbol{M}$. Die Gleichung (6.14) wird auch als *generalisiertes Eigenwertproblem*[5] bezeichnet. Das gesamte System ist gelöst, wenn man die generalisierten Eigenvektoren von (6.14) betrachtet. Falls das Eigenwert-Eigenvektorpaar $(\lambda_i, \boldsymbol{u_i})$ Gleichung (6.14) löst, dann auch $(\lambda_i, \mu\boldsymbol{u_i})$ für jedes μ. Mit Gleichung (6.15) läßt sich der Wert von μ_i bestimmen. Setzt man das Eigenwert-Eigenvektorpaar $(\lambda_i, \mu\boldsymbol{u_i})$ in Gleichung (6.15) ein, erhält man $\mu_i^2\boldsymbol{u}_i^T\boldsymbol{C}\boldsymbol{u}_i = 1$. Durch Umformung und mit Betrachtung von (6.14) ergibt sich:

$$\mu_i = \sqrt{\frac{1}{\boldsymbol{u}_i^T\boldsymbol{C}\boldsymbol{u}_i}} = \sqrt{\frac{\lambda_i}{\boldsymbol{u}_i^T\boldsymbol{S}\boldsymbol{u}_i}}. \tag{6.16}$$

Indem man nun $\hat{\boldsymbol{a}} = \mu_i\boldsymbol{u}_i$ setzt, läßt sich Gleichung (6.13) lösen. Man erhält bis zu 6 Lösungen. Die Lösung mit dem kleinsten Rest bei $\hat{\boldsymbol{a}}_i^T\boldsymbol{S}\hat{\boldsymbol{a}}_i = \lambda_i$ wird gewählt. Zu beachten hierbei ist noch folgendes: Die Lösung von (6.14) ergibt 6 Eigenwert-Eigenvektorpaare $(\lambda_i, \boldsymbol{u}_i)$. Jedes dieser Paare läßt ein lokales Minimum entstehen, wenn in (6.16) der Term unter der Wurzel positiv ist. Verallgemeinert heißt dies, ist $\boldsymbol{S}$ positiv definit, dann auch der Nenner $\boldsymbol{u}_i^T\boldsymbol{S}\boldsymbol{u}_i$ für alle $\boldsymbol{u}_i$. Daher existiert eine Lösung der Wurzel, wenn $\lambda_i > 0$. Dies bedeutet wiederum, daß jede Lösung für (6.13) einen *positiven* generalisierten Eigenwert haben muß.

Nachdem wir die Koeffizienten für Gleichung (6.9) gefunden haben, müssen wir nun noch die Ellipsenparameter (x_0, y_0, a, b, θ) explizit ausrechnen. Hierfür wurden die Gleichungen aus dem Buch von [Korn61] verwendet. Ersetzt man in Gleichung (6.9) den Vektor $\boldsymbol{a} = [a\ b\ c\ d\ e\ f]^T$ durch $\boldsymbol{b} = [\hat{m}_{11}\ 2\hat{m}_{12}\ \hat{m}_{22}\ 2\hat{m}_{13}\ 2\hat{m}_{23}\ \hat{m}_{33}]^T$, läßt sich Gleichung (6.9) umschreiben zu:

$$F(\boldsymbol{b}, \boldsymbol{x}) = (\hat{m}_{11}x + \hat{m}_{12}y + \hat{m}_{13})x + (\hat{m}_{21}x + \hat{m}_{22}y + \hat{m}_{23})y + (\hat{m}_{31}x + \hat{m}_{32}y + \hat{m}_{33}) = 0, \tag{6.17}$$

[5] Die Lösung eines solchen Problems wird in Anhang A gezeigt.

mit $\hat{m}_{ik} = \hat{m}_{ki} \quad (i, k = 1, \ldots, 3)$.

Hierzu lassen sich folgende Invarianten bestimmen bezüglich Rotation und Translation des Kegelschnittes angeben:

$$\hat{I} = \hat{m}_{11} + \hat{m}_{22}, \quad \hat{D} = \det \begin{pmatrix} \hat{m}_{11} & \hat{m}_{12} \\ \hat{m}_{21} & \hat{m}_{22} \end{pmatrix}, \quad \hat{M} = \det \begin{pmatrix} \hat{m}_{11} & \hat{m}_{12} & \hat{m}_{13} \\ \hat{m}_{21} & \hat{m}_{22} & \hat{m}_{23} \\ \hat{m}_{31} & \hat{m}_{32} & \hat{m}_{33} \end{pmatrix}. \tag{6.18}$$

Betrachtet man weiterhin die charakteristische quadratische Form zu (6.17)

$$F_0(x, y) = \hat{m}_{11}x^2 + 2\hat{m}_{12}xy + \hat{m}_{22}y^2 \tag{6.19}$$

und die Eigenwerte $\hat{\lambda}_1, \hat{\lambda}_2$ zur charakteristischen Gleichung von (6.19)

$$\begin{aligned} \det \begin{pmatrix} \hat{m}_{11} - \hat{\lambda} & \hat{m}_{12} \\ \hat{m}_{21} & \hat{m}_{22} - \hat{\lambda} \end{pmatrix} &= 0, \; bzw. \\ \hat{\lambda}^2 - \hat{I}\hat{\lambda} + \hat{D} &= 0, \end{aligned} \tag{6.20}$$

lassen sich die expliziten Ellipsenparameter wie folgt ermitteln:

Für den Mittelpunkt der Ellipse (x_0, y_0) gilt:

$$\hat{m}_{11}x_0 + \hat{m}_{12}y_0 + \hat{m}_{13} = 0 \quad \text{und} \quad \hat{m}_{21}x_0 + \hat{m}_{22}y_0 + \hat{m}_{23} = 0. \tag{6.21}$$

Also folgt:

$$x_0 = -\frac{1}{\hat{D}} \cdot \det \begin{pmatrix} \hat{m}_{13} & \hat{m}_{12} \\ \hat{m}_{23} & \hat{m}_{22} \end{pmatrix} \tag{6.22}$$

und

$$y_0 = -\frac{1}{\hat{D}} \cdot \det \begin{pmatrix} \hat{m}_{11} & \hat{m}_{13} \\ \hat{m}_{21} & \hat{m}_{23} \end{pmatrix}. \tag{6.23}$$

Für diese Gleichungen muß $\hat{D} \neq 0$ gelten. Dies ist aber bei einer Ellipse immer der Fall, und daß es sich um eine Ellipse handelt, haben wir schon bei der Berechnung von Vektor $\boldsymbol{a}$ sichergestellt.

Es gilt weiterhin, daß die Hauptachsen der Ellipse entlang den Eigenvektoren der Matrix $(\hat{m}_{ik})$ laufen. Der Richtungkosinus $cos(\theta')$ bzw. $sin(\theta')$ der Normalen zu einer Hauptachse genügt den Gleichungen:

$$\begin{aligned} (\hat{m}_{11} - \hat{\lambda})\cos(\theta') + a_{12} \cdot \sin(\theta') &= 0 \\ \hat{m}_{21} \cdot \cos(\theta') + (a_{22} - \hat{\lambda}) \cdot \sin(\theta') &= 0. \end{aligned} \tag{6.24}$$

Mit Hilfe dieser Gleichungen läßt sich θ' nach Einsetzen von $\hat{\lambda}_2$ bestimmen:

$$\tan(\theta') = \frac{\hat{m}_{11} - \hat{\lambda}_2}{-\hat{m}_{12}} = \frac{\hat{m}_{21}}{\hat{\lambda}_2 - \hat{m}_{22}}. \tag{6.25}$$

Den Rotationsparameter θ der Ellipse erhält man dann durch folgende Verifikation:

$$\theta = \begin{cases} \theta' - \pi & \text{falls,} \quad \theta' > \frac{\pi}{2} \\ \theta' + \pi & \text{falls,} \quad \theta' < -\frac{\pi}{2} \\ \theta' & \text{sonst.} \end{cases} \tag{6.26}$$

Die Radien a, b der Ellipse erhält man schließlich durch ihre Verbindung zu den Invarianten $\hat{M}, \hat{D}, \hat{I}$ und den Eigenwerten $\hat{\lambda}_1$,$\hat{\lambda}_2$ der charakteristischen Gleichung:

$$a = \sqrt{-\frac{1}{\hat{\lambda}_2}\frac{\hat{M}}{\hat{D}}}, \tag{6.27}$$

$$b = \sqrt{-\frac{1}{\hat{\lambda}_1}\frac{\hat{M}}{\hat{D}}}. \tag{6.28}$$

Mit Hilfe der gefundenen Ellipsenparameter wird die untere Halbellipse ins Bild projiziert und jedes Pixel der errechneten Ellipse wird mit dem Kantenbild verglichen. Nur wenn ein angemessener Prozentsatz der projizierten Pixel mit den Pixeln des Kantenbildes übereinstimmen, gilt die untere Halbellipse als gefunden. Die Abbildung 6.21 zeigt einen Vergleich anhand eines vergrößerten Teilbereiches der errechneteten Ellipse und des Kantenbildes.

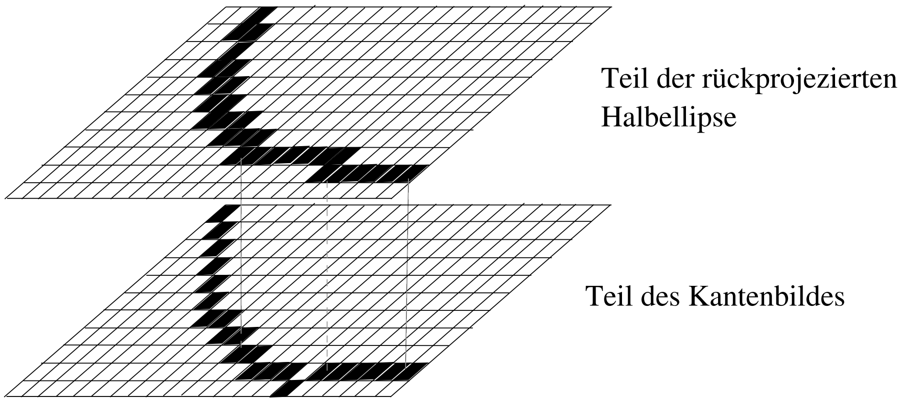


Abbildung 6.21: Ein pixelweiser Vergleich der rückprojizierten Halbellipse mit den Pixeln des Kantenbildes. Dargestellt ist hier ein vergrößerter Teilbereich. Die gestrichelten Linien stellen drei Beispielvergleiche dar.

Die Abbildung 6.22 zeigt das gesamte Beispielbild mit eingezeichneten gefundenen Halbellipsen.

Abbildung 6.22: Beispielbild mit gefundenen projizierten Halbellipsen (schwarz mit grauer Umrandung).

Die Vorgehensweise für die Ellipsenextraktion läßt sich zusammenfassend im folgenden Schema skizzieren.

Schema der Ellipsenextraktion

- Ableitung des Bildes in x- und y- Richtung
- Ermittlung des Gradientenbetrags- und des Gradientenorientierungsbildes
- Ausdünnen des Gradientenbetragsbildes mittels des kodierten Gradientenwinkelbildes
- Erstellung eines Kantenbildes mittels Binarisierung durch eine Schwellwertoperation
- Schließen von Lücken zwischen Kontursegmenten mit Abstand kleiner δ_a
- Segmentierung von verbundenen Konturen mit Hilfe einer Winkelmessung der durch jeweils einem Geradensegment approximierten Kontursegmente an den Verzweigungspixeln
- Elimination von Kanten über Subtraktion des Ergebnisses einer orientierungsselektiven Hough-Transformation
- Segmentierung der einzelnen Kontursegmente durch Approximation in Kantensegmente und Winkelmessung
- Zuweisung eines eindeutigen Grauwertes ("Label") für jedes Kontursegment
- Ellipsenapproximation an jedem gelabelten Kontursegment ab einer bestimmten Größe
- Verifikation durch Rücktransformation der errechneten Ellipse auf das Kantenbild und Vergleich der jeweiligen übereinanderliegenden Pixel

6.2.2 Ellipsenerkennung

Wie auch bei der Kantenextraktion schon beschrieben, nützt es nichts, nur einfach alle Ellipsen aus einem Bild zu extrahieren. Man muß nun auch noch zwischen den gefundenen Ellipsen die richtige, in unserem Fall die untere Halbellipse des Zylinders, verifizieren. Dafür lassen sich wieder Quasi-Invarianten des Zylinders beschreiben, die unsere Suche vereinfachen sollen. Wie auch beim Quader verändern sich die Grauwerte der Flächen des Zylinders von Aufnahme zu Aufnahme nicht stark, unter der Annahme, daß der Zylinder homogene Seitenflächen hat, und sich die Beleuchtungsverhältnisse nur wenig von Aufnahme zu Aufnahme verändern. Das Gleiche gilt für die Größe des Zylinders und damit verbunden auch für die Größe seiner beiden sichtbaren Halbellipsen, bei einer nur relativ geringen Veränderung des Zylinders in Richtung Kamera, gegenüber einer größeren Entfernung der Kamera zum Objekt. Da der Zylinder senkrecht in der Hand liegt, sind die zu sehenden Halbellipsen des Zylinders - wenn überhaupt - in der Projektion nur wenig gedreht zu sehen. Darum läßt sich für eine Halbellipse he_i bestehend aus N Punkten (p_j^x, p_j^y) folgender oberer benachbarter Durchschnittsgrauwert GW^o ermitteln:

$$GW^o(he_i) = \frac{1}{N}\sum_{j=1}^{N} gw(p_j^x, p_j^y - 1). \tag{6.29}$$

Macht man nun initial einmal eine Aufnahme des Zylinders alleine mit homogenem Hintergrund, lassen sich durch diese visuelle Demonstration über den oberen benachbarten Durchschnittsgrauwert der Ellipsen der Grauwert der Deckfläche des Zylinders gw^{DF} und der Grauwert der vorderen Fläche gw^{VF} beschreiben. Desweiteren läßt sich die zu erwartende Größe der Hauptachsen der Ellipsen bestimmen, die wir mit ga und gb bezeichnen wollen.

Da die Halbellipse, die die obere Deckfläche nach unten begrenzt, immer sichtbar ist, wird sie als erstes extrahiert. Die Methode, die gesuchte Halbellipse he^{DF} aus einer Menge von R gefundenen Halbellipsen zu verifizieren, lautet folgendermaßen. Hierbei sei $a^{(he_i)}$ der zu he_i zugehörige Hauptradius, entsprechendes gilt für den anderen Radius $b^{(he_i)}$.

$$he^{DF} := \arg\{he_i | (GW^o(he_i) \approx gw^{DF}) \wedge (a^{(he_i)} \approx ga) \wedge (b^{(he_i)} \approx gb); i = 1, ...R\}. \tag{6.30}$$

Das Zeichen "$\approx$" soll hier wieder die Gleichheit bis auf eine vom Benutzer jeweilig anzugebende Fehlerschranke δ_i anzeigen.

Hat man diese obere Ellipse erkannt, läßt sich die eigentlich gesuchte Halbellipse he^{VF} durch folgende Gleichung verifizieren. Wir unterscheiden hierbei wieder, wie auch schon vorher bei der Ermittlung des gesuchten Segmentes beim Quader, zwischen der Ermittlung der Halbellipse bei der ersten Aufnahme und der Ermittlung bei allen anderen Aufnahmen. Das Argument $min_{|x_0^{(he^{DF})} - x_0^{(he_j)}|}$ bedeutet hierbei jeweils, daß man diejenige Halbellipse sucht, deren Mittelpunkt (x_0, y_0) den kleinsten horizontalen Abstand zur Halbellipse he^{DF} hat. Dies begründet sich in der Tatsache, daß der Zylinder fast senkrecht im Bild zu sehen ist, und die Ellipsen somit fast genau untereinander liegen. Bei der ersten Aufnahme wird die Ellipse he^{VF} folgendermaßen erkannt:

$$\begin{aligned} he^{VF} := \arg\{min_{|x_0^{(he^{DF})} - x_0^{(he_j)}|}\{he_j|(he^{DF} \neq he^{VF}) \quad &\wedge \quad (GW^o(he_j) \approx gw^{vF}) \wedge (a^{(he_j)} \approx ga) \\ &\wedge \quad (b^{(he_j)} \approx gb); j = 1, ...R\}\}. \qquad (6.31) \end{aligned}$$

Nachdem man die gewünschte Ellipse verifiziert hat, rechnet man, genau wie bei der Kantenverifikation, den horizontalen und vertikalen Abstand $\boldsymbol{abst} = (abst^x, abst^y)$ zwischen den Mittelpunkten von he^{DF} und he^{VF} aus, um bei den nachfolgenden Aufnahmen wieder eine Suchbereichseinschränkung zu haben (siehe Abb. 6.23).

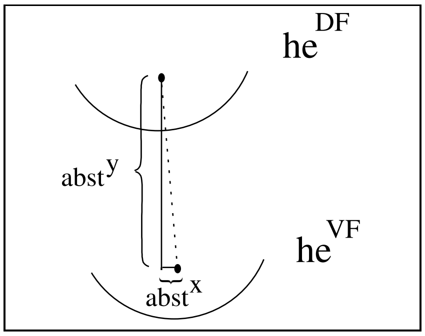


Abbildung 6.23: Ermittlung des Abststandes $\boldsymbol{abst} = (abst^x, abst^y)$ zwischen zwischen den Mittelpunkten von he^{DF} und he^{VF}.

Die Gleichung (6.31) wird dann also für jede weitere Aufnahme noch um folgende Bedingungen erweitert:

$$\begin{aligned} he^{VF} := \arg\{min_{|x_0^{(he^{DF})} - x_0^{(he_j)}|}\{he_j|... \quad &\wedge \quad (x_0^{(he^{DF})} + abst^x \approx x_0^{(he_j)}) \\ &\wedge \quad (y_0^{(he^{DF})} + abst^y \approx y_0^{(he_j)}); \\ & \quad\; j = 1, ...R\}\}. \qquad (6.32) \end{aligned}$$

Sollten diese Bedingungen nicht erfüllt sein, läßt sich - wie beim Quader - die Symmetrieeigenschaft des Zylinders nutzen, und man nimmt die um $abst^x$ und $abst^y$ nach unten projizierte Halbellipse he^{DF} als he^{VF}. Dies gewährleistet auch dann eine Verifikation, wenn die untere Ellipse nicht erkannt wurde (z.B. bei Verdeckung derselben).

Bei jeder Aufnahme, in der die untere Ellipse verifiziert wurde, wird wieder ein neuer Abstand $(abst^x, abst^y)$ errechnet, um für die nächste Aufnahme den Suchbereich für he^{VF} einzuschränken.

Um nun einen markanten Punkt als Istwert für die Regelung zu extrahieren, wird - wie oben schon erwähnt - derjenige Punkt der Halbellipse genommen, der dieselbe Tangentensteigung hat, wie das vordere Berandungssegment des Lochbrettes (siehe Abb. 6.15)[6]. Somit

[6]Wie die Steigung der Kante des Lochbrettes ermittelt wird, wird im nächsten Abschnitt beschrieben.

hat man in beiden Bildern einen eindeutigen auf die Bildebenen projizierten 3D Punkt bestimmt. Für die Tangentenpunktbestimmung gelten die nachstehenden Gleichungen. Eine Tangente im Punkt $\boldsymbol{E_p} = (x_p, y_p)$ auf der Ellipse mit Mittelpunkt (x_0, y_0), den Hauptradius a und den Nebenradius b hat die Gleichung:

$$\frac{x \cdot (x_p - x_0)}{a^2} + \frac{y \cdot (y_p - y_0)}{b^2} = 1. \tag{6.33}$$

Formt man die Gleichung um, läßt sich die Steigung m_t der Tangente in diesem Punkt angeben mit

$$m_t = \frac{b^2 \cdot (x_p - x_0)}{a^2 \cdot (y_p - y_0)}. \tag{6.34}$$

Derjenige Punkt der errechneten unteren Halbellipse, der die ähnlichste Tangentensteigung wie die vordere Berandung des Lochbrettes hat, ist der gesuchte markante Punkt im Bild. Die Abbildung 6.24 zeigt die verifizierte Halbellipse mit unserem gesuchten markanten Punkt als Istwert für die Regelung.

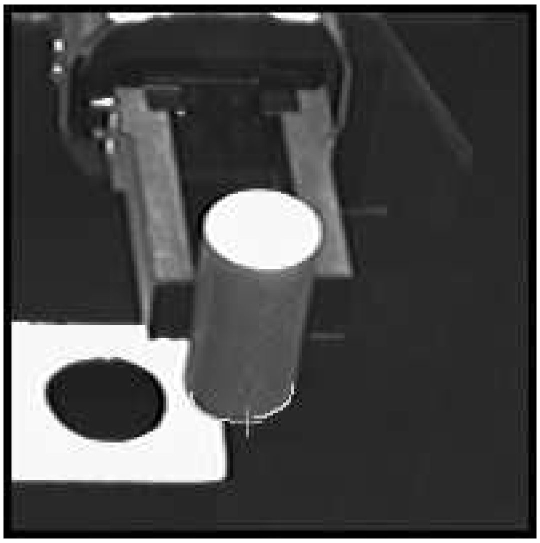


Abbildung 6.24: Beispielbild mit verifizierter Halbellipse und markantem Punkt.

Nachdem die ersten beiden Abschnitte die Istwertbestimmung als markanten Punkt des jeweils anzunähernden Objektes beschrieben haben, zeigt nun der nächste Abschnitt die Bestimmung des Sollwertes für die Annäherung.

6.3 Sollwertbestimmung beim Loch

Dieser Abschnitt beschreibt die Bestimmung unseres Sollwertes für die Regelung, nämlich die Beschreibung eines am Loch befindlichen markanten Punktes, zu dem das Objekt mit

Hilfe des markanten Punktes am Objekt hingeregelt werden soll. Um den markanten Punkt am Loch zu bestimmen, wird zuerst einmal das Lochbrett selber extrahiert. Da die Hauptaufgabe der Arbeit in der Regelung und Erkennung der zu regelnden Objekte liegen und nicht in der Erkennung des Lochbrettes, gelten für das Lochbrett folgende beispielhafte Annahmen:

Das Lochbrett hat die geometrische Form eines Rechtecks. Es befindet sich im vorderen Wirkungsbereich des Roboters, damit es vom Roboter erreicht werden kann. Die Lochbrettfläche ist homogen und hell gegenüber den Löchern, was aber bei einer Beleuchtung von oben eine natürliche Situation ist.

Da das Lochbrett die geometrische Form eines Rechtecks besitzt und somit aus klar definierten Geraden besteht, geschieht die Extraktion des Brettes wieder mit der in Abschnitt 6.1.1 beschriebenen orientierungsselektiven Hough-Transformation. Weil man nicht vorher weiß, wo sich das Lochbrett im Bild befindet, müßte man ein Hough-Transformation über das gesamte 768x576 Pixel große Bild machen. Durch die Annahme, daß sich das Lochbrett im vorderen Wirkungbereich des Roboterarmes befinden muß, läßt sich die Suche auf die untere Hälfte der beiden Bilder beschränken. Damit Berandungssegmente der Roboterhand die Suche nach den Berandungssegmenten des Lochbrettes nicht stören, wird der Arm für die Aufnahme kurz aus dem Bild in seine Ausgangsposition gefahren, bei der er senkrecht nach oben zeigt. Für die Maximasuche im Houghraum wird wieder, wie bei den Berandungssegmenten des Quaders, die Quasi-Invarianz paralleler Berandungssegmente gegenüber der Projektion genutzt. Zwei im Weltkoordinatensystem parallele Segmente sind bei großer Entfernung der Kamera gegenüber kleiner Größe des Objektes auch (nahezu) parallel nach der Projektion auf die Bildebene. Somit befinden sich auch die zwei Maxima im Houghbild auf demselben horizontalen Streifen mit einer Breite kleiner δ_b, denn der vertikale Wert im Houghbild bestimmt die Orientierung der Geraden. Der Parameter δ_b ist eine vom Benutzer anzugebende Schranke. Ein guter Wert ist eine Breite von 5 Pixeln, was eine Ungenauigkeit der quasiparallelen Berandungssegmente von 5 Grad in der Orientierung entspricht. Man sucht also nach dem absoluten Maximum im Houghbild und nach einem zweiten auf demselben Streifen. Nach löschen aller Maxima dieses Streifens sucht man wieder nach dem jetzt größten Maximum des Bildes und nach dem zweiten auf demselben Streifen. Nachdem man über die Maxima die Parameter für die Geraden erhalten hat, lassen sich über die Schnittpunkte der Houghgeraden im Bild die Ecken des Lochbrettes definieren. Die Abbildung 6.25 zeigt ein Beispielbild mit Lochbrett und rückprojizierter Houghgeraden.

Im folgenden sei die linke obere Ecke des Brettes mit $\boldsymbol{e_1} = (e_1^x, e_1^y)$, die linke untere mit $\boldsymbol{e_2}$, die rechte obere mit $\boldsymbol{e_3}$ und die rechte untere mit $\boldsymbol{e_4}$ beschrieben. Für die Steigung ν der Geraden $\overline{e_2 e_4}$, die der Orientierung der vorderen Kante des Lochbrettes entspricht, gilt:

$$\nu \;=\; \frac{e_2^y - e_4^y}{e_4^x - e_2^x}. \tag{6.35}$$

Hierbei ist zu beachten, daß die y-Koordinate des Kamerakoordinatensystems nach unten zeigt.

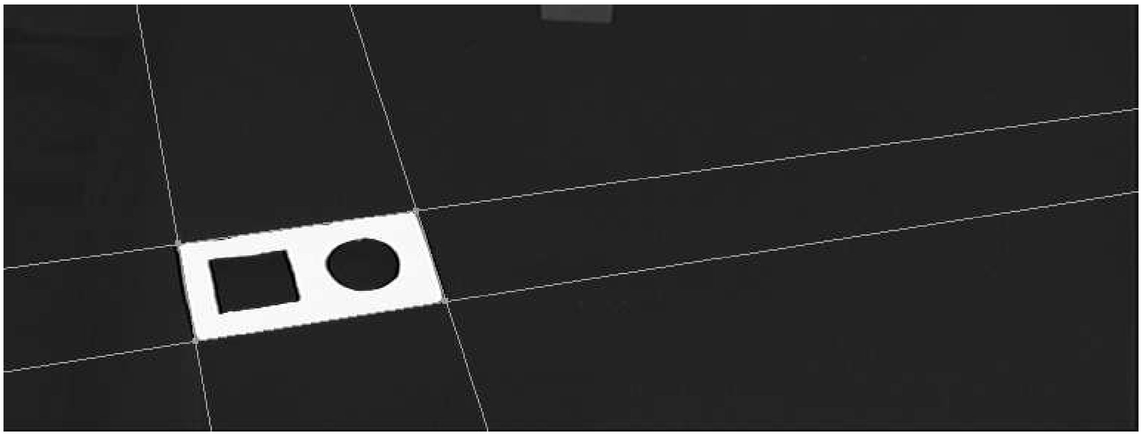


Abbildung 6.25: Beispielbild des Lochbrettes mit zugehörigen rückprojizierten Houghgeraden, deren Schnittpunkte die Eckpunkte des Lochbrettes beschreiben.

Um nun die Konturpunkte der Löcher zu extrahieren, geht ein Kontursucher auf einem durch einen Schwellwert binarisierten Bild ausgehend vom Mittelpunkt des linken Berandungssegmentes $\overline{e_1 e_2}$ des Lochbrettes auf einer Geraden mit der Orientierung ν solange in Richtung des rechten Berandungssegmentes $\overline{e_3 e_4}$ bis ein starker Grauwertunterschied von hell auf dunkel detektiert wird im Punkt $\boldsymbol{L_{s1}}$ (dies ist der erste Punkt des Loches). Entlang dieser Grauwertkante läuft der Kontursucher und merkt sich die Konturpunkte $\boldsymbol{kp_i}$ solange, bis wieder der Startpunkt $\boldsymbol{L_{s1}}$ erreicht wurde. Innerhalb der Kontur wird mit einem Füllalgorithmus die Fläche mit dem Grauwert der Lochbrettfläche ausgefüllt, damit der Kontursucher beim zweiten Durchlauf über dieses Loch weitergeht und auf die zweite Lochkontur stößt im Punkt $\boldsymbol{L_{s2}}$, diese umfährt, und sich deren Punkte ebenfalls merkt. Die Abbildung 6.26 zeigt ein Beispiel für die Extraktion der Konturpunkte des Loches für den Quader.

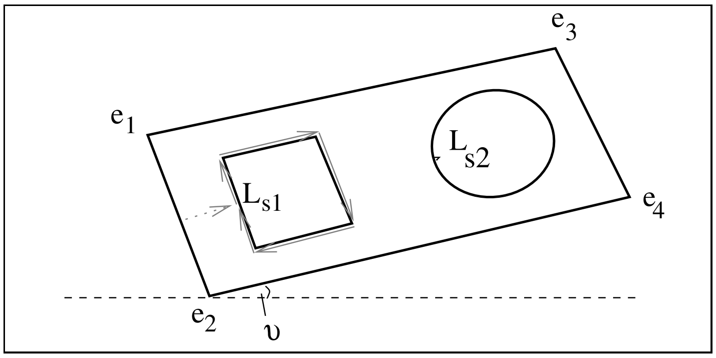


Abbildung 6.26: Extraktion der Konturpunkte des Loches für den Quader mit Hilfe des Kontursuchers.

Bemerkung: Falls man ein großes Lochbrett hat, auf dem die Löcher nicht zentriert liegen, läßt sich der Algorithmus leicht modifizieren, um trotzdem die Lochkonturen zu extrahieren. In diesen Fall startet der Kontursucher nicht vom Mittelpunkt des linken Berandungssegmentes $\overline{e_1e_2}$, sondern von dem Pixel des Segmentes $\overline{e_1e_2}$, daß einen Pixel unterhalb von e_1 liegt. Der Kontursucher startet in Richtung $\overline{e_3e_4}$ mit der Orientierung ν, bis er auf einen Grauwertunterschied von hell auf dunkel trifft. Sollte dieser Unterschied schon an einem Pixel des Berandungssegmentes $\overline{e_3e_4}$ sein, ist kein Loch gefunden, und der Sucher startet wieder von $\overline{e_1e_2}$ jedoch ein Pixel tiefer. Diese Ablauf geschieht solange, bis der Kontursucher auf die Lochkontur stößt. Diese umfährt er, merkt sich die Punkte, und das innere der Kontur wird mit dem Grauwert der Lochbrettes aufgefüllt. Dann beginnt der Kontursucher wieder vom Segment $\overline{e_1e_2}$ erneut mit dem oben beschrieben Ablauf.

Nachdem man die Konturpunkte des Loches extrahiert hat, gilt es nun zum einen das jeweilige Loch als Loch für den Quader oder für den Zylinder zu verifizieren, und zum anderen einen markanten Punkt aus diesen Lochkonturpunkten zu extrahieren. Zur Verifikation wird mit Hilfe der jeweiligen Konturpunkte eine im Abschnitt 6.2.1 schon beschriebene Ellipsenapproximation gemacht. Mit Hilfe der gefundenen Ellipsenparamter wird die Ellipse auf die Lochkonturpunkte im Bild projiziert. Dasjenige Loch, dessen Konturpunkte sich stärker mit der projizierten Ellipse decken, ist das Loch für den Zylinder. Dementsprechend ist das andere Loch das Loch für den Quader. Für die Extraktion eines markanten Punktes am Loch muß zunächst bestimmt werden, wo der markante Punkt sich am Loch befinden soll. Der markante Lochpunkt des Zylinderloches soll derjenige Punkt sein, der den kleinsten euklidischen Abstand zur vorderen Kante des Lochbrettes $\overline{e_2e_4}$ hat. Dies entspricht demjenigen Punkt, dessen Tangente dieselbe Steigung hat wie das vordere Berandungssegment des Lochbrettes. Der markante Lochpunkt des Quaderloches ist der Mittelpunkt des Segmentes, das am nächsten zum vorderen Berandungssegment des Brettes liegt. Vergleicht man die Definition des Lochpunktes mit dem markanten Punkt des Objektes, läßt sich erkennen, daß bei einer Übereinstimmung des jeweiligen Punktes, d.h. wenn der Soll-Ist-Wert gleich Null ist, das Objekt genau so über dem Loch ist, daß es nur noch senkrecht ins Loch eingeführt werden muß[7].

Um nun den markanten Punkt des jeweiligen Loches zu erhalten, wird zuerst der geometrische Schwerpunkt $\boldsymbol{schw} = (schw_x, schw_y)$ für jede Lochkontur, die im Folgenden aus N gefundenen Punkten bestehen soll, ermittelt.

$$(schw_x, schw_y) = \frac{1}{N}\sum_{k=1}^{N}(kp_{x\,k}, kp_{y\,k}). \tag{6.36}$$

Mit Hilfe dieses Schwerpunktes und der Steigung $m_{e_1e_2}$ des rechten Kantensegmentes bzw. linken Kantensegmentes $m_{e_3e_4}$ läßt sich eine Gerade G definieren, die die Kontur des Loches genau in unserem jeweiligen gesuchten markanten Punkt $\boldsymbol{kpm}$ schneidet. Es gilt für die Gerade G aufgrund der Punkt-Steigungs-Form:

$$y - schw_y = m_{e_3e_4}(x - schw_x). \tag{6.37}$$

[7]Der Ausgleich der möglichen Verdrehung des Quaders gegenüber dem Loch wird noch im folgenden beschrieben.

Jeder Konturpunkt der extrahierten Lochkontur wird in die Geradengleichung eingesetzt. Derjenige Konturpunkt, für den die Gleichung minimal wird, ist genau der Schnittpunkt der Geraden mit der Kontur und unser gesuchter markanter Punkt. Zu beachten hierbei ist, daß die Gerade die Kontur in zwei Punkten schneidet. Es wird derjenige Punkt genommen, der einen größeren y-Wert als der Schwerpunkt hat:

$$\begin{aligned}(kpm_x, kpm_y) = \arg \big\{(kp_{x\,k}, kp_{y\,k}) \quad | \quad & ((m_{e_3e_4} kp_{x\,k} - kp_{y\,k} + schw_y - m_{e_3e_4} schw_x) \to \min) \\ & \wedge\ (kp_{y\,k} > schw_y);\ k = 1, ..., N\big\}. \end{aligned} \tag{6.38}$$

Die Abbildung 6.27 zeigt die Extraktion des markanten Punktes am Loch für den Zylinder.

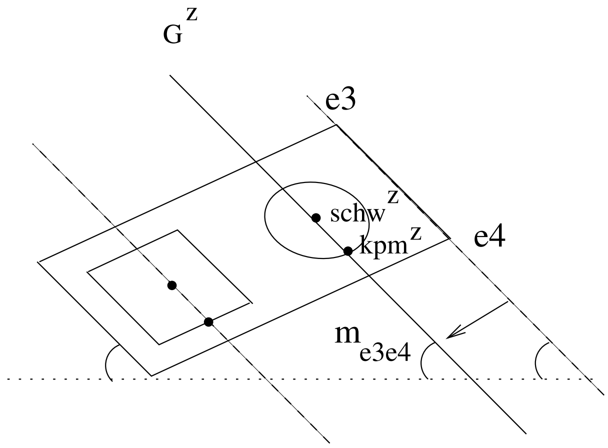


Abbildung 6.27: Extraktion des markanten Punktes kpm^z des Loches für den Zylinder durch Bestimmung des Schnittpunktes der Lochkontur mit der Geraden G^z, die über den Schwerpunkt $schw^z$ und der Steigung m_{e3e4} erzeugt wurde.

Bemerkung: Da unser Loch für das Objekt ein wenig größer ist als die Objektgrundfläche selber, lassen wir in der Praxis den markanten Punkt entlang der Gerade G in Richtung des Schwerpunktes der Lochkontur um 3 Pixel translatieren, um die Annäherung nicht direkt zur Kante des Loches sondern zentraler zu gestalten. Die Abbildung 6.28 zeigt noch einmal an einem Beispiel die extrahierten Konturpunkte der Löcher sowie den markanten Punkt des jeweiligen Loches.

Der nächste Abschnitt beschreibt den Ablauf der geregelten Anäherung und die Probleme, die dabei auftreten können, sowie deren Lösung.

6.4 Regelung bei der Annäherung

Nachdem uns nun alle benötigten Werkzeuge zur Verfügung stehen, kann die eigentliche geregelte Annäherung eines Objektes an ein Loch stattfinden. Hierfür werden zunächst die

Abbildung 6.28: Beispielbild des Lochbrettes mit eingezeichnetem Schwerpunkt und markantem Punkt des jeweiligen Loches.

Projektionsmatrizen eingelesen, die mit dem in Kapitel 3 beschriebenen Verfahren errechnet wurden. Dann wird das Loch bestimmt, das zu demjenigen Objekt paßt, das mit den in Kapitel 5 vorgestellten Verfahren bestimmt wurde. Bei dem eigentlichen Regelungszyklus ist zu beachten, daß die Suche des markanten Objektpunktes im linken und rechten Bild nicht auf dem gesamten 768x576 Pixel großen Bild vollzogen wird, sondern auf einem Ausschnitt mit einer Größe von 200x200 Pixel. Aufgrund der Vorgaben, die wir mit dem in Kapitel 5 beschriebenen Ablauf geschaffen haben, wissen wir für die Aufnahme vor dem ersten Annäherungsschritt, daß sich das Objekt in der Bildmitte beider Kameras befindet, denn das Objekt wurde ja auf den Sehstrahl der linken Kamera ausgerichtet, und die Vergenz der rechte Kamera wurde solange geregelt, bis sich die Projektion des Objektes auf die rechte Kameraebene in der Mitte des Bildes befand. Vor jedem weiteren Annäherungschritt wird eine Schätzung des neuen Suchbereiches vorgenommen, die sich auf die Rückprojektion der durch u(t) neu erreichten 3D Koordinate des Objektes auf die linke und rechte Kameraebene stützt. Diese Rückprojektion geschieht mit Hilfe der Gleichung (3.1). Die Beschränkung auf den Ausschnitt bringt einen großen Zeitgewinn mit sich, und die Objektsuche wird außerdem durch die enge Begrenzung des Suchraumes erleichtert, da Objekte, die die Suche stören könnten, ausgegrenzt werden. Die Schätzung hat aber den Nachteil, daß durch die direkte Anwendung der Gleichung (3.1) bei einer zu schlechten Kalibrierung die Lage des Suchfensters im Bild falsch prognostiziert wird, so daß das Objekt sich nicht in dem errechneten Suchbereich befindet, was genau die Philosophie des Reglers zunichte machen würde, nämlich eine gute Annäherung trotz schlechter Kalibrierung. Deswegen darf das Suchfenster nicht zu klein gewählt werden, damit sich das Objekt auch noch in dem Suchfenster befindet, wenn schlecht kalibriert wurde. Die Abbildung 6.29 zeigt als Beispiel eine Originalaufnahme mit der linken Kamera mit eingezeichnetem Suchfenster und dem Abstandsvektor, sowie sich den daraus resultierenden ersten beiden Vektorkomponenten $e_{2D_x}(t), e_{2D_y}(t)$ des Fehlervektors $\boldsymbol{e_{2D}}(t)$.

Weil die Annäherung auf direktem Wege verläuft und keine Hindernisvermeidung in dieser

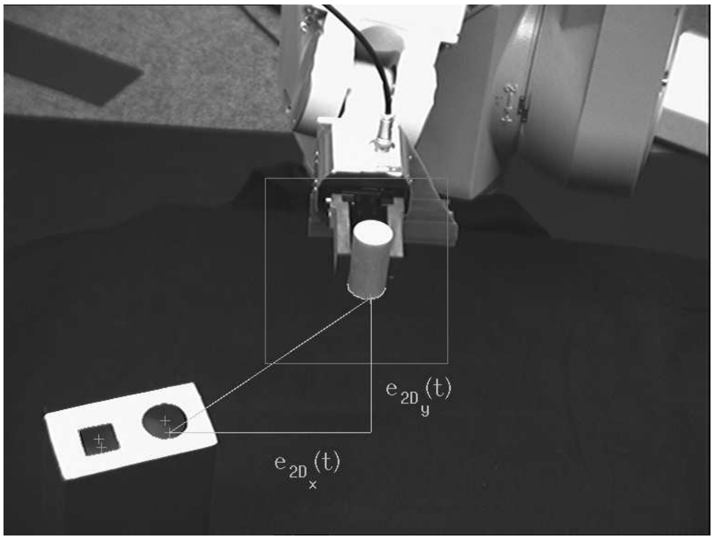


Abbildung 6.29: Aufnahme mit eingezeichnetem Suchfenster und dem Abstandsvektor als Fehler zwischen Sollwert (Lochpunkt) und Istwert (Objektpunkt).

Arbeit vorgesehen ist, kann es in bestimmten Konstellationen zu Komplikationen kommen. Falls der Startpunkt tiefer ist als das anzufahrende Loch, könnte das Objekt oder die Hand auf dem Weg zum Loch möglicherweise durch das Lochbrett durchfahren und es beschädigen. Um dieses Problem zu vermeiden, muß garantiert werden, daß das Objekt sich immer dem Lochbrett von oben nähert. Dies läßt sich erreichen, indem man nicht den markanten Punkt des Loches direkt anfährt, sondern zuerst einen virtuellen Punkt **VP**, der sich z.B. 10 cm über dem markanten Lochpunkt **LP** befindet. Hat man diesen Punkt durch Regelung erreicht, regelt man als nächstes zu dem eigentlichen Lochpunkt hin. Auf diesem Wege wird eine Kollision mit dem Lochbrett auf dem Weg zum Lochpunkt vermieden. Den virtuellen Punkt **VP** erhält man, indem der im linken und rechten Bild zur Verfügung stehende Lochpunkt **LP** mit Hilfe der Gleichung (3.6) in einen 3D Weltpunkt umgerechnet wird. Auf diesem wird dann in der Z-Komponente der Offset von z.B. 100 mm addiert. Mit der Gleichung (3.1) erhält man dann den virtuellen 2D Punkt im rechten bzw. im linken Bild, über den der Soll-Ist-Vergleich vorgenommen wird.

Ein weiteres Problem ist, daß die Grundfläche des Objektes relativ zum Loch verdreht sein kann. Da sich die Hauptachse HA_o des Objektes kollinear zur Normalen N_L der Lochebene befindet, kann diese Verdrehung nur eine einfache Rotation gegenüber der Z-Achse des Weltkoordinatensystems sein. Beim Zylinder kann dieses Problem ausgeschlossen werden, weil dessen Grundfläche rotationssymmetrisch ist. Bei dem Quader hingegen muß für eine paßgerechte Annäherung an das Loch eine geeignete Rotation gefunden werden, damit das Kantensegment, das den markanten Objektpunkt beinhaltet, die gleiche Orientierung wie die vordere Kante des Lochbrettes und damit implizit auch die gleiche Orientierung wie die vordere Kante des Loches selber hat. Die Abbildung 6.30 soll dies noch einmal verdeutlichen.

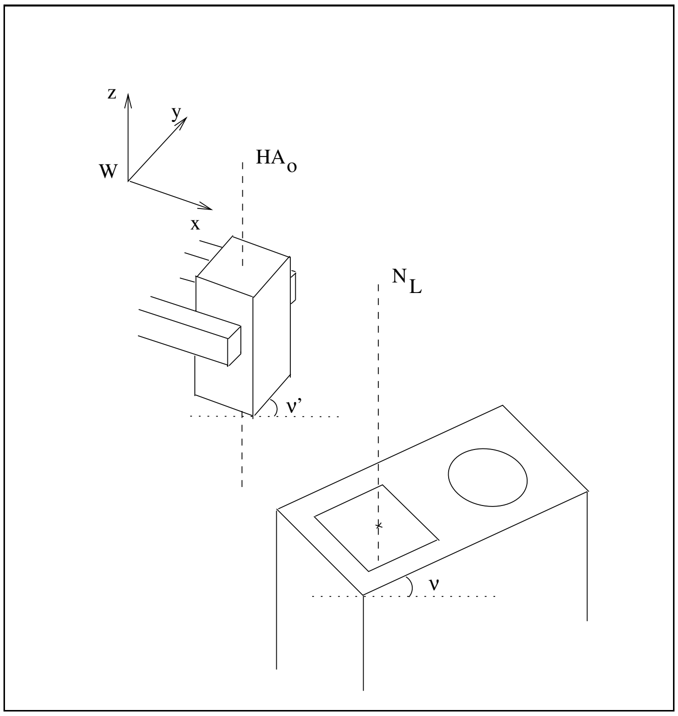


Abbildung 6.30: Mögliche Verdrehung des Objektes gegenüber der Lochberandung beim Quader. Für eine paßgerechte Annäherung muß durch eine geeignete Rotation um HA_o der Winkel ν' gleich dem Winkel ν werden.

Um diese Rotation zu erreichen, wird zuerst das Objekt auf den virtuellen Punkt über das Lochbrett geregelt. Hier wird dann in einer Regelschleife die Orientierung ν' der Objektkante im linken Bild bestimmt, mit der Orientierung ν ein Soll-Ist-Vergleich gemacht und der Quader um seine Hauptachse um $K_{P_R} \cdot (\nu' - \nu)$ Grad gedreht, bis der Fehler $(\nu' - \nu)$ gleich Null, bzw. kleiner einer Schranke δ_r ist. Die Drehung des Quaders um seine Hauptachse entspricht der Drehung um die X-Achse des Toolkoordinatensystems. In Eulerkoordinaten entspricht dies nach Gleichung (5.10) und (5.12) also einer Drehung um $(yaw = 90°, pitch = K_{P_R} \cdot (\nu' - \nu)°, roll = -90°)$ relativ zum Toolkoordinatensystem. Der Proportionalitätsfaktor kann beliebig, jedoch nicht zu hoch gewählt werden, eine geeignete Wahl ist z.B. $K_{P_R} = 0.5$.

Die Abbildung 6.31 zeigt ein Schema der geregelten Annäherung, die noch einmal den Zyklus von Bildaufnahme, Objekterkennung, und Annäherung des Objektes an das Loch verdeutlichen soll.

Die Abbildungen 6.32 und 6.32 zeigen Ergebnisse eines Testlaufs. In Abbildung 6.32 wurde der Abstand vom Objektpunkt zum Lochpunkt in jeder Vektorkomponente des Weltkoordinatensystems nach jeden Annäherungschritt gemessen. Man sieht hier jeweils den typischen asymptotischen Verlauf eines P-Reglers, der Proportionalitätsfaktor $\mathcal{K}_P$ ist 0.2. Der "Knick" in der Kurve der Z-Komponente beruht darauf, daß zuerst der virtuelle Punkt 10 cm über dem Punkt an dem Loch angesteuert wurde, und als der Punkt erreicht wurde der

Sollwert als die Projektion des eigentlichen Lochpunktes gesetzt wurde. Verlaufskurve Die Abbildung 6.33 zeigt die Trajektorie der Bewegung des Roboterarmes mit dem Objekt im 3D Raum des Weltkoordinatensystems. Für eine Abbildung vor und nach der Annäherung sein hier noch einmal auf Abbildung 2.7 verwiesen.

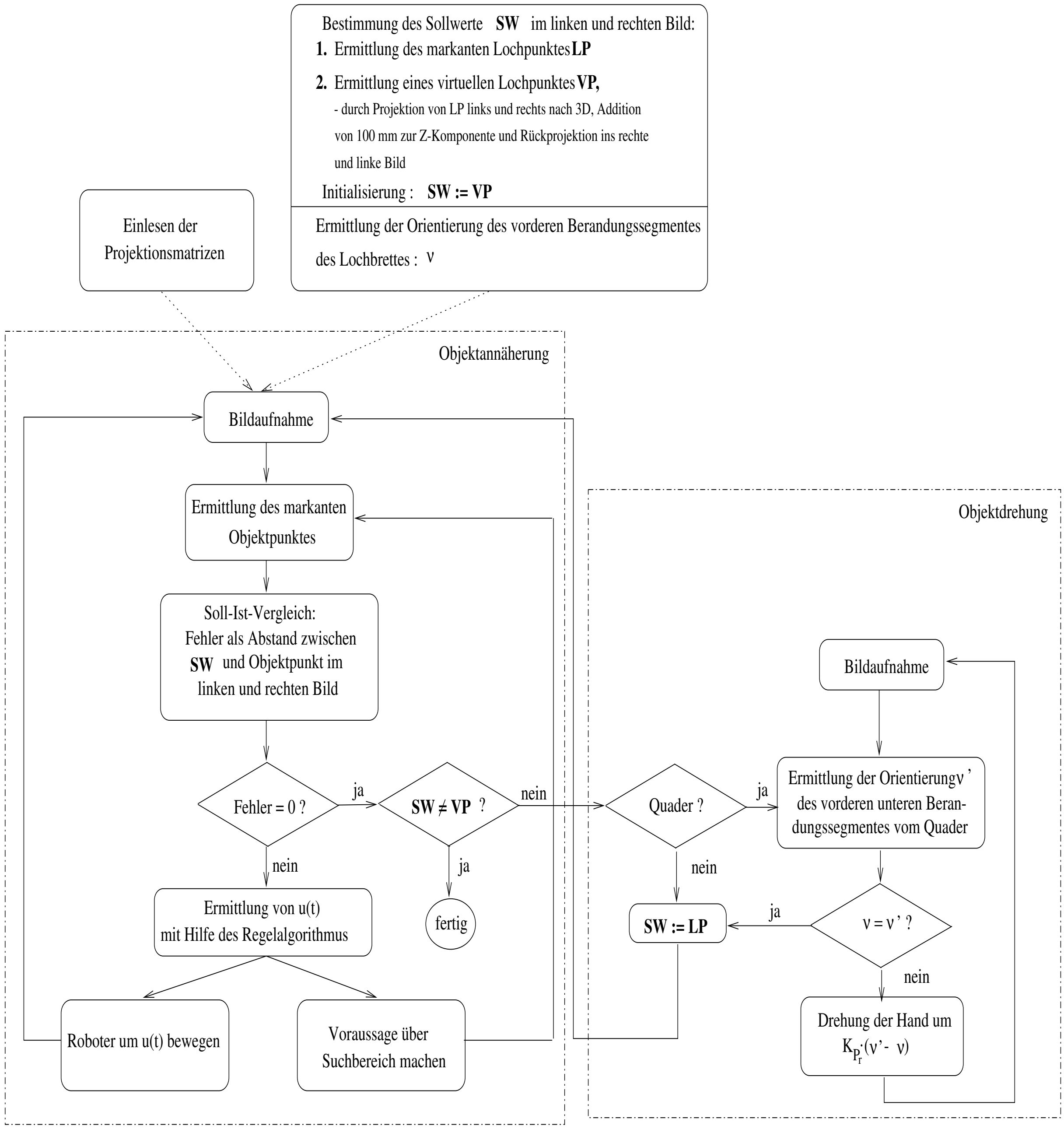


Abbildung 6.31: Schema für die geregelte Annäherung des Objektes an das für ihn bestimmte Loch und für die Drehung des Quaders am virtuellen Punkt.

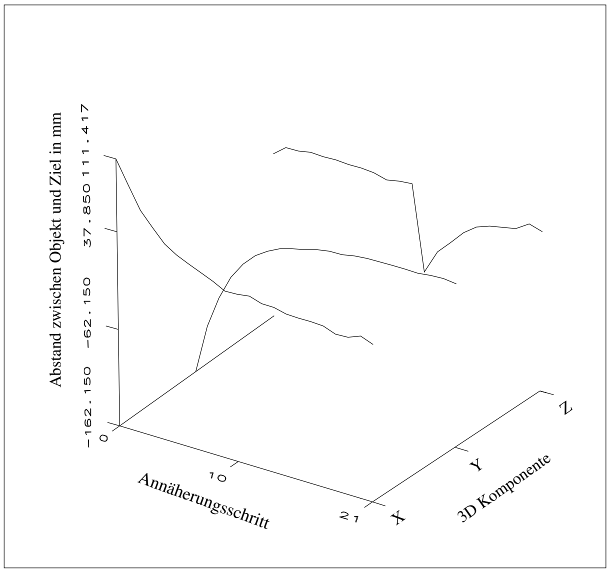


Abbildung 6.32: Ergebnis eines Testlaufs der Annäherung. Die Kurven zeigen den Verlauf des Abstandes vom Objektes zum Loch während der Annäherung in jeder Komponente des Weltkoordinatensystems.

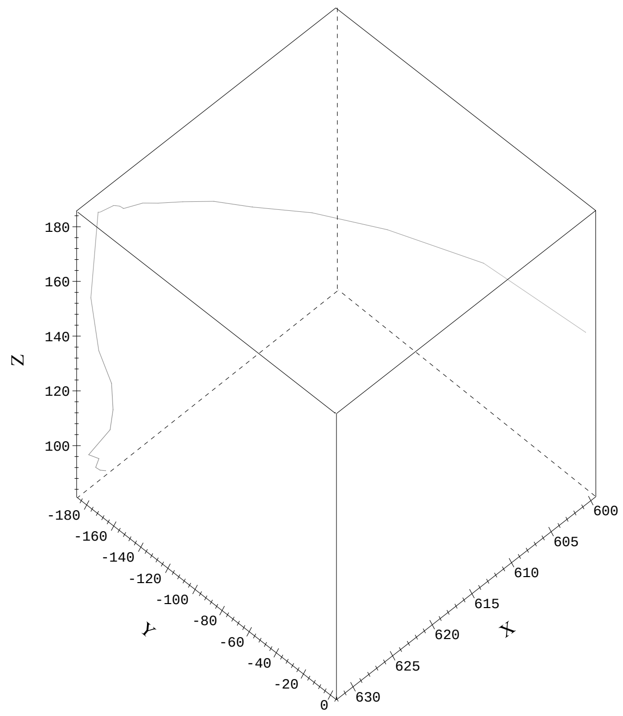


Abbildung 6.33: Ergebnis eines Testlaufes der Annäherung. Die Kurve zeigt die Bewegung des Roboterarmes mit dem Objekt im Weltkoordinatensystem.

Kapitel 7

Diskussion und Ausblick

In den vorherigen Kapiteln wurden die theoretischen Grundlagen und die Anwendung aller Module des Gesamtsystems gezeigt. Dieses Kapitel widmet sich nun der Diskussion der erzielten Ergebnisse und liefert einen Ausblick auf mögliche Erweiterungen und Verbesserungen des Systems.

7.1 Diskussion

Es werden mit Hilfe der Module verschiedene Beispiele für Arten von elementaren Wahrnehmungs-Aktions-Zyklen gezeigt, deren Kooperation ein Gesamtsystem darstellt, das die spezifische Aufgabe, ein Objekt an seiner Grundfläche zu erkennen und es an das für das Objekt bestimmte Loch anzunähern, löst. Das einzelne Modul wird hierbei eingesetzt, um die Aufgabe eines anderen Moduls zu unterstützen. Das Modul zur Ermittlung des Sehstrahls z.B. dient den Modulen der Objektausrichtung zur Kamera, die wiederum die Grundflächenerkennung vereinfachen. Das Modul der Vergenzregelung der rechten Kamera dient dem eigentlichen Annäherungsmodul, indem durch die Fixierung auf einen Startpunkt die Objekterkennung auf einen Suchbereich beschränkt wird. Erst das Zusammenspiel der Module macht das Lösen der Aufgabe möglich.

Der von uns entworfene Regelalgorithmus wird hierbei nicht nur für die Manipulation des Objektes - also für die Annäherung des Objektes an das Loch - verwendet, sondern auch für die Objektklassifikation, indem mit ihm der Sehstrahl der linken Kamera ermittelt wird. Dies ist ein neuer Aspekt in der Verwendung der visuellen Regelung, bei der bisher nur die Objektmanipulation, wie das Greifen eines Objektes oder die Lageveränderung eines Objektes, im Vordergrund stand. Unser Regelalgorithmus hat sich dabei als äußerst robust gegenüber Kalibrierungsstörungen erwiesen. Eine fehlerhafte Kalibrierung kann also durch den Regelungsalgorithmus ausgeglichen werden. Weiterhin sind für den Algorithmus keine expliziten Kenntnisse der extrinsischen und intrinsischen Kameraparameter nötig, so daß Fehler, die bei der Ermittlung dieser Parameter auftreten, ausgeschlossen werden können. Es ist also durchaus sinnvoll, diesen Ansatz der visuell basierten Regelung für eine Objektmanipulation weiter zu verfolgen.

Die für die Regelung bei der Annäherung nötige Erkennung des Objektes benötigt nur allgemeine Annahmen über Symmetrieeigenschaften des Objektes und keine expliziten Modellparameter. Das Verfahren verzichtet hierbei bewußt auf eine vollständige Rekonstruktion der gesamten Szene, sondern benutzt eine zielgerichtete Rekonstruktion für das Erreichen der Aufgabe, indem über eine Extraktion von relevanten Berandungssegmenten und Ausnutzung von Invarianten die Erkennung auf einen markanten Punkt reduziert wird. Dies ermöglicht eine enorme Zeitersparnis, so daß trotz der im Gegensatz zu anderen Arbeiten[1] verwendeten komplexeren Objekterkennung ein annehmbarer Regelungszyklus entsteht.

Der nächste Abschnitt beschreibt eine Einsatzmöglichkeit, sowie mögliche Erweiterungen und Verbesserungen des Systems.

7.2 Ausblick

Der Einsatz dieses Gesamtsystems ist im folgenden Themengebiet vorstellbar. Denkbar ist z.B. ein System in der Industrie, wo verschiedene Werkteile hinsichtlich ihrer Form klassifiziert und sortiert werden sollen. Da für das System z.Zt. nur eine Erkennung der symmetrischen Objekte Quader und Zylinder realisiert ist, muß hier eine Erweiterung implementiert werden, die eine Erkennung unter Ausnützung anderer Eigenschaften des Objektes ermöglicht. Eine sinnvolle Erweiterung des Systems ist in diesem Zusammenhang auch, den Manipulator die Objekte vorher von einem nicht vorherbestimmten Ort greifen zu lassen. Auch dieser Greifvorgang läßt sich mit unserem Regelungsalgorithmus realisieren. Hierbei ist der Sollwert der Punkt am Objekt, an dem die Finger zum Greifen ansetzen müssen, und der Istwert der Punkt genau zwischen den Fingern des Greifers. Das Problem liegt hierbei in der Objekterkennung. Wenn die Objektmenge aus den symmetrischen Objekten "Quader" und "Zylinder" besteht, läßt sich der Sollwert beim Quader als Mittelpunkt der Geraden zwischen den Mittelpunkten der Deck- und Grundfläche bestimmen. Die Mittelpunkte der Deck- und Grundfläche werden durch deren Berandungsegmente bestimmt, wobei das notwendige hintere Berandungssegment der Grundfläche durch eine Projektion des hinteren Berandungssegmentes der Deckfläche erhalten wird. Beim Zylinder ist ein möglicher Sollwert der Mittelpunkt der Geraden zwischen den Mittelpunkten der Halbellipsen, die die Deck- und Grundfläche beranden. Die Abbildung 7.1 soll die Lokalisation der Greifpunkte noch einmal an einer Skizze verdeutlichen.

Die Berandungssegmente des Quaders, der Finger sowie die Halbellipsen, die den Zylinder beranden, lassen sich mit den in Abschnitt 6.1 und 6.2 erläuterten Algorithmen extrahieren. Für den Quader muß zusätzlich die Orientierung der die Fingerspitzen des Manipulators nach vorne berandenden Segmente mit den vorderen Berandungssegmenten des Quaders

[1] siehe Kapitel 1.

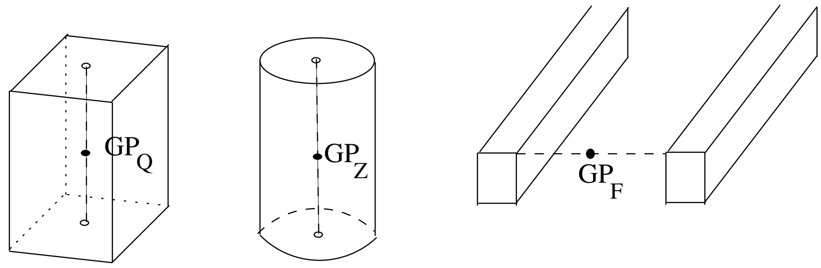


Abbildung 7.1: Mögliche Greifpunkte als Sollwert beim Quader und Zylinder und als Istwert bei den Roboterfingern.

übereinstimmen. Hierfür ist eine wie in Kapitel 6.4 beschriebene geregelte Drehung der Hand möglich.

Wenn man jedoch andere allgemeinere Objekte greifen will, müssen andere Verfahren zur Erkennung angewendet werden. Eine Möglichkeit ist z.B. ein Lernen der Ansichten und der Greifpunkte mit neuronalen Netzen [Tröster97].

Eine andere Ausbaumöglichkeit des Systems ist eine Erweiterung der Module in mehrere parallel arbeitende verhaltensbasierte Module, die sich nicht nur, wie in unserer Arbeit, sequentiell unterstützen sondern gleichzeitig miteinander arbeiten [Brooks86]. Denkbar wäre hierbei z.B. ein Modul, das das System während der Annäherung des Objektes unterstützt, indem es über einen am Greifer angebrachten Sensor eine fortwährende Hindernisabfrage vornimmt, ein anderes Modul könnte für die Abfrage zuständig sein, daß der Greifer nicht aus dem Bildbereich des Kopfes geht, usw. Die Implementation dieser Multiagentensysteme verlangt allerdings größere Rechenkapazitäten, da unter der Bedingung der Parallelität für jeden Prozeß auch ein eigener Prozessor vorhanden sein muß. Eine Umsetzung in viele kleine verhaltensbasierte parallel laufende Module auf unserem mit nur vier Prozessoren ausgestatteten Multithreadprozessorsystem ist daher z.Zt. noch nicht möglich, da allein schon durch das Stereosehen zwei Prozessoren für die Extraktion parallel laufen.

Anhang A

Mathematische Ergänzungen

A.1 Herleitung der Modellparameter

In diesem Abschnitt soll die Herleitung der in Kapitel 5 gezeigten Modellparameter $(\zeta_1, \ldots, \zeta_m, \eta)$ aus der Minimierung des Quadratischen Fehlers

$$\rho \;=\; \frac{1}{N}\sum_{t=1}^{N}\left(r_t - \eta - \sum_{j=1}^{m}\zeta_j r_{t-j}\right)^2 \tag{A.1}$$

gezeigt werden. Die Herleitung geschieht mit der Methode der kleinsten Quadrate, indem man ρ minimiert.

$$\rho \rightarrow min! \tag{A.2}$$

Dies läßt sich erreichen, indem man ρ partiell nach den Unbekannten ableitet und die Gleichungen gleich Null setzt (notwendige Bedingung für ein Extremum).

$$\begin{aligned}
\frac{\partial\rho}{\partial\zeta_k} &= -\frac{2}{N}\sum_{t=1}^{N}\left(r_t - \eta - \sum_{j=1}^{m}\zeta_j r_{t-j}\right) * r_{t-k} = 0, \quad k = 1, ..., m \text{ und} \\
\frac{\partial\rho}{\partial\eta} &= -\frac{2}{N}\sum_{t=1}^{N}\left(r_t - \eta - \sum_{j=1}^{m}\zeta_j r_{t-j}\right) = 0.
\end{aligned} \tag{A.3}$$

Durch Umformung erhält man die Normalgleichungen:

$$\begin{aligned}
\sum_{t=1}^{N} r_t r_{t-k} &= \sum_{t=1}^{N}\eta\zeta_{t-k} + \sum_{j=1}^{m}\zeta_j\sum_{t=1}^{N} r_{t-k}r_{t-j}, \quad k = 1, ..., m \;\text{ und} \\
\sum_{t=1}^{N} r_t &= \sum_{t=1}^{N}\eta + \sum_{j=1}^{m}\zeta_j\sum_{t=1}^{N} r_{t-j}.
\end{aligned} \tag{A.4}$$

In Matrixschreibweise ergibt sich:

$$\begin{bmatrix} \sum\limits_{t=1}^{N} r_{t-1}r_t \\ \sum\limits_{t=1}^{N} r_{t-2}r_t \\ \vdots \\ \sum\limits_{t=1}^{N} r_{t-m}r_t \\ \sum\limits_{t=1}^{N} r_t \end{bmatrix} = \begin{bmatrix} \sum\limits_{t=1}^{N} {r^2}_{t-1} & \cdots & \sum\limits_{t=1}^{N} r_{t-1}r_{t-m} & \sum\limits_{t=1}^{N} r_{t-1} \\ \sum\limits_{t=1}^{N} r_{t-2}r_{t-1} & \cdots & \sum\limits_{t=1}^{N} r_{t-2}r_{t-m} & \sum\limits_{t=1}^{N} r_{t-2} \\ \vdots & \ddots & \vdots & \vdots \\ \sum\limits_{t=1}^{N} r_{t-m}r_{t-1} & \cdots & \sum\limits_{t=1}^{N} {r^2}_{t-m} & \sum\limits_{t=1}^{N} r_{t-m} \\ \sum\limits_{t=1}^{N} r_{t-1} & \cdots & \sum\limits_{t=1}^{N} r_{t-m} & N \end{bmatrix} \cdot \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_m \\ \eta \end{bmatrix} \tag{A.5}$$

Die Lösung für die gesuchten Modellparameter ist also:

$$\begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_m \\ \eta \end{bmatrix} = \begin{bmatrix} \sum\limits_{t=1}^{N} {r^2}_{t-1} & \cdots & \sum\limits_{t=1}^{N} r_{t-1}r_{t-m} & \sum\limits_{t=1}^{N} r_{t-1} \\ \sum\limits_{t=1}^{N} r_{t-2}r_{t-1} & \cdots & \sum\limits_{t=1}^{N} r_{t-2}r_{t-m} & \sum\limits_{t=1}^{N} r_{t-2} \\ \vdots & \ddots & \vdots & \vdots \\ \sum\limits_{t=1}^{N} r_{t-m}r_{t-1} & \cdots & \sum\limits_{t=1}^{N} {r^2}_{t-m} & \sum\limits_{t=1}^{N} r_{t-m} \\ \sum\limits_{t=1}^{N} r_{t-1} & \cdots & \sum\limits_{t=1}^{N} r_{t-m} & N \end{bmatrix}^{-1} \cdot \begin{bmatrix} \sum\limits_{t=1}^{N} r_{t-1}r_t \\ \sum\limits_{t=1}^{N} r_{t-2}r_t \\ \vdots \\ \sum\limits_{t=1}^{N} r_{t-m}r_t \\ \sum\limits_{t=1}^{N} r_t \end{bmatrix} \tag{A.6}$$

A.2 Generalisiertes Eigenwertproblem

In diesem Abschnitt soll die Lösung eines generalisierten Eigenwertproblems der Form

$$\boldsymbol{C}\boldsymbol{a} = \lambda \boldsymbol{S}\boldsymbol{a} \tag{A.7}$$

gezeigt werden[1]. Die Eigenwerte und Eigenvektoren dieses Problems werden ebenso als generalisiert bezeichnet.

Falls $\boldsymbol{S}$ nicht singulär ist, läßt sich Gleichung (A.7) umformen zu

$$(\boldsymbol{S}^{-1}\boldsymbol{C})\boldsymbol{a} = \lambda \boldsymbol{a} \tag{A.8}$$

Die Matrix $\boldsymbol{S}^{-1}\boldsymbol{C}$ ist nicht symmetrisch, aber es läßt sich ein symmetrisches Eigenwertproblem formulieren, indem man die Cholesky-Zerlegung $\boldsymbol{S} = \boldsymbol{L} \cdot \boldsymbol{L}^T$ anwendet, wobei $\boldsymbol{L}$ eine untere Dreiecksmatrix ist. Wenn man nun Gleichung (A.7) mit $\boldsymbol{L}^{-1}$ multipliziert, erhält man

$$\boldsymbol{W} \cdot (\boldsymbol{L}^T \cdot \boldsymbol{a}) \;=\; \lambda(\boldsymbol{L}^T \cdot \boldsymbol{a}), \tag{A.9}$$

wobei

$$\boldsymbol{W} = \boldsymbol{L}^{-1} \cdot \boldsymbol{C} \cdot (\boldsymbol{L}^{-1})^T \tag{A.10}$$

ist. Die Matrix $\boldsymbol{W}$ ist symmetrisch und die Eigenwerte dieser Matrix entsprechen den Eigenwerten des Originalproblems (A.7). Die Eigenvektoren sind $\boldsymbol{L}^T \cdot \boldsymbol{a}$.

[1] vgl. Kap. 6.2.1.

Literaturverzeichnis

[Atiquzzaman94] M. Atiquzzaman, M.W. Akthar *Complete Line Segment Description Using the Hough Transform* Image and Computer Vision Bd. 12, Nr. 5, Juni 1994.

[Allen93] P.K. Allen, A. Timcenko *Automated Tracking and Grasping of a Moving Objekt with a Robotic Hand-Eye System* IEEE Transactions on Robotics and Automation Bd. 9, S. 152-165, April 1993.

[Binford93] T.O. Binford, T.S. Levitt *Quasi-Inavriants: Theory and Exploitation* Defense Advanced Research Projects Agency(DARPA)'93, S. 819-829, 1993.

[Bookstein79] F.L. Bookstein *Fitting Conic Sections to Scattered Data* Computer Graphics and Image Processing Bd. 9, S. 56-71, 1979.

[Brooks86] R. Brooks *A Robust Layered Control System for a Mobile Robot* IEEE Journal of Robotics and Automation Bd. 2, S. 14-23, 1986.

[Castaño94] A. Castaño, S. Hutchinson *Visual Compliance: Task-Directed Visual Servo Control* IEEE Transactions on Robotics and Automation Bd. 10, Nr. 3, S. 334-342, Juni 1994.

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

[Corke93] P.I. Corke *Visual Control of Robot Manipulators - A Review* Robotics and Automated Systems Bd. 7, S. 1-31, 1993.

[Dubois86] S.R. Dubois *An Autoregressive Model Approach to Two-Dimensional Shape Classification* IEEE Transactions on Pattern Analysis and Machine Intelligence Bd. 8, Nr. 1 , Januar 1986.

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

[Fitzgibbon95] A. Fitzgibbon, R. Fisher *A Buyer's Guide to Conic Fitting* Proceedings of British Machine Vision Conference, Birmingham, 1995.

[Fitzgibbon96] A. Fitzgibbon, M. Pilu, R. Fisher *Direct Least-Square Fitting of Ellipses* International Conference on Pattern Recognition, Vienna, August 1996.

[Freeman77] H. Freeman, L.S. Davis *A Corner Finding Algorithm for Chain Coded Curves* IEEE Trans. Comput. Bd. 26, S. 297-303, 1977.

[Gander94] W. Gander, G.H. Golub, R. Strebel *Least-Square Fitting of Circles and Ellipses* BIT Bd.43, S. 558-578, 1994.

[Hager95] G.D. Hager, W. Chang, A.S. Morse *Robot Hand-Eye Coordination Based on Stereo Vision* IEEE Control Systems Magazine, Februar 1995.

[Harrell89] R.C. Harrell, D.C. Slaughter, P.D. Adsit *A Fruit-Tracking System For Robotic Harvesting.* Machine Vision and Applications, S. 69-80,1989.

[Hill79] J. Hill, W.T. Park *Real Time Control of a Robot with a Mobile Camera* Proc. 9th ISIR, Washington D.C., S. 233-246, März 1979.

[Ho95] C.-T. Ho, L.-H. Chen *A Fast Ellipse/Circle Detector Using Geometric Symmetry* Pattern Reconition, Bd. 28, Nr. 1, S. 117-124, 1995.

[Hough62] P.V.C. Hough *Methods and Means for Recognizing Complex Patterns* U.S. Patent 3,069,654.

[Hu62] M.K. Hu *Visual Pattern Recognition by Moment Invariants* IEEE Trans. Inform. Theory, Bd. 8, S. 179-187, Februar 1962.

[Hutchinson96] S. Hutchinson, G.D. Hager, P.I. Corke *A Tutorial on Visual Servo Control* IEEE Transactions on Robotics and Automation, Bd. 12, Nr. 5, Oktober 1996.

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

[Jain95] R. Jain *Machine Vision* International Edition, 1995.

[Kashyap81] R.L. Kashyap, R. Chellappa *Stochastic Models for Closed Boundary Analysis: Representation and Reconstruction* IEEE Trans. Inform. Theory, Bd. 27, S. 627-637, 1981.

[Korn61] G.A. Korn, T.M. Korn *Mathematical Handbook for Scientists and Engineers* Mc Graw-Hill Book Co., 1961.

[Kunze96] S. Kunze *Visuell basiertes Anordnen von Werkstücken mit dem Greifer eines Roboterarmes* Studienarbeit am Lehrstuhl für Kognitive Systeme der Christian-Albrechts-Universität zu Kiel, August 1996.

[Leavers92] V.F. Leavers *Survey: Which Hough Transform?* Computer Vision, Graphics and Image Processing - Image Understanding, Bd. 58, Nr. 2, S. 250-264, September 1993.

[Mundy92] J.L. Mundy, A. Zisserman *Geometric Invariance in Computer Vision* The MIT Press, 1992.

[Nair95] P.S. Nair, A.T. Saunders *Hough Transform Based Ellipse Detection Algorithm* Pattern Recognition Letters 17, S. 777-784, 1996.

[Pauli97] J. Pauli *Projective Invariance and Orientation Consensus for Extracting Boundary Configurations* Mustererkennung 1997, 19. DAGM-Symposium Braunschweig, S. 375-383, September 1997.

[Rosin93] P.L. Rosin *A Note on the Least Square Fitting of Ellipses* Pattern Recognition Letters, (14):799-808, Oktober 1993.

[Sarkar93] S. Sarkar, K.L. Boyer *Perceptual Organization in Computer Vision: A Review and a Proposal for a Classificarory Structure* IEEE Transaction on Systems, Man, and Cybernetics, Bd. 23, Nr. 2, März/April 1993.

[Sanderson80] A.C. Sanderson, L.E. Weiss *Image-based Visual Servo Control Using Relational Graph Error Signals* Proc. IEEE, S. 1074-1077, 1980.

[Soltyasik89] R. Soltysiak *Wissensbasierte Prozeßregelung* Handbuch der Informatik 11.2, 1989.

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

[Tröster97] J. Tröster *Visuell basiertes Greifen gestützt auf neuronale Netze* Diplomarbeit am Lehrstuhl für Kognitive Systeme der Christian-Albrechts-Universität zu Kiel, 1997.

[Veen80] T.M. van Veen, F.C.A. Groen *Discretization Errors in the Hough Transformation* Pattern Recognition Bd. 14, Nr. 1-6, S. 137-145, 1981.

[Wunsch96] P. Wunsch, G. Hirzinger *Echtzeit-Lagebestimmung dreidimensionaler Objekte aus Bildsequenzen zur visuellen Lageregelung eines Industrieroboters* Proceedings in Artificial Intelligence, Bd. 4 . Infix-Verlag, S. 166-173, 1996.

[Zahn72] C.T. Zahn, R.S. Roskies *Fourier descriptors for plane closed curves* IEEE Trans. Comput., Bd. C-21, S. 269-281, März 1972.

[Zhang95] Z. Zhang *Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting* INRIA Nr. 2676, Oktober 1995.

Danksagung

An dieser Stelle möchte ich allen meinen tiefsten Dank aussprechen, die mir bei der Entstehung dieser Arbeit geholfen haben.

Insbesondere Danke ich meinen Eltern, die immer für mich da waren und mich moralisch und finanziell unterstützt und mein Studium überhaupt erst ermöglicht haben. Einen großen Dank gilt auch Sabine, die jedesmal eine Schulter frei hatte, falls ein kreativer Tiefpunkt bei mir vorherrschte.

Ich danke weiterhin meinen Betreuer Josef Pauli, der trotz seiner vielen anderen Arbeit immer ein offenes Ohr für meine Probleme hatte und die Arbeit durch kreative Kritik und neuen Ideen immer wieder auf die richtige Bahn gelenkt hat.

Zuletzt danke ich noch meinen Korrekturleserinnen Judith und Sabine, die ihr möglichstes getan haben, die Arbeit lexikalisch und grammatikalisch zu "säubern".

Erklärung:

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

Kiel, den