Ein Hand-Auge-System zur visuell basierten Lokalisierung und Identifikation von Objekten

Diplomarbeit

am Lehrstuhl für Kognitive Systeme
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Prof. Dr. Gerald Sommer

vorgelegt von
Stefan Kunze

Betreuer: Dr. Josef Pauli

Januar 1999

Zusammenfassung

In dieser Diplomarbeit wurde ein Hand-Auge-System entwickelt, welches die Fähigkeit hat, durch polygonale Objektberandungen charakterisierbare, ebene Oberflächen von Objekten in einer komplexen Szene zu erkennen und diese dann nach weiteren, auf der Oberfläche befindlichen Objekten abzusuchen. Die visuellen Daten werden durch ein am Roboterarm montiertes Kamerasystem gewonnen, das durch den Roboterarm gezielt im Arbeitsbereich positioniert und orientiert werden kann. Das Auffinden der Oberfläche erfolgt in hierarchischer Weise. Ausgehend von Bildgeraden werden über mehrere Zwischenstufen die 3D-Modelle der in der Szene befindlichen Objektoberflächen berechnet. Die Gewinnung der 3D-Information ist notwendig, damit der Roboter das Kamerasystem in eine bestimmte Position und Orientierung zur Oberfläche eines in der Szene befindlichen Objektes bringen kann. Es werden insbesondere eingesetzt: die orientierungsselektive Hough-Transformation zur Detektion von Geraden, ein Verfahren zur Berechnung von Korrespondenzen von Geraden, die Rekonstruktion von Geraden und Punkten, ein Verfahren zur Suche von Kreisen in Graphen, sowie ein Verfahren zum Vergleich von polygonalen Konturen. Dabei werden wechselseitige Beziehungen zwischen Geometrie und Photometrie ausgenutzt. Die Detektion und Lokalisation von kleinen Objekten auf der Objektoberfläche erfolgt nach dem erscheinungsbasierten, neuronalen Ansatz. Hier werden insbesondere RBF-Netze eingesetzt. Es wird ein lernfähiger, lokaler, verschiebungsinvarianter und annähernd beleuchtungsinvarianter Operator zur Detektion von Objekten entwickelt. Ein Anwendungsbeispiel ist die Detektion von Elektronikbauteilen auf Platinen, an dem das System getestet wurde.

Inhaltsverzeichnis

Kapitel 1

Einführung

1.1 Problemstellung

Biologische Systeme wie beispielsweise der Mensch machen zur Wahrnehmung ihrer Umwelt von ihrer Fähigkeit Gebrauch, die Parameter ihrer visuellen Sensoren auf unterschiedlichste Art und Weise anzupassen. Sucht man z. B. eine Stecknadel, von der man weiß, daß diese sich auf einem Tisch im Raum befindet, so wird man zunächst nach dem Tisch Ausschau halten, sich auf diesen zubewegen und den Tisch in einem geeigneten Abstand und Blickwinkel absuchen. Eine direkte Suche dieses kleinen Objektes im ganzen Raum kann aus verschiedenen Gründen ein ungeeignetes Vorgehen sein: das Objekt ist relativ zur Auflösung des Sensors zu klein, die momentane Position, aus der man in Richtung des Objektes guckt, verhindert eine Erkennung. Die Tatsache, das wir nicht passive, sondern aktive Beobachter sind, eröffnet uns eine Reihe von Strategien zur Objekterkennung.

Der Einsatz einer aktiven Kamera läßt sich in analoger Weise für technische Systeme motivieren (*active vision* [1]). Die am obigen Beispiel erläuterte Strategie läßt sich in die drei folgenden Abschnitte einteilen:

1. Erkennung umfassender, großer Objekte in der Szene, unter denen sich ein im Bezug auf die Zielstellung eindeutig bestimmbares Objekt, das Zielobjekt, befindet.
2. Bewegung des Kamerasystems in eine bestimmte Position und Orientierung zum Zielobjekt.
3. Erkennung kleiner Objekte auf dem angesteuerten Objekt von der neuen Blickposition aus.

Die Fähigkeit der Kamera, Position und Orientierung zu verändern, wird in dieser Arbeit durch die Montage des Kamerasystems an den Endeffektor eines Roboterarmes ermöglicht. Ein solches System wird als Hand-Auge-System bezeichnet.

Die Schwierigkeit liegt in der Entwicklung oder Auswahl geeigneter Objekterkennungsverfahren. Die Objekterkennung besteht in der Trennung zwischen Objekt und Hintergrund der Szene (*figure-ground separation* [43]). Was in der Szene Objekt oder Hintergrund ist, hängt von der Interpretation ab.

Die Erkennung der Objekte muß auf Basis geeigneter Merkmale und Informationen erfolgen. Welche Merkmale und Informationen aus den Bilddaten extrahiert werden, und welche Verfahren hierzu angewendet werden, hängt von der jeweiligen Zielstellung und den Umweltbedingungen ab.

Im Falle der Erkennung des großen Objektes reicht, im Gegensatz zum kleinen Objekt, eine Lokalisation und Identifikation des gesuchten Objektes im Bild nicht aus. Da das Kamerasystem durch den Manipulator positioniert und orientiert werden soll, muß zusätzlich die Position und die Orientierung des großen Objektes in der dreidimensionalen Szene berechnet werden, sowohl zur Selbstlokalisation des Kamerasystems und somit des Roboters, als auch zur anschließenden Bewegung des Kamerasystems in die gewünschte Stellung relativ zum Objekt durch den Manipulator. Da nicht a priori bekannt ist, wo und in welcher Orientierung sich das große Objekt in der Szene befindet, muß dieses Verfahren invariant gegenüber Lage und Orientierung des Objektes sein.

Entscheidend für die Auswahl und Entwicklung der Verfahren ist auch die Komplexität der Szene. Diese wird bestimmt durch das Vorhandensein weiterer Objekte außer dem Zielobjekt in der Szene und mögliche Okklusionen des Zielobjektes. Einerseits ist eine vollständige, präzise 3D-Rekonstruktion des Zielobjektes unnötig und unmöglich, da ein Objekt meistens nicht von allen Seiten beobachtbar ist, andererseits muß die Modellierung detailiert genug sein, um von weiteren in der Szene befindlichen Objekten unterscheiden zu können. Der Verzicht auf eine möglichst vollständige und perfekte Modellierung der Umwelt ist eine der wesentlichen Grundsätze verhaltensbasierter Systeme und führt zu Verfahren größerer Robustheit und geringerer Komplexität [4]. Insgesamt wird außerdem erwartet, daß die Verfahren innerhalb eines bestimmten Rahmens in nicht zu speziellen Beleuchtungsverhältnissen anwendbar sind.

Interessant ist auch die Frage nach der Flexibilität und der Lernfähigkeit der Verfahren bezüglich ähnlicher oder anderer Objekte.
Klassische Verfahren basieren auf einer geometrischen, symbolischen Beschreibung der Objekte, in Form höherer Datenstrukturen (z. B. Listen, Graphen). Die Kontur eines flachen, eckigen Objektes läßt sich beispielsweise durch ein Polygon, also einer Liste von Vektoren, beschreiben. Dann werden Ähnlichkeitsmaße benötigt, die zwei Repräsentationen vergleichbar machen. Dabei kommt der Invarianz der Verfahren gegenüber Verschiebung, Rotation und Skalierung je nach Aufgabenstellung eine unterschiedliche Bedeutung zu. Zur Identifikation der in einem oder mehreren Bildern lokalisierten Objekte erfolgt ein Vergleich mit einem in einer Modelldatenbank gespeichertem Modell. Auf diese Weise wird das Problem gelöst, den aus den Bilddaten extrahierten Strukturen Bedeutung zu geben [12]. Sollen jedoch Objekte mit anderen Eigenschaften, also z. B. runde Objekte statt eckige, erkannt werden, kann eine Repräsentation durch andere Datentypen notwendig werden und somit auch die Entwicklung neuer Algorithmen.
Einen anderen Weg bieten erscheinungsbasierte Verfahren in Verbindung mit neuronalen, konnektionistischen Ansätzen. Die Basis für erscheinungsbasierte Systeme bilden zweidimensionale Ansichten eines Objektes. Merkmale wie beispielsweise die Form des Objektes

werden hier im Gegensatz zu den klassischen Verfahren nicht explizit modelliert, photometrische Eigenschaften, Orientierung des Objektes und Beleuchtungsverhältnisse sind als Ganzes in einer Ansicht implizit enthalten. Erscheinungsbasierte Verfahren nutzen die hohe Korrelation zwischen den verschiedenen Ansichten eines Objektes. Eine Karhunen-Loeve-Transformation dieser Datenmenge verlagert die weitere Objekterkennung in einen relativ niedrigdimensionalen Unterraum [17, 20, 21, 23, 36].
Neuronale Ansätze haben im Gegensatz zu klassischen Methoden den Vorteil, daß sich die zugrundeliegenden Algorithmen nicht ändern, es muß lediglich eine Menge von Parametern neu angepaßt werden. Ein praktischer Vorteil ist auch die hohe Parallelisierbarkeit und die verhältnismäßig einfachen Operationen auf den Daten.

In dieser Arbeit kommen beide Paradigmen der Objekterkennung zur Anwendung: "...the time has come to build systems out of diverse components..." [18]. Die Erkennung des großen, umgebenden Objektes erfolgt nach dem geometrischen, modellbasierten Ansatz, die Erkennung kleiner Objekte nach dem erscheinungsbasierten Ansatz, wobei die Klassifikation unter Einsatz Radialer Basis Funktionen Netze erfolgt.

Diese Vorgehensweise soll an dem Beispiel der Detektion von Elektronikbauteilen („kleine" Objekte) auf Platinen („große" Objekte) genauer motiviert werden.

Die Menge der Elektronikbauteile besteht aus einem Spannungsregler, einem Elektrolytkondensator, einem Widerstand, einem IC-Chip, einem Transistor und einem Potentiometer. Abb. 1.2 zeigt Bilder dieser Objekte auf homogenem Hintergrund.

Es fällt auf, daß die Objekte keinerlei Textur aufweisen. Die Objekte sind von unterschied-

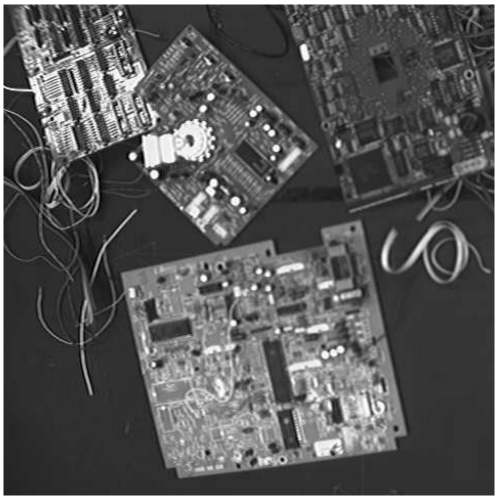
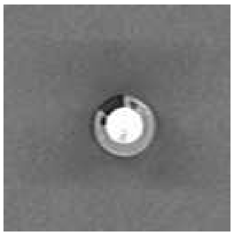


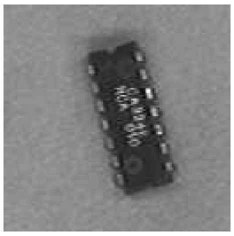
Abbildung 1.1: Szene mit Platinen

(a) Spannungsregler



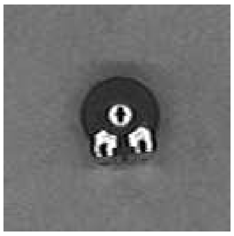
(b) Kondensator

(c) Widerstand



(d) IC

(e) Transistor



(f) Potentiometer

Abbildung 1.2: Bilder der Elektronikbauteile licher Form und Gestalt und weisen nur sehr wenige Details auf. Der Mangel an Details und Textur würde eine auf den Kanten der Bilder basierendes, modellbasiertes Verfahren begünstigen, allerdings erfordert das Vorhandensein der verschiedenen Formen eine aufwendige Objektmodellierung, auch aus dem Grunde, daß die Objekte aufgrund ihrer äußeren Kontur allein nicht unterscheidbar sind. Beispielsweise sind der Kondensator und das Potentiometer kreisförmig, so daß zur Unterscheidung geometrisch nur schwer modellierbare Details hinzugezogen werden müssen. Erschwerend kommt hinzu, daß der Hintergrund, auf dem sich die zu erkennenden Objekte in der Anwendungsphase befinden, inhomogen ist (siehe Abb. 1.1).

Daher erfolgt die Erkennung dieser Objekte mit dem erscheinungsbasierten Ansatz. Das Fehlen von Textur und die wenigen Details führen bei einer Karhunen-Loeve-Transformation zu sehr niedrigdimensionalen Unterräumen. Im Unterschied zu den Arbeiten von Murase und Nayar [23, 20, 21] erfolgt die weitere Klassifikation in den Unterräumen jedoch durch RBF-Netze. Ebenso wird die Laplacepyramide zwecks Merkmalsextraktion und Dimensionsreduktion als Vorverarbeitung eingesetzt.

Da die optische Achse der Kamera durch den Einsatz des Roboters in eine senkrechte Orientierung und einen bestimmten Abstand zur Platine gebracht werden kann, genügt die Menge der Ansichten, die man durch Rotation des Objektes um deren senkrechte Achse in einem festen Abstand erhält. Insgesamt ergibt sich ein nichtlinearer, verschiebungsinvarianter und annähernd beleuchtungsinvarianter Operator zur Detektion gelernter Objekte.

Zur Erkennung der Platine ist der erscheinungsbasierte Ansatz in der soeben beschriebenen Form nicht anwendbar. Die Platine zeichnet sich allein durch ihre polygonförmige Kontur aus. Eine Objekterkennung muß invariant gegenüber den auf der Platine befindlichen Objekten sein. Diese können in beliebiger Anzahl und Position auf der Platine verteilt sein und sind in einer Ansicht gegenüber der relevanten Kontur überrepräsentiert. Um diese Invarianz zu erreichen, müßten also nicht nur die Ansichten verschiedener Orientierungen der Platine in ein Lernverfahren einfließen, sondern auch Ansichten der Platinen mit einer Vielzahl unterschiedlicher Objekte an allen möglichen Positionen auf der Platine selbst, was die Lernaufgabe somit unlösbar macht.

In der Arbeit wurde daher zur Erkennung der Platine der modellbasierte Weg gewählt. Dabei wurde das Prinzip der perzeptuellen Organisation angewendet [31, 32, 41]. Dieses besteht darin, stufenweise Merkmale und Repräsentationen zu neuen, komplexeren zu verbinden, von Merkmalen auf der Signalebene bis hin zu symbolischen, geometrischen Beschreibungen der zu extrahierenden Objekte.

1.2 Kurzbeschreibung des Systems

Abbildung 1.3 skizziert die wesentlichen Komponenten des Systems und deren Beziehung. Es teilt sich in eine Vorbereitungs- und eine Anwendungsphase.

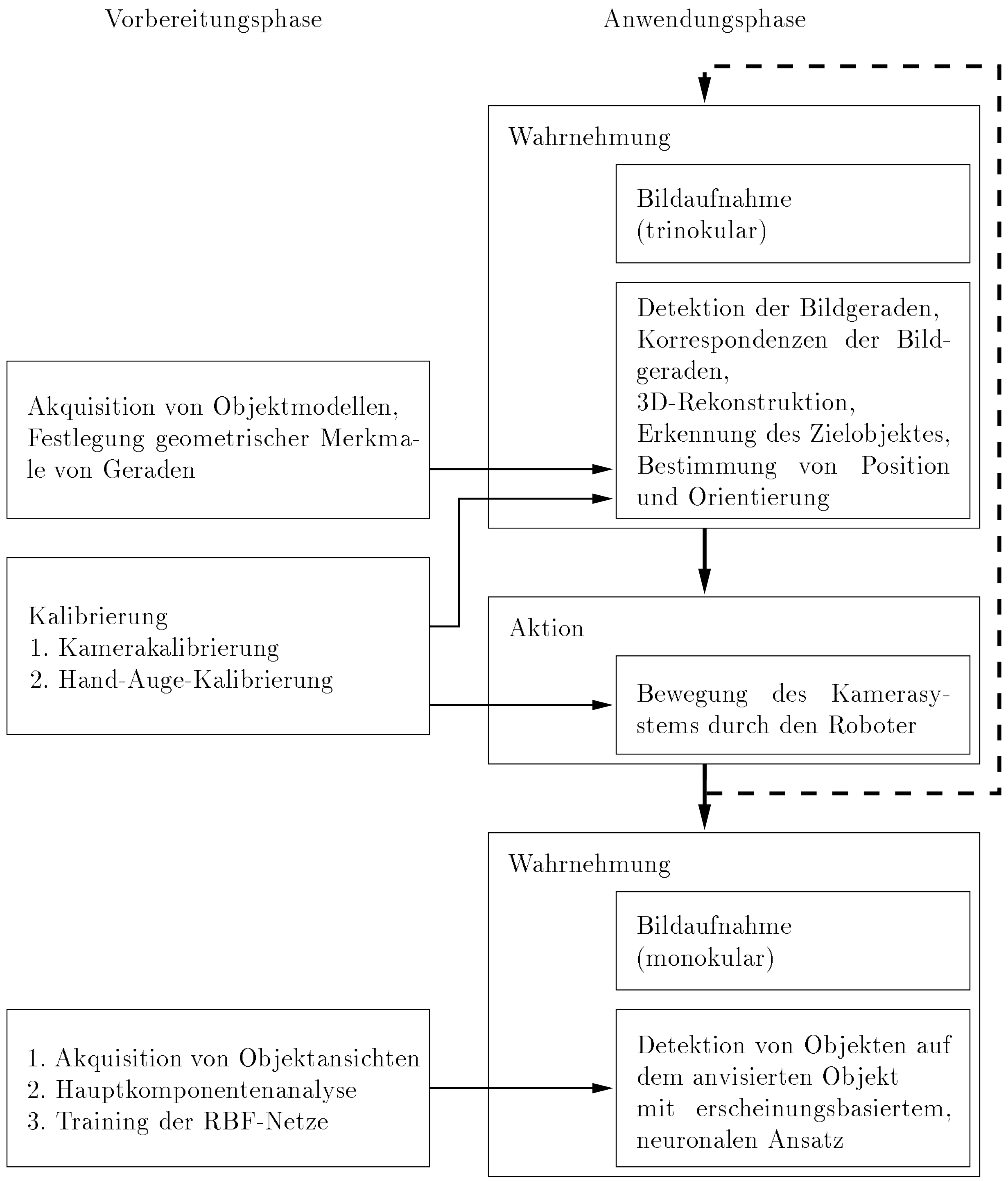


Abbildung 1.3: Gesamtablauf

Für die 3D-Rekonstruktion der Objekte ist eine Kamerakalibrierung notwendig. Die Bewegung des Kamerasystems relativ zur erkannten Oberfläche setzt eine Hand-Auge-Kalibrierung voraus, durch welche die räumliche Beziehung zwischen dem Endeffektor des Roboters und dem Kamerasystem geschätzt wird.

Für die modellbasierte Erkennung der Objektoberfläche ist das Speichern einiger Objektmodelle nötig, mit denen die in der Szene rekonstruierten und lokalisierten Oberflächen verglichen und identifiziert werden. Diese können visuell basiert gewonnen werden, indem die Verfahren zur Extraktion der Oberflächen, die ohnehin Teil des Erkennungsverfahrens sind, auf eine triviale Szene angewendet werden. In dieser Szene befindet sich ein planares, homogenes Modellobjekt auf ebenfalls homogenem Hintergrund, so daß sich genau die Geraden der Objektkontur ergeben.
Auf geometrische Merkmale basierende Kriterien reduzieren den Rechenaufwand bei der Erkennung. Diese Merkmale werden a priori festgelegt.

Für die erscheinungsbasierte, neuronale Erkennung müssen zunächst die Ansichten der zu erkennenden „kleinen“ Objekte aufgenommen werden. Die Aufnahme der Ansichten erfolgt automatisch unter Verwendung eines Drehtellers, der über die Peripherie des Rechners angesteuert wird. Bevor die RBF-Netze der Bildoperatoren trainiert werden, wird eine Hauptkomponentenanalyse der vorverarbeiteten Ansichten durchgeführt.

Die Anwendungsphase besteht aus der Erkennung einer Zieloberfläche, der Bewegung des Kamerasystems relativ zur Oberfläche, so daß eine weitere Kamera mit einem festen Abstand senkrecht auf die Oberfläche blickt, sowie die Erkennung von Objekten nach erfolgter Bildaufnahme durch diese Kamera. Bildaufnahme, Erkennung der Oberfläche und Bewegung müssen in Form eines Zyklus aus Gründen, die im weiteren Verlauf der Arbeit genannt werden, zweimal durchlaufen werden.

Zur Rekonstruktion von 3D-Geraden werden die korrespondieren 2D-Geraden in *drei* Ansichten benötigt. Da nur zwei baugleiche Kameras zur Verfügung stehen, erhalten wir die drei Ansichten durch eine Stereoanordnung der beiden Kameras und einer zusätzlichen Aufnahme, die sich durch Manipulation des Kamerasystems in eine zur Stereoanordnung feste Position ergibt. Danach erfolgt die Detektion der Bildgeraden durch die orientierte Hough-Transformation. Die Tripel der korrespondierenden Geraden in den drei Ansichten werden durch ein punktweises Matchingverfahren ermittelt, welches die epipolare Bedingung ausnutzt und sich auf ein Template-Matching- und ein Votierungsverfahren stützt. Anschließend erfolgt die Rekonstruktion der 3D-Geraden aus den Tripeln von Bildgeraden. Aus diesen Geraden werden dann in hierarchischer Weise Objektmodelle zusammengesetzt. Dabei findet eine Abstraktion der primitiven 3D-Geraden auf die höher strukturierten Objektmodelle statt, wobei zwecks Verifikation der extrahierten Strukturen auf die Bilddaten der Ansichten zurückgegriffen wird. Abbildung 1.4 stellt die einzelnen Ebenen dieser Hierarchie graphisch dar. Die Menge der Geraden wird in einen die Szene beschreibenden Graphen überführt. Die Kanten dieses Graphen drücken die topologische Beziehung der 3D-Geraden aus. Beispielsweise sind sich schneidende Geraden Kandidaten für Ecken der 3D-Objektoberflächen in der Szene. Selektionskriterien dienen der Reduktion des Graphen.

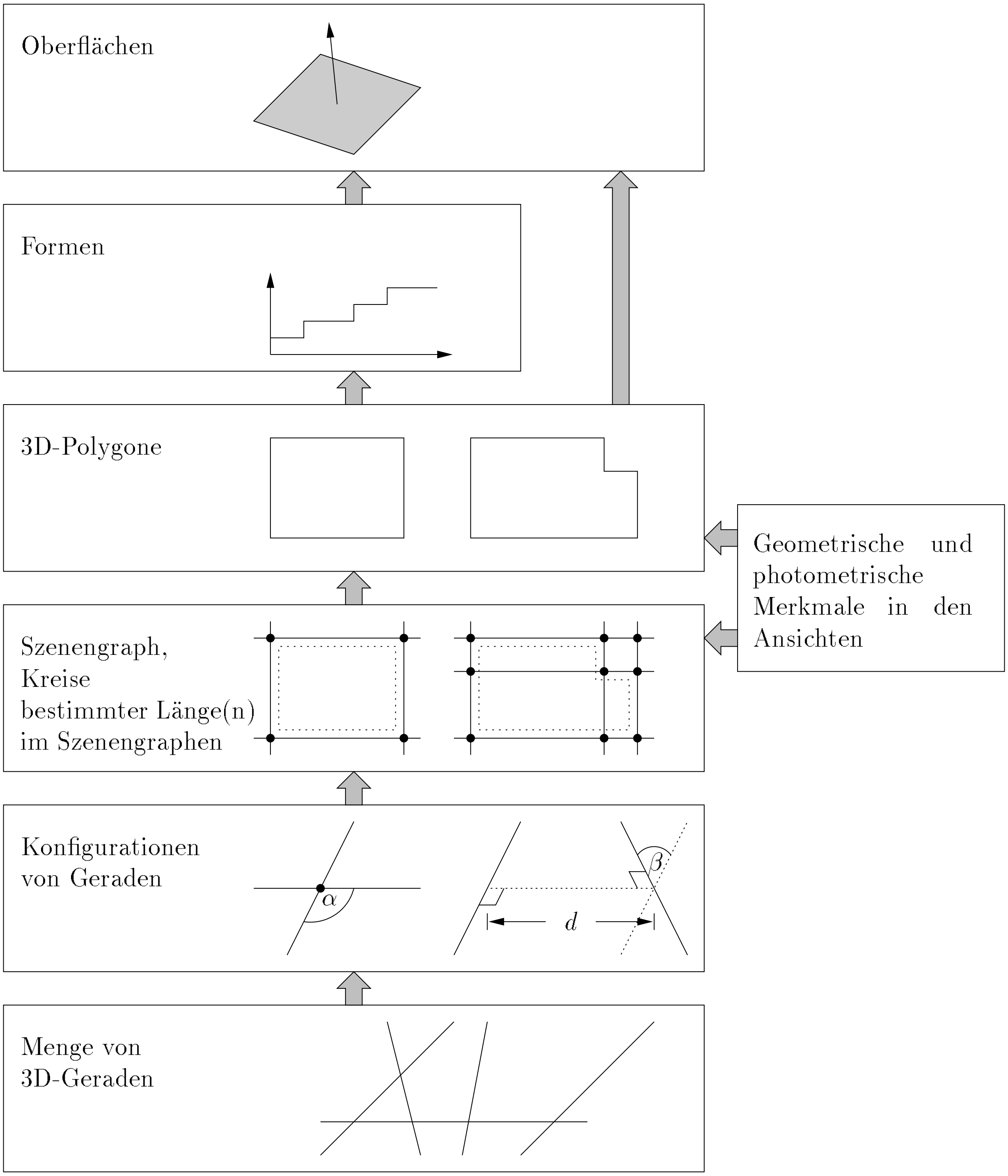


Abbildung 1.4: Abstraktionsschema

Aus den Kreisen bestimmter Länge im Szenengraphen werden die Polygonzüge berechnet. Um Polygone, die keinen Objektberandungen entsprechen, auszuschließen, müssen auf den Bilddaten der Ansichten basierende Kriterien erfüllt sein. Aus den verbleibenden Polygonzügen wird dann eine für die Identifikation der Objektoberflächen benötigte Formrepräsentation berechnet. Diese ist Teil des endgültigen Oberflächenmodells, das insbesondere die zur Lokalisation des Kamerasystems benötigten 3D-Daten enthält. Zwecks Identifikation der in der Szene lokalisierten Oberflächenmodelle findet ein Vergleich mit den in der Datenbank gespeicherten Modellen statt, wodurch ein Zielobjekt eindeutig bestimmt wird.

Danach erfolgt die Manipulation des Kamerasystem durch den Roboterarm derart, daß sich das Zielobjekt in einer vordefinierten Position und Orientierung innerhalb des Kamerasystems nach der Bewegung befindet. Eine weitere, am Endeffektor des Roboterarmes montierte Kamera mit geeigneter Brennweite blickt nun auf die Oberfläche des Zielobjektes. Es folgt eine Bildaufnahme und die Anwendung der trainierten, neuronalen und objektspezifischen Operatoren zur Erkennung der „kleinen“ Objekte.

1.3 Einbettung der Arbeit in die Literatur

Bei der Literaturrecherche wurde kein System gefunden, welches mit dem in dieser Arbeit entwickelten System als Ganzes direkt vergleichbar ist. Daher soll eine Einordnung der wesentlichen Teilverfahren erfolgen, außerdem werden die Quellen der aus der Literatur entnommenen Verfahren angegeben.

Eine Einführung in die Robotik gibt Craig [7] und liefert die für die Architektur des verwendeten Roboterarmes benötigten Transformationen.

Als Grundlage für die Kamerakalibrierung diente Faugeras [9]. Allerdings führte die dort beschriebene lineare Methode [9, Kap. 3.4.1.2] zur Schätzung der Projektionsmatrix $\tilde{\boldsymbol{P}}$ zu numerischen Instabilitäten, eine entsprechende Lösung dieses Problems findet sich in Kapitel 4.1. Die Hand-Auge-Kalibrierung besteht in der Lösung der Matrixgleichung der Form $\boldsymbol{AX} = \boldsymbol{XB}$, wobei die zu schätzende Matrix $\boldsymbol{X}$ die Rotation und Translation beschreibt, durch welche das Koordinatensystem der Hand und das Koordinatensystem des Auges oder Kamerasystems in räumlicher Beziehung stehen. Die Lösungsverfahren in [35, 33, 38] berechnen die Rotation und die Translation separat, in [8] wird ein ganzheitlicher Ansatz vorgestellt. In dieser Arbeit braucht aufgrund der Art und Weise, in der die Kamerakalibrierung erfolgt, nur der Translationsteil geschätzt zu werden.

Die geometrische, modellbasierte Objekterkennung besteht aus zwei Phasen: der perzeptuellen Organisation und der Verifikation. Als perzeptuelle Organisation wird das schrittweise Gruppieren von aus dem Bild extrahierten Merkmalen und Bildprimitiva zu höherstrukturierten, abstrakteren Objektrepräsentationen verstanden [31, 32, 43, 41]. Im Verlaufe der perzeptuellen Organisation wird in den genannten Arbeiten a priori-Wissen in Form von Invarianten, Quasi-Invarianten, Ähnlichkeiten, Symmetrien und Regularitäten eingesetzt, ebenso ist die Eignung der verwendeten Merkmale für die Objekterkennung bekannt. Die Verifikation besteht im Überprüfen von merkmalsbasierten, logischen Prädikaten (*hypothesis verification*) und dem Vergleich der gefundenen Objektrepräsentationen mit Modellen in einer Datenbank. Die Akquisition dieser Modelle erfolgt aus Bildern, in denen sich die Objekte durch einfache Bildverarbeitung vom Hintergrund separieren lassen. Durch den Vergleich mit den Modellen in der Datenbank lassen sich außerdem Objekte eindeutig identifizieren. Auch in dieser Arbeit erfolgt eine Verifikation und Identifikation mit einem Modell in der Datenbank.

Die Bedeutung topologischer Aspekte bei der Objekterkennung wird in der Arbeit von Rothwell [30] hervorgehoben. In dem dort vorgestellten System werden außerdem topologische Beziehungen mit geometrischen Informationen kombiniert.

Grundlegend für die erscheinungsbasierte Objekterkennung sind die Arbeiten von Murase, Nayar und Nene [23, 20, 24, 22, 25, 21]. Charakteristisch in diesen Verfahren ist eine low-level Vorverarbeitung der Trainingsansichten von Objekten in verschiedenen Orientierungen, eine anschließende Normierung, mit der die Beleuchtungsunabhängigkeit des Objekterkenners erreicht wird, sowie die Repräsentation der Objektmannigfaltigkeiten in den durch Hauptkomponentenanalyse erzeugten Eigenräumen. Die Objekterkennung besteht darin, eine neue Objektansicht in derselben Weise vorzuverarbeiten, in den Eigenraum zu projizieren und der bezüglich des euklidischen Abstands nächsten Objektmannigfaltigkeit zuzuordnen. Dieses erfolgt durch die Modellierung der Mannigfaltigkeiten durch Splines und ein Punktsuchverfahren.

Ein experimenteller Vergleich von modellbasierten, geometrischen Verfahren mit dem erscheinungsbasierten Verfahren findet sich in [19]. In diesem Report wurden die Testergebnisse für Ansichten von Objekten in verschiedenen Orientierungen mit unterschiedlichen Graden an Okklusionen durch ein weiteres Objekt dokumentiert. Die Objekte befinden sich jedoch alle auf sehr homogenem Hintergrund, im Unterschied zu den Szenen mit Platinen in dieser Arbeit.

Ein auf Segmenten als geometrische Bildprimitiva basierendes System wird von Zhang und Faugeras beschrieben [42].

Die Systeme zur Objekterkennung in den aufgeführten Quellen gehen von üblichen Beleuchtungsverhältnissen aus. Tan und Freeman [34] dagegen setzen in ihren Verfahren Streifenmuster ein, die durch eine Lichtquelle in die Szene projiziert werden.

Die Bildprimitiva, auf denen sich die geometrische Objekterkennung dieser Arbeit stützt, sind Geraden. Diese werden durch Hough-Transformation [13] detektiert. Eine Übersicht über verschiedene Ausprägungen und Anwendungen der Hough-Transformation gibt Leavers [16]. Als Standardwerke, in denen die Hough-Transformation beschrieben werden, sind das Buch von Gonzalez [11] und Jähne [14] zu erwähnen.

Grundlegende, geometrische Verfahren für das Stereomatching und die Rekonstruktion von Geraden werden in dem Buch von Faugeras [9] beschrieben.

Das für die Verifikation und Identifikation der Objekte verwendete Metrik auf Formen sowie der effiziente Algorithmus sind Arkin et al. [2] entnommen.

Die erscheinungsbasierte Objekterkennung basiert im Unterschied zu den Ansätzen von Murase und Nayar nicht auf der Modellierung der Objektmannigfaltigkeiten in den Eigenräumen durch Splines und dem in [24] veröffentlichten Algorithmus zur Punktsuche, sondern erfolgt durch RBF-Netze [28, 10].

1.4 Aufbau der Arbeit

Kapitel 2 skizziert den Versuchsaufbau und beschreibt die wesentlichen Hardwarekomponenten. Es werden die für die Verfahren relevanten Daten der Kameras angegeben und die charakteristischen Eigenschaften des Roboters und dessen Peripherie zusammengefaßt.

Kapitel 3 führt in die Beschreibung von Beziehungen zwischen Koordinatensystemen und die damit verbundenen Koordinatentransformationen von Punkten durch lineare Abbildungen (Matrizen) ein. Diese Beziehungen bzw. Transformationen beschreiben nicht nur den Zustand des Roboters, sondern sind auch für die Koordination zwischen der Hand des Roboters und dem System der Kameras von Bedeutung. Dieses Kapitel bildet die Grundlage für die Hand-Auge-Kalibrierung und die Berechnung der neuen Position und Orientierung der Roboterhand nach Erkennung der Objektoberfläche.

Gegenstand von Kapitel 4 sind zwei Kalibrierungen. Die Kamerakalibrierung dient der Schätzung der für die Rekonstruktion von Punkten und Geraden benötigten Projektionsmatrizen. Dabei erfolgt eine knappe Beschreibung der Geometrie an einer Kamera. Die Hand-Auge-Kalibrierung besteht im allgemeinen in der Schätzung der Orientierung und Translation zwischen dem Koordinatensystem der Hand und des Kamerasystems. Für das in dieser Arbeit entwickelte System genügt die Schätzung des Translationsteils. Die Beziehung zwischen beiden Koordinatensystemen wird für die gezielte Bewegung des Kamerasystems relativ zum erkannten Objekt benötigt.

Kapitel 5 beschreibt das Verfahren zur Lokalisation und Identifikation von polygonalen, planaren Objektoberflächen. Es wird zunächst das auf Bildgeraden basierende Verfahren motiviert. Die Detektion der Bildgeraden erfolgt durch die schnelle, orientierungsselektive Hough-Transformation, wobei der Sobeloperator zur Kantendetektion eingesetzt wird. Danach wird auf das Verfahren zur Bestimmung der Maxima eingegangen, sowie auf die Probleme bezüglich ihrer Bestimmung. Es werden einige Strategien zur Verbesserung der Detektion der Bildgeraden beschrieben sowie die Erweiterung des Votierungsalgorithmus in Hinblick auf das Stereomatching von Geraden. Danach werden die punktweisen Verfahren zur Berechnung von Korrespondenzen von Geraden in binokularen und trinokularen Ansichten vorgestellt. Das Matching erfolgt unter Ausnutzung der epipolaren Bedingung und durch das Einbeziehen eines Template-Matchings zur Auflösung von Mehrdeutigkeiten und eines Votierungsverfahrens zur Steigerung der Robustheit in realen Szenen. Das Verfahren für zwei Ansichten wird zu einem Verfahren auf drei Ansichten erweitert, um beliebige Geraden rekonstruieren zu können. Den Matchingverfahren schließt sich die Rekonstruktion der korrespondierenden Geraden an. Danach wird die 3D-Szene der rekonstruierten Geraden durch einen ungerichteten Graphen repräsentiert. Auf der Lage der Geraden basierende Kriterien führen zur Reduktion des Graphen. Durch Kreissuche im Graphen werden die Polygone sowohl im 3D-Kamerakoordinatensystem als auch in den korrespondierenden Ansichten berechnet. Die Polygone werden rekonstruiert, im Laufe der weiteren Informationsverarbeitung findet eine Abstraktion auf 3D-Oberflächenmodelle statt. Es werden das Oberflächenmodell sowie das Ähnlichkeitsmaß zum Vergleich zweier Modelle beschrieben.

Kapitel 6 beinhaltet die Berechnung der neuen Manipulatorposition und -orientierung zur Bewegung des Kamerasystems in eine gewünschte Position und Orientierung zur erkannten Oberfläche.

Kapitel 7 beschreibt die Testergebnisse zu den wichtigsten der in den bisherigen Kapiteln vorgestellten Verfahren bzw. des gesamten Systems zur Ausrichtung des Kamerasystems relativ zum erkannten Objekt. Es werden die Fehler der Kamerakalibrierung, der Hand-Auge-Kalibrierung, der Rekonstruktion von Punkten und Geraden bestimmt. Dieses erfolgt auch innerhalb eines Wahrnehmungs-Aktions-Zyklus. Die Leistung und Grenzen des Verfahrens zur Oberflächenerkennung werden anhand einiger Beispielszenen gezeigt.

Kapitel 8 beschreibt das erscheinungsbasierte, neuronale Verfahren zur Erkennung der Elektronikbauteile auf der Platine, zu dem das Kamerasystem ausgerichtet wurde. Es wird kurz erläutert, unter welchen Voraussetzungen die Objektansichten, mit denen trainiert wird, aufgenommen wurden. Als Vorverarbeitung im Sinne einer Skalierung der Ansichten und Merkmalsextraktion werden die Gauß- und die Laplacepyramide knapp beschrieben. Es schließt sich das objektspezifische Normierungsverfahren an, durch welches die Beleuchtungsinvarianz der Erkennung auch bei inhomogenem Hintergrund erreicht werden soll. Danach erfolgt die Beschreibung der Hauptkomponentenanalyse. Durch sie werden die objektabhängigen, niedrigdimensionalen Eigenräume erzeugt, in denen die Objekterkennung letztlich stattfindet. Es wird die Wahl der Anzahl von Ansichten und Hauptkomponenten diskutiert. Anschließend wird kurz in RBF-Netze eingeführt, sowie die Verfahren zur Bestimmung der Zentren und der Breiten der Basisfunktionen beschrieben. Wesentlich ist auch die Wahl der Trainingsmuster für die verschiedenen, zu testenden Trainingsmöglichkeiten. Aus den einzelnen Verarbeitungskomponenten dieses Kapitels wird abschließend ein nichtlinearer, verschiebungsinvarianter Operator zur Erkennung eines Objektes konstruiert.

In Kapitel 9 werden die Testergebnisse für den Operator vorgestellt. Zunächst werden die Netze auf unterschiedliche Art und Weise trainiert. Es folgt der Test der trainierten Netze in Bezug auf die Trainingsdaten. Danach werden die Klassifikationsleistungen der Operatoren in der Umgebung der trainierten Mittelpunkte der Trainingsansichten sowie auf Bildern realer Szenen mit Platinen untersucht.

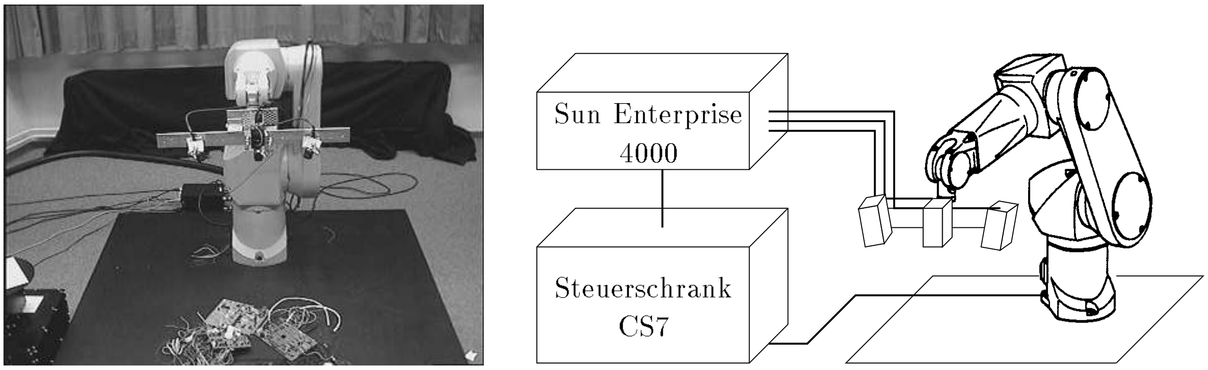
Kapitel 10 faßt die Ergebnisse dieser Arbeit zusammen und schließt mit einem Ausblick.

Kapitel 2

Versuchsaufbau und Hardware

2.1 Versuchsaufbau

Abbildung 2.1 zeigt den Aufbau der technischen Komponenten. Im Zentrum steht der Roboterarm, an dem 3 Kameras starr befestigt sind. Die zwei äußeren Kameras dienen der Aufnahme der Bilddaten, die zur Erkennung und Lokalisation der polygonalen Objektoberfläche benötigt werden. Die innere Kamera wird zur Erkennung von Objekten auf der erkannten Oberfläche eingesetzt. Die Kameras sind mit den Framegrabber-Karten in der Workstation, einer SUN Enterprise 4000, verbunden. Die Workstation besitzt derzeit 4 Prozessoren mit einer Taktfrequenz von jeweils 167 MHz. Dadurch wird eine parallele Durchführung von Berechnungen ermöglicht. Die Realisierung von Nebenläufigkeiten erfolgte in dieser Arbeit in Form von Threads und erfordert grundsätzlich den Entwurf paralleler Algorithmen und deren Implementation. Der Roboter ist über eine interne Schnittstelle mit dem Steuerschrank verbunden, der wiederum über eine serielle Schnittstelle mit der Sun Enterprise 4000 verbunden ist. Abbildung 2.1(b) skizziert den technischen Aufbau.



(a) Bild des Versuchsaufbaus

(b) Komponenten des Systems

Abbildung 2.1: Der Versuchsaufbau

2.2 Hardware

In diesem Kapitel werden die wichtigsten Hardwarekomponenten – Kameras und Roboterarm – beschrieben.

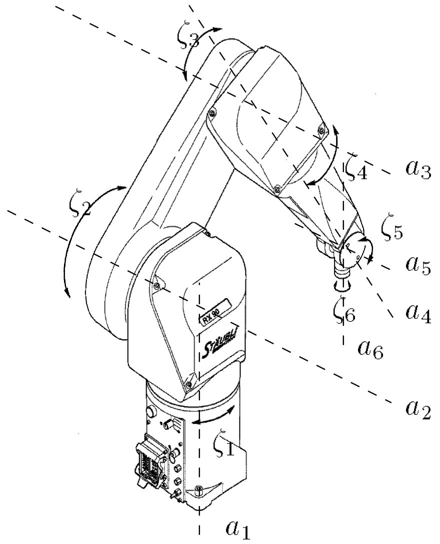
2.2.1 Die Kameras

Es werden drei Grauwert-CCD-Kameras verwendet. Diese sind starr am Endeffektor des Roboters montiert. Die Bilder der beiden Kameras, die zur Erkennung der Oberfläche eingesetzt werden, haben eine maximale Größe von 640×480 Pixel. Die verwendeten Bilder haben eine Größe von 480×480 Pixel. Die Brennweite der Objektive dieser Kameras beträgt 12 mm. Die Bilder der Kamera, die zur Erkennung der kleinen Objekte eingesetzt werden, haben eine maximale Größe von 678×576 Pixel. Die Größe der verwendeten Bilder beträgt 512×512 Pixel. Die Brennweite des Objektivs beträgt hier 25 mm.

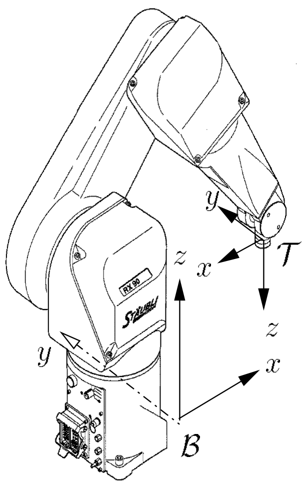
2.2.2 Der Roboter

Der Roboterarm RX90 der Firma Stäubli ist ein Knickarmroboter zur Boden- und Deckenmontage und gehört zur Klasse der anthropomorphen Roboter. Der Endeffektor des Armes kann im Arbeitsbereich beliebig positioniert und orientiert werden. Die Gliedmaßen des Roboterarmes sind durch 6 Gelenke (*joints*) verbunden, die eine Drehung zweier verbundener Gliedmaßen um eine Rotationsachse erlauben, wie Abbildung 2.2(a) verdeutlicht. Die den Achsen a_1, a_4 und a_6 entsprechenden Verbindungen werden als *cylindrical joint*, die übrigen Verbindungen werden als *revolute joint* bezeichnet [7]. Ein Winkel zwischen zwei benachbarten Gliedmaßen bildet eine Stellgröße, so daß sich insgesamt die den Roboter charakterisierenden 6 Freiheitsgrade ergeben. Die Stellgrößen bestimmen eindeutig die Position und Orientierung des Endeffektors; zur Festlegung der Lage des Endeffektors ist jedoch die Beschreibung durch kartesische Koordinatensysteme und Orientierungswinkel geeigneter. Die vorhandene Peripherie des Roboters löst die Probleme der inversen Kinematik und inversen Dynamik, d. h. die Berechnung der Stellgrößen zu vorgegebenen Koordinaten und Winkeln unter den gegebenen physikalischen Zwängen.

Das äußere Bezugssystem ist durch das Welt- oder Basiskoordinatensystem festgelegt. Zur Beschreibung von Position und Orientierung des Endeffektors ist das Toolkoordinatensystem definiert, dessen Ursprung sich auf der Rotationsachse der letzten Gliedmaße befindet. Die Position dieses Ursprungs ist der Vektor der Koordinaten im Weltkoordinatensystem. Die Orientierung des Toolkoordinatensystems relativ zum Weltkoordinatensystem ist in speziellen Eulerschen Winkeln definiert, auf die im nächsten Kapitel eingegangen wird.



(a) Die Rotationsachsen



(b) Basis- und Toolkoordinatensystem

Abbildung 2.2: Der Roboterarm RX90

Kapitel 3

Grundlagen

3.1 Koordinatensysteme und Transformationen

In dieser Arbeit spielen verschiedene kartesische Koordinatensysteme eine Rolle: das Basiskoordinatensystem des Roboters, das Toolkoordinatensystem des Roboters und des weiteren das Koordinatensystem des Kamerasystems und das Koordinatensystem der Objektoberfläche. Durch diese Koordinatensysteme wird sowohl die Position als auch die Orientierung in den Systemen untereinander beschreibbar. Beispielsweise legt die Lage des Toolkoordinatensystems im Basiskoordinatensystem die Position und die Orientierung der Roboterhand eindeutig fest. Betrachtet man die Koordinaten eines Punktes im Basiskoordinatensystem, so stellt sich die Frage, wie die Koordinaten bezüglich des Toolkoordinatensystems lauten. Von Interesse sind also sowohl die Beschreibung der relativen Lage von Koordinatensystemen als auch die induzierten Koordinatentransformationen.

Die Beziehung wird durch 4×4–Matrizen beschrieben, die sich in einen Rotationsanteil und einen Translationsanteil zerlegen lassen. Wie in Abb. 3.1 dargestellt, seien zwei Koordinatensysteme $\mathcal{X}$ und $\mathcal{Y}$ gegeben. Der Vektor $\widehat{\boldsymbol{t}}$ beschreibt die Lage des Ursprungs von $\mathcal{Y}$ in $\mathcal{X}$, und die orthonormale Matrix $\widehat{\boldsymbol{R}}$ legt die Orientierung von $\mathcal{Y}$ bezüglich $\mathcal{X}$ fest:

$$\widehat{\boldsymbol{A}} = \begin{pmatrix} \widehat{\boldsymbol{R}} & \widehat{\boldsymbol{t}} \\ \mathbf{0} & 1 \end{pmatrix} \tag{3.1}$$

Durch $\widehat{\boldsymbol{A}}$ sind auch zwei lineare Abbildungen gegeben, welche die Koordinaten eines Vektors $\boldsymbol{p}$ von einem System in das andere transformieren:

$$\begin{aligned} &\boldsymbol{\Phi} : \mathcal{Y} \to \mathcal{X}; \\ &\begin{pmatrix} {}^{\mathcal{Y}}\boldsymbol{p} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} {}^{\mathcal{X}}\boldsymbol{p} \\ 1 \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{R}} & \widehat{\boldsymbol{t}} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} {}^{\mathcal{Y}}\boldsymbol{p} \\ 1 \end{pmatrix} \end{aligned} \tag{3.2}$$

und die zu $\boldsymbol{\Phi}$ inverse Abbildung

$$\begin{aligned} &\boldsymbol{\Phi}^{-1} : \mathcal{X} \to \mathcal{Y}; \\ &\begin{pmatrix} {}^{\mathcal{X}}\boldsymbol{p} \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} {}^{\mathcal{Y}}\boldsymbol{p} \\ 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} {}^{\mathcal{X}}\boldsymbol{p} \\ 1 \end{pmatrix} \end{aligned} \tag{3.3}$$

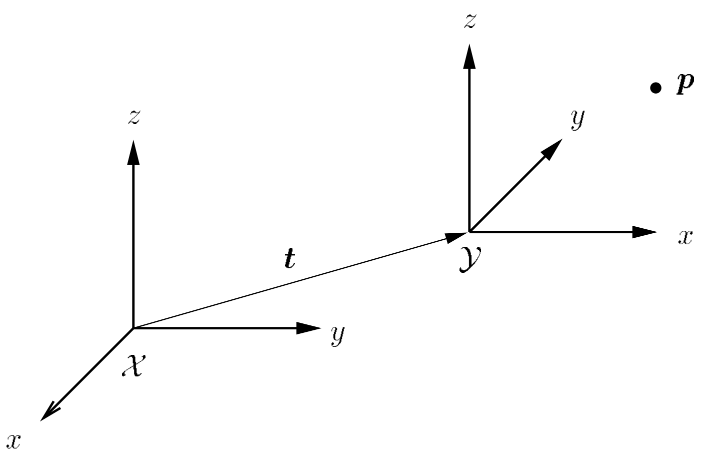


Abbildung 3.1: Beziehung zweier Koordinatensysteme

mit $\boldsymbol{R} := \widehat{\boldsymbol{R}}^T$ und $\boldsymbol{t} := -\widehat{\boldsymbol{R}}^T\widehat{\boldsymbol{t}}$. Im weiteren Verlauf werden die Matrix in (3.1), welche die Lage von $\mathcal{Y}$ in $\mathcal{X}$ beschreibt und die Matrix in (3.3), welche die Koordinatentransformation von $\mathcal{X}$ nach $\mathcal{Y}$ beschreibt, verwendet. Um eine durch die Inversion entstehende Verwirrung zu vermeiden, werden die Matrizen und Vektoren, welche die Lage beschreiben, mit dem Symbol $\widehat{\ }$ gekennzeichnet. Matrizen ohne Kennzeichnung repräsentieren wie üblich Abbildungen.

3.2 Das Robotermodell

Der Zustand des Endeffektors des RX90 ist durch die Lage und die Orientierung des Toolkoordinatensystems $\mathcal{T}$ relativ zum Basiskoordinatensystem $\mathcal{B}$ festgelegt. Der Ursprung des Toolkoordinatensystems ist durch 3 Weltkoordinaten ${}^{\mathcal{B}}x$, ${}^{\mathcal{B}}y$ und ${}^{\mathcal{B}}z$ festgelegt, die Orientierung durch drei sog. ZYZ-Euler-Winkel α, β und γ, die technisch auch als *yaw*, *pitch* und *roll* bezeichnet werden.

Definition 1 (ZYZ-Euler-Winkel) *Es seien drei Winkel α, β und γ gegeben. Die Orientierung des Toolkoordinatensystems $\mathcal{T}$ bezüglich $\mathcal{B}$ ergibt sich schrittweise aus einem Toolkoordinatensystem, das zunächst gleichorientiert mit $\mathcal{B}$ ist und mit $\mathcal{T}_0$ bezeichnet sei. Das System $\mathcal{T}_0$ werde zunächst um seine Z-Achse um den Winkel α gedreht, das neue System sei mit $\mathcal{T}_1$ bezeichnet. Danach erfolgt eine Drehung um die Y-Achse des Systems $\mathcal{T}_1$ um β. Das entstandene System $\mathcal{T}_2$ wird zuletzt um seine Z-Achse um den Winkel γ gedreht, so daß sich insgesamt das Koordinatensystem $\mathcal{T}$ ergibt.*

Der Zustand des Roboters ist somit durch das Tupel

$$({}^{\mathcal{B}}x, {}^{\mathcal{B}}y, {}^{\mathcal{B}}z, \alpha, \beta, \gamma) \tag{3.4}$$

vollständig beschrieben.

Um die Kinematik des Roboterarmes im Rahmen der linearen Algebra beschreiben zu können, werden zwei Funktionen benötigt: eine, die zu gegebenen ZYZ-Euler-Winkeln die äquivalente Orientierungsmatrix liefert, und die Funktion, die eine Orientierungsmatrix auf die entsprechenden ZYZ-Euler-Winkel abbildet [7]:

Bemerkung 1 *Seien α, β und γ die Winkel einer ZYZ-Euler-Winkel-Transformation. Dann ist die Orientierung des Toolkoordinatensystems $\mathcal{T}$ bezüglich des Basiskoordinatensystems $\mathcal{B}$ durch die Matrix*

$$\widehat{\boldsymbol{R}}(\alpha,\beta,\gamma) = \begin{pmatrix} \cos(\alpha)\cos(\beta)\cos(\gamma)-\sin(\alpha)\sin(\gamma) & -\cos(\alpha)\cos(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta) \\ \sin(\alpha)\cos(\beta)\cos(\gamma)+\cos(\alpha)\sin(\gamma) & -\sin(\alpha)\cos(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta) \\ -\sin(\beta)\cos(\gamma) & \sin(\beta)\sin(\gamma) & \cos(\beta) \end{pmatrix} \tag{3.5}$$

gegeben.

Die umgekehrte Richtung liefert die folgende Bemerkung:

Bemerkung 2 *Ist*

$$\widehat{\boldsymbol{R}} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \in SO(3) \tag{3.6}$$

eine Matrix, welche die Orientierung von $\mathcal{T}$ in $\mathcal{B}$ beschreibt, so sind die äquivalenten ZYZ-Euler-Winkel durch

$$\beta = \operatorname{atan2}(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}) \tag{3.7a}$$

$$\alpha = \begin{cases} 0 & ,\textit{falls } \sin(\beta) = 0 \\ \operatorname{atan2}(\frac{r_{23}}{\sin\beta}, \frac{r_{13}}{\sin\beta}) & ,\textit{sonst} \end{cases} \tag{3.7b}$$

$$\gamma = \begin{cases} \operatorname{atan2}(-r_{12}, r_{11}) & ,\textit{falls } \sin(\beta) = 0 \wedge \beta = 0 \\ \operatorname{atan2}(r_{12}, -r_{11}) & ,\textit{falls } \sin(\beta) = 0 \wedge \beta = \pi \\ \operatorname{atan2}(\frac{r_{32}}{\sin\beta}, -\frac{r_{31}}{\sin\beta}) & ,\textit{sonst} \end{cases} \tag{3.7c}$$

gegeben.

Es sei noch angemerkt, daß die beiden Funktionen nicht invers zueinander sind, da die Funktion in Bemerkung 1, die zu gegebenen ZYZ-Euler-Winkeln die entsprechende Orientierungsmatrix liefert, nicht injektiv und somit nicht invertierbar ist: beispielsweise beschreiben $(\frac{\pi}{2}, 0, \frac{\pi}{2})$ und $(0, 0, \pi)$ die gleiche Orientierung.

Kapitel 4

Kalibrierung

Gegenstand dieses Kapitels sind zwei Arten von Kalibrierungen, die in diesem Hand-Auge-System benötigt werden: die Kamerakalibrierung und die Hand-Auge-Kalibrierung.

Die Kalibrierung einer Kamera besteht in der Schätzung der Abbildung von einem externen 3D-Koordinatensystem auf die Bildebene dieser Kamera. Diese Abbildung ist eine lineare Projektion, somit besteht die Kalibrierung einer Kamera in der Schätzung einer Projektionsmatrix. Voraussetzung ist, daß die Kameras durch das Lochkameramodell beschreibbar sind, was insbesondere durch die Wahl geeigneter Objektive gegeben ist. Die Komponenten dieser Projektionsmatrix beinhalten die intrinsischen und extrinsischen Parameter der Kamera. Die intrinsischen Parameter sind abhängig von der inneren Geometrie der Kamera. Diese wird durch die Brennweite und die Abmessungen und Lage der CCD-Sensoren bestimmt. Die extrinsischen Parameter beschreiben die vom äußeren Koordinatensystem abhängigen Parameter, in dem die Kamera positioniert und orientiert ist. Wird nur die Projektionsmatrix geschätzt, so wird dies als implizite Kalibrierung bezeichnet, werden außerdem die intrinsischen und die extrinsischen Parameter berechnet, spricht man von expliziter Kalibrierung [39]. Welche Art der Kamerakalibrierung benötigt wird, hängt von der Problemstellung ab. Für das Stereomatching in Kapitel 5.2 und für die Rekonstruktion von Geraden und Punkten (Kap. 5.3 u. 5.6.3) werden lediglich die Projektionsmatrizen der Kameras benötigt.

Das äußere 3D-Koordinatensystem ist bezüglich der Anordnung der Kameras fest, es soll als Kamerakoordinatensystem bezeichnet werden. Da der Roboter bewegt wird, wird auch das Kamerakoordinatensystem mitbewegt. Dabei bleiben das Toolkoordinatensystem und das Kamerakoordinatensystem bei der Bewegung in fester Position und Orientierung zueinander (siehe Abb. 4.1).

Für die Rekonstruktion von Geraden müssen mindestens drei Ansichten vorliegen. Da jedoch nur 2 baugleiche Kameras zur Verfügung stehen, wird das Stereokamerasystem mittels einer festen, relativen Bewegung des Tools in eine für eine dritte Ansicht geeignete Lage gebracht. Es müssen also drei Kamerakalibrierungen durchgeführt werden: zwei für die Kameras des Stereokamerasystems und eine für die dritte Kamera, die man durch die feste, relative Bewegung des Tools erhält.

Um die Oberfläche von Objekten durch eine Kamera visuell untersuchen zu können, muß

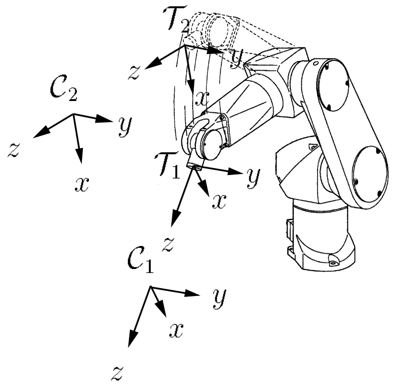


Abbildung 4.1: Wird das Tool von der durch das Toolkoordinatensystem $\mathcal{T}_1$ beschriebenen Lage nach $\mathcal{T}_2$ bewegt, so wird das Kamerakoordinatensystem von $\mathcal{C}_1$ nach $\mathcal{C}_2$ bewegt. Kamerasystem und Tool bleiben in den beiden Zuständen in fester Position und Orientierung zueinander.

diese in eine vorgegebene Orientierung zur Oberfläche des Objektes bewegt werden. Die Koordinaten des Objektes sind im Kamerakoordinatensystem, das durch die Kamerakalibrierung festgelegt ist. Die Koordinaten des Objektes ergeben sich aus der Rekonstruktion. Die Bewegung des Kamerakoordinatensystems in eine andere Position und Orientierung wird durch eine Veränderung der Position und Orientierung des Toolkoordinatensystems bewirkt. Das Kamerasystem und das Toolkoordinatensystem stehen, wie bereits erwähnt wurde, in fester Beziehung zueinander. Nach Kapitel 3.1 ist diese Beziehung gleichbedeutend mit einer Koordinatentransformation. Um die Koordinatentransformation der Toolkoordinatensysteme berechnen zu können (Kapitel 6), muß die Beziehung zwischen Toolkoordinatensystem und Kamerakoordinatensystem bekannt sein. Das Ziel der Hand-Auge-Kalibrierung ist also die Schätzung der Abbildung zwischen dem Toolkoordinatensystem des Roboters und dem Kamerakoordinatensystem, das sich aus der Kalibrierung der festen Stereokameras ergibt.

4.1 Kamerakalibrierung

Sowohl zum Stereomatching als auch zur Rekonstruktion von Punkten und Geraden werden die Kalibrierungsmatrizen der Kameras benötigt, welche die Abbildung vom Koordinatensystem des Kamerasystems auf die Bildebenen der Kameras beschreiben. Die Linearität dieser Abbildung ergibt sich durch die Verwendung projektiver anstatt euklidischer Koordinaten [9]. Voraussetzung ist, daß die Kameras durch das Lochkameramodell beschreibbar sind, was durch die Wahl geeigneter Objektive gegeben ist.

Die Projektionsmatrix $\tilde{\boldsymbol{P}}$ beschreibt die Abbildung des projektiven Raumes $\mathcal{P}^3$ auf die projektive Ebene $\mathcal{P}^2$:

$$\tilde{\boldsymbol{P}} : \mathcal{P}^3 \to \mathcal{P}^2; \tilde{\boldsymbol{M}} \mapsto \tilde{\boldsymbol{m}} := \tilde{\boldsymbol{P}}\tilde{\boldsymbol{M}} \tag{4.1}$$

Dabei sind $\tilde{\boldsymbol{M}} = (x, y, z, t)$ und $\tilde{\boldsymbol{m}} = (u, v, s)$ die projektiven Koordinaten des projektiven Raumes $\mathcal{P}^3$ bzw. der projektiven Ebene $\mathcal{P}^2$.

Um Vektoren mit projektiven Koordinaten von Vektoren mit euklidischen Koordinaten unterscheiden zu können, werden diese durch das Symbol ˜ gekennzeichnet, ebenso projektive Matrizen.

Ziel der Kalibrierung ist die Schätzung der 3×4-Matrix $\tilde{\boldsymbol{P}}$. Die Projektionsmatrix $\tilde{\boldsymbol{P}}$ läßt sich nach Faugeras [9] in der Form

$$\tilde{\boldsymbol{P}} = \begin{pmatrix} \tilde{\boldsymbol{Q}}_1{}^T \\ \tilde{\boldsymbol{Q}}_2{}^T \\ \tilde{\boldsymbol{Q}}_3{}^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}_1{}^T & q_{14} \\ \boldsymbol{q}_2{}^T & q_{24} \\ \boldsymbol{q}_3{}^T & q_{34} \end{pmatrix} = \begin{pmatrix} \boldsymbol{P} & \tilde{\boldsymbol{p}} \end{pmatrix} \tag{4.2}$$

darstellen. Der Vektor $\tilde{\boldsymbol{Q}}_1$ beschreibt die Ebene im projektiven Raum $\mathcal{P}^3$, die auf die durch $u = 0$ bestimmte Gerade in der projektiven Ebene $\mathcal{P}^2$ projiziert werden. Ebenso korrespondiert $\tilde{\boldsymbol{Q}}_2$ mit der Geraden $(u, 0, s)$ und $\tilde{\boldsymbol{Q}}_3$ mit der Geraden $(u, v, 0)$. Abbildung 4.2 dient der Veranschaulichung.

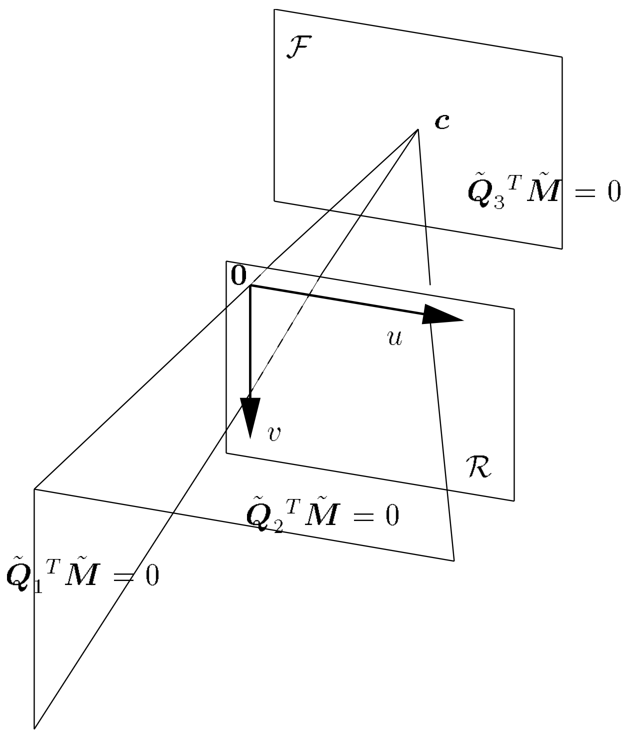


Abbildung 4.2: Die den Zeilenvektoren von $\tilde{\boldsymbol{P}}$ entsprechenden Ebenen (nach [9])

Das optische Zentrum der Kamera, dessen euklidische Koordinaten mit dem Vektor $\boldsymbol{c}$ bezeichnet seien, ist der Schnitt der 3 Ebenen:

$$\tilde{\boldsymbol{P}} \begin{pmatrix} \boldsymbol{c} \\ 1 \end{pmatrix} = \boldsymbol{0} \tag{4.3}$$

Mit $\tilde{\boldsymbol{P}} \begin{pmatrix} \boldsymbol{c} \\ 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{P} & \tilde{\boldsymbol{p}} \end{pmatrix} \begin{pmatrix} \boldsymbol{c} \\ 1 \end{pmatrix} = \boldsymbol{P}\boldsymbol{c} + \tilde{\boldsymbol{p}}$ folgt:

$$\boldsymbol{c} = -\boldsymbol{P}^{-1}\tilde{\boldsymbol{p}} \tag{4.4}$$

Die Matrix $\tilde{\boldsymbol{P}}$ wird gemäß Gleichung (4.2) in den Vektor

$$\boldsymbol{q} = (\boldsymbol{q}_1{}^T, q_{14}, \boldsymbol{q}_2{}^T, q_{24}, \boldsymbol{q}_3{}^T, q_{34}) \tag{4.5}$$

umnotiert. Für eine Messung

$$(\boldsymbol{M}_i, \boldsymbol{m}_i) = ((M_{i\,1}, M_{i\,2}, M_{i\,3})^T, (u_i, v_i)^T) \tag{4.6}$$

korrespondierender Welt- und Bildkoordinaten folgt aus $\tilde{\boldsymbol{m}} = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{M}}$:

$$\boldsymbol{q}_1{}^T\boldsymbol{M}_i - u_i\boldsymbol{q}_3{}^T\boldsymbol{M}_i + q_{14} - u_i q_{34} = 0 \tag{4.7a}$$

$$\boldsymbol{q}_2{}^T\boldsymbol{M}_i - v_i\boldsymbol{q}_3{}^T\boldsymbol{M}_i + q_{24} - v_i q_{34} = 0 \tag{4.7b}$$

Für n Messungen erhält man aus (4.7) mit der $2n \times 12$-Matrix

$$\boldsymbol{A} := \begin{pmatrix} M_{1\,1} & M_{1\,2} & M_{1\,3} & 1 & 0 & 0 & 0 & 0 & -u_1M_{1\,1} & -u_1M_{1\,2} & -u_1M_{1\,3} & -u_1 \\ 0 & 0 & 0 & 0 & M_{1\,1} & M_{1\,2} & M_{1\,3} & 1 & -v_1M_{1\,1} & -v_1M_{1\,2} & -v_1M_{1\,3} & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{n\,1} & M_{n\,2} & M_{n\,3} & 1 & 0 & 0 & 0 & 0 & -u_nM_{n\,1} & -u_nM_{n\,2} & -u_nM_{n\,3} & -u_n \\ 0 & 0 & 0 & 0 & M_{n\,1} & M_{n\,2} & M_{n\,3} & 1 & -v_nM_{n\,1} & -v_nM_{n\,2} & -v_nM_{n\,3} & -v_n \end{pmatrix} \tag{4.8}$$

das i. allg. überbestimmte Gleichungssystem

$$\boldsymbol{A}\boldsymbol{q} = \boldsymbol{0} \tag{4.9}$$

Um die triviale Lösung $\boldsymbol{q} = \boldsymbol{0}$ auszuschließen, wird in [9] die Bedingung $\|\boldsymbol{q}_3\|^2 = 1$ gefordert und die Berechnung des Lösungsvektors nach einem in [9, Anhang A] beschriebenen Verfahren vorgeschlagen. Dieses hat sich jedoch als numerisch instabil erwiesen. Daher fordern wir die Bedingung $q_{34} = 1$, wodurch sich Gleichung (4.9) in das folgende Gleichungssystem umformen läßt:

$$\underbrace{\begin{pmatrix} M_{11} & M_{12} & M_{13} & 1 & 0 & 0 & 0 & 0 & -u_1M_{11} & -u_1M_{12} & -u_1M_{13} \\ 0 & 0 & 0 & 0 & M_{11} & M_{12} & M_{13} & 1 & -v_1M_{11} & -v_1M_{12} & -v_1M_{13} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ M_{n1} & M_{n2} & M_{n3} & 1 & 0 & 0 & 0 & 0 & -u_1M_{n1} & -u_1M_{n2} & -u_1M_{n3} \\ 0 & 0 & 0 & 0 & M_{n1} & M_{n2} & M_{n3} & 1 & -v_1M_{n1} & -v_1M_{n2} & -v_1M_{n3} \end{pmatrix}}_{\boldsymbol{A}} \boldsymbol{q} = \underbrace{\begin{pmatrix} u_1 \\ v_1 \\ \cdots \\ u_n \\ v_n \end{pmatrix}}_{\boldsymbol{b}} \tag{4.10}$$

Die Lösung dieses „neuen“ Gleichungssystems $\boldsymbol{Aq} = \boldsymbol{b}$ im Sinne des kleinsten quadratischen Fehlers ergibt sich theoretisch mit der Pseudoinversen $\boldsymbol{A}^{+} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$ zu

$$\boldsymbol{q} = \boldsymbol{A}^{+}\boldsymbol{b} \tag{4.11}$$

Aus Gründen der numerischen Stabilität wird jedoch ein auf dem Householder-Verfahren basierendes, iteratives Verfahren aus [29] auf das Gleichungssystem

$$(\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{q} = \boldsymbol{A}^T\boldsymbol{b} \tag{4.12}$$

angewendet.

Zur praktischen Durchführung der Kalibrierung befindet sich das Stereokamerasystem in der in Abbildung 4.3 dargestellten räumlichen Beziehung zum Toolkoordinatensystem. Mit dem Tool des Roboters wird ein räumliches Gitter von Punkten abgefahren und jeweils das Bild des in Abbildung 4.4(a) dargestellten Kalibrierungsmusters aufgenommen, welches sich auf der Arbeitsfläche befindet. Durch das auf der Hough-Transformation basierende Verfahren zur Detektion von Geraden, dessen Beschreibung in Kapitel 5.1 erfolgt, erhält man pro Bild genau zwei Bildgeraden, deren Schnittpunkt dann subpixelgenau $\boldsymbol{m}_i$ definiert (siehe Abb. 4.4(b)). Die den Schnittpunkten $\boldsymbol{m}_i$ zugeordneten 3D-Punkte $\boldsymbol{M}_i$ werden derart gewählt, daß sie den angefahrenen Positionen entsprechend ein räumliches Gitter bilden und das Kamerakoordinatensystem $\mathcal{C}$ definieren, welches mit dem Toolkoordinatensystem $\mathcal{T}$ gleich orientiert ist (siehe Abb. 4.3). Diese Eigenschaft vereinfacht die anschließende Hand-Auge-Kalibrierung.
Die soeben beschriebene Kamerakalibrierung wird für die drei Kameras, d. h. die zwei Kameras der Stereoanordnung und die aus der relativen Bewegung resultierende „dritte“ Kamera, durchgeführt. Das Ergebnis sind die drei Projektionsmatrizen $\tilde{\boldsymbol{P}}_1$, $\tilde{\boldsymbol{P}}_2$ und $\tilde{\boldsymbol{P}}_3$.

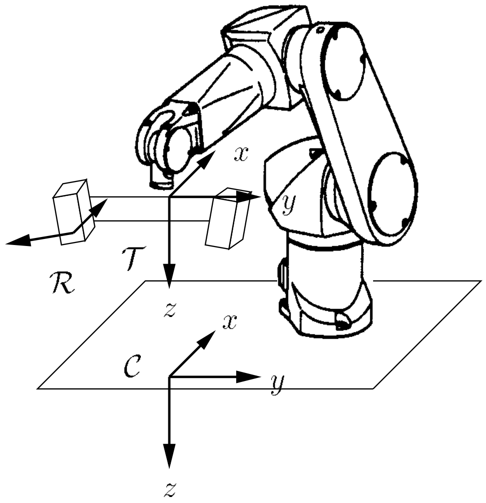
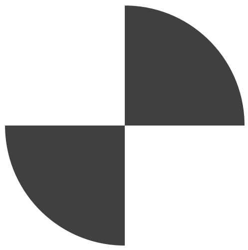
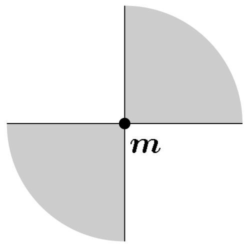


Abbildung 4.3: Zustand des Roboters zur Kamerakalibrierung



(a) Muster



(b) Schnittpunkt

Abbildung 4.4: Kalibrierungsmuster

4.2 Hand-Auge-Kalibrierung

Die Beziehung zwischen dem Toolkoordinatensystem $\mathcal{T}$ und dem Koordinatensystem $\mathcal{C}$ ist durch eine Koordinatentransformation festgelegt, die sich durch die 4×4-Matrix

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{R}_X & \boldsymbol{t}_X \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{4.13}$$

darstellen läßt[1]. Die 3×3-Matrix $\boldsymbol{R}_X$ ist der Rotationsanteil, der die Orientierung des Toolkoordinatensystems $\mathcal{T}$ im Kamerakoordinatensystem $\mathcal{C}$ angibt, $\boldsymbol{t}_X$ ist die Lage des Ursprungs des Toolkoordinatensystems im Kamerakoordinatensystem. Abbildung 4.5 stellt die Beziehung zwischen den Koordinatentransformationen graphisch dar. Es werden zwei Zustände des Tools betrachtet. Die entsprechenden Toolkoordinatensysteme seien mit $\mathcal{T}_1$ und $\mathcal{T}_2$ bezeichnet. Zu diesen beiden Koordinatensystemen gehören zwei Kamerakoordinatensysteme $\mathcal{C}_1$ und $\mathcal{C}_2$. Die Toolkoordinatensysteme $\mathcal{T}_1$ und $\mathcal{T}_2$ definieren eine Abbildung $T : \mathcal{T}_1 \to \mathcal{T}_2$, ebenso wird eine Abbildung $C : \mathcal{C}_1 \to \mathcal{C}_2$ definiert. Für die Abbildungen T, C und X gilt, wie Abb. 4.5 verdeutlicht,

$$\boldsymbol{C}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{T} \tag{4.14}$$

Analog zu Gleichung (4.13) lassen sich $\boldsymbol{C}$ und $\boldsymbol{T}$ in Rotationsanteil und Translationsanteil aufspalten. Die Matrix $\boldsymbol{R}_C$ ist dann die Orientierung von $\mathcal{C}_1$ im Koordinatensystem $\mathcal{C}_2$, $\boldsymbol{t}_C$ die Koordinaten des Ursprungs von $\mathcal{C}_1$ im Koordinatensystem $\mathcal{C}_2$. Entsprechend beschreibt $\boldsymbol{R}_T$ die Orientierung von $\mathcal{T}_1$ im Koordinatensystem $\mathcal{T}_2$, $\boldsymbol{t}_T$ die Koordinaten des Ursprungs von $\mathcal{T}_1$ im Toolkoordinatensystem $\mathcal{T}_2$. Durch Ausmultiplizieren erhält man aus Gleichung

[1] siehe Kapitel 3.1

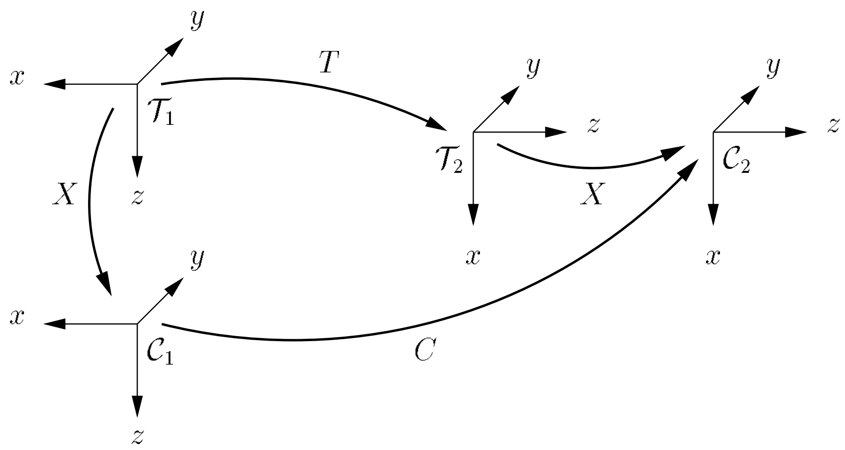


Abbildung 4.5: Diagramm der Abbildungen zwischen den Koordinatensystemen $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{C}_1$ und $\mathcal{C}_2$

(4.14):

$$\boldsymbol{R}_C \boldsymbol{R}_X = \boldsymbol{R}_X \boldsymbol{R}_T \tag{4.15}$$

$$(\boldsymbol{R}_C - \boldsymbol{I})\boldsymbol{t}_X = \boldsymbol{R}_X \boldsymbol{t}_T - \boldsymbol{t}_C \tag{4.16}$$

Durch die spezielle Art der Kamerakalibrierung gilt in dieser Arbeit der Spezialfall $\boldsymbol{R}_X = \boldsymbol{I}$. Somit vereinfacht sich Gleichung 4.15 zu

$$\boldsymbol{R}_C = \boldsymbol{R}_T \tag{4.17}$$

$$(\boldsymbol{R}_C - \boldsymbol{I})\boldsymbol{t}_X = \boldsymbol{t}_T - \boldsymbol{t}_C \tag{4.18}$$

Es braucht also nur noch der Translationsanteil $\boldsymbol{t}_X$ geschätzt werden. Die Matrix $\boldsymbol{R}_C - \boldsymbol{I}$ ist i. allg. singulär. Daher ist das Anfahren von mindestens drei Zuständen des Tools nötig und der Fehler wird minimal, wenn die z-Achsen der drei Toolkoordinatensysteme und somit auch der Kamerakoordinatensysteme orthogonal zueinander sind [38]. Das Ziel ist also die Gewinnung von $\boldsymbol{R}_{T_i}, \boldsymbol{t}_{T_i}, \boldsymbol{t}_{C_i}$, $i = 1, \ldots, n$ mit $n \geq 3$ aus korrespondierenden Roboterdaten und Kamerakoordinaten, so daß sich das zu lösende überbestimmte Gleichungssystem

$$\begin{pmatrix} \boldsymbol{R}_{T_1} - \boldsymbol{I} \\ \vdots \\ \boldsymbol{R}_{T_n} - \boldsymbol{I} \end{pmatrix} \boldsymbol{t}_X = \begin{pmatrix} \boldsymbol{t}_{T_1} - \boldsymbol{t}_{C_1} \\ \vdots \\ \boldsymbol{t}_{T_n} - \boldsymbol{t}_{C_n} \end{pmatrix} \tag{4.19}$$

ergibt.

Die Kalibrierung erfolgt unter Verwendung der zwei (äußeren) Kameras und des Kalibrierungsmusters in Abbildung 4.4. Für jedes Bild des Kalibrierungsmusters werden die Bildkoordinaten des Schnittpunktes der beiden Geraden bestimmt (Abb. 4.4). Aus den korrespondierenden Schnittpunkten beider Kameras werden mit dem in Kapitel 5.6.3 beschriebenen Verfahren die 3D-Koordinaten des Schnittpunktes bezüglich des momentanen Kamerakoordinatensystems rekonstruiert.
Zur Berechnung der $\boldsymbol{R}_{T_i}, \boldsymbol{t}_{T_i}, \boldsymbol{t}_{C_i}$, $i = 1, \ldots, n$ werden $n+1$ Punkte

$$^{\mathcal{C}_1}\boldsymbol{M}_1, \ldots, {}^{\mathcal{C}_{n+1}}\boldsymbol{M}_{n+1} \tag{4.20}$$

in den Kamerakoordinatensystemen $\mathcal{C}_1, \ldots, \mathcal{C}_{n+1}$ mit den korrespondierenden Roboterdaten

$$(x_1, y_1, z_1, \alpha_1, \beta_1, \gamma_1), \ldots, (x_{n+1}, y_{n+1}, z_{n+1}, \alpha_{n+1}, \beta_{n+1}, \gamma_{n+1}) \tag{4.21}$$

benötigt, welche die Lage der Toolkoordinatensysteme $\mathcal{T}_1, \ldots, \mathcal{T}_{n+1}$ bezüglich des Basiskoordinatensystems $\mathcal{B}$ beschreiben. Nach der Konvention in Kapitel 3.1 und Gleichung (3.5) erhält man für die Lage der Toolkoordinatensysteme:

$$\widehat{\boldsymbol{W}}_i = \begin{pmatrix} \widehat{\boldsymbol{R}}_{W_i} & \widehat{\boldsymbol{t}}_{W_i} \\ \boldsymbol{0} & 1 \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{R}}(\alpha_i, \beta_i, \gamma_i) & (x_i, y_i, z_i)^T \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{4.22}$$

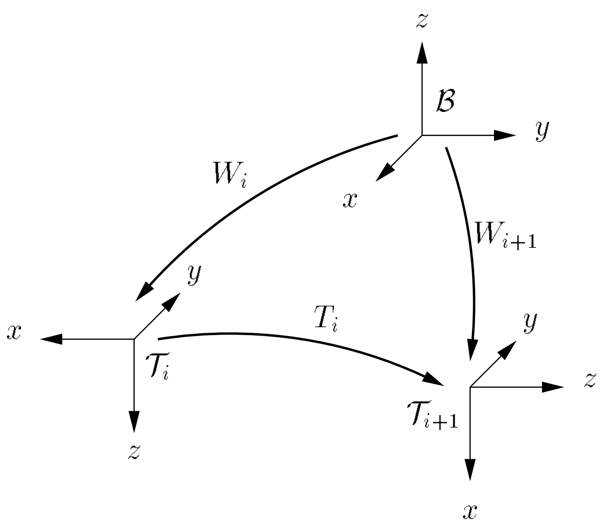


Abbildung 4.6: Diagramm der Abbildungen zwischen dem Basiskoordinatensystem $\mathcal{B}$ und den Toolkoordinatensystemen $\mathcal{T}_i$ und $\mathcal{T}_{i+1}$

Für $\boldsymbol{T}_i$ gilt (siehe Abb. 4.6):

$$\begin{aligned} \boldsymbol{T}_i &= \boldsymbol{W}_{i+1}\boldsymbol{W}_i{}^{-1} && (4.23)\\ &= \widehat{\boldsymbol{W}}_{i+1}{}^{-1}\widehat{\boldsymbol{W}}_i && (4.24) \end{aligned}$$

Somit folgen

$$\begin{aligned} \boldsymbol{R}_{T_i} &= \widehat{\boldsymbol{R}}_{W_{i+1}}{}^T\widehat{\boldsymbol{R}}_{W_i} && (4.25)\\ &= \widehat{\boldsymbol{R}}(\alpha_{i+1},\beta_{i+1},\gamma_{i+1})^T\widehat{\boldsymbol{R}}(\alpha_i,\beta_i,\gamma_i) && (4.26) \end{aligned}$$

und

$$\begin{aligned} \boldsymbol{t}_{T_i} &= \widehat{\boldsymbol{R}}_{W_{i+1}}(\widehat{\boldsymbol{t}}_{W_i}-\widehat{\boldsymbol{t}}_{W_{i+1}}) && (4.27)\\ &= \widehat{\boldsymbol{R}}(\alpha_{i+1},\beta_{i+1},\gamma_{i+1})((x_i,y_i,z_i)^T-(x_{i+1},y_{i+1},z_{i+1})^T) && (4.28) \end{aligned}$$

Die Matrix $\boldsymbol{C}_i$ beschreibt die Transformation der Koordinaten des rekonstruierten Punktes vom Kamerakoordinatensystem $\mathcal{C}_i$ nach $\mathcal{C}_{i+1}$:

$$^{\mathcal{C}_{i+1}}\boldsymbol{M}_{i+1} = \boldsymbol{R}_{C_i}{}^{\mathcal{C}_i}\boldsymbol{M}_i + \boldsymbol{t}_{C_i} \qquad (4.29)$$

Mit $\boldsymbol{R}_{C_i} = \boldsymbol{R}_{T_i}$ folgt:

$$\boldsymbol{t}_{C_i} = {}^{\mathcal{C}_{i+1}}\boldsymbol{M}_{i+1} - \boldsymbol{R}_{T_i}{}^{\mathcal{C}_i}\boldsymbol{M}_i \qquad (4.30)$$

Somit lassen sich aus den Daten in (4.20) und (4.21) die $\boldsymbol{R}_{T_i}, \boldsymbol{t}_{T_i}, \boldsymbol{t}_{C_i}$, $i = 1,\ldots,n$ gemäß (4.26), (4.28) und (4.30) berechnen, die zur Lösung des Gleichungssystems (4.19) benötigt werden.

Kapitel 5

Erkennung von Objektoberflächen

In diesem Kapitel erfolgt die Beschreibung des Objekterkennungsverfahrens für die durch ein ebenes Polygon charakterisierbaren Objekte, wie es bei Seitenflächen polyederförmiger Objekte der Fall ist. Wie in der Einleitung motiviert wurde, ist diese Erkennung modellbasiert.

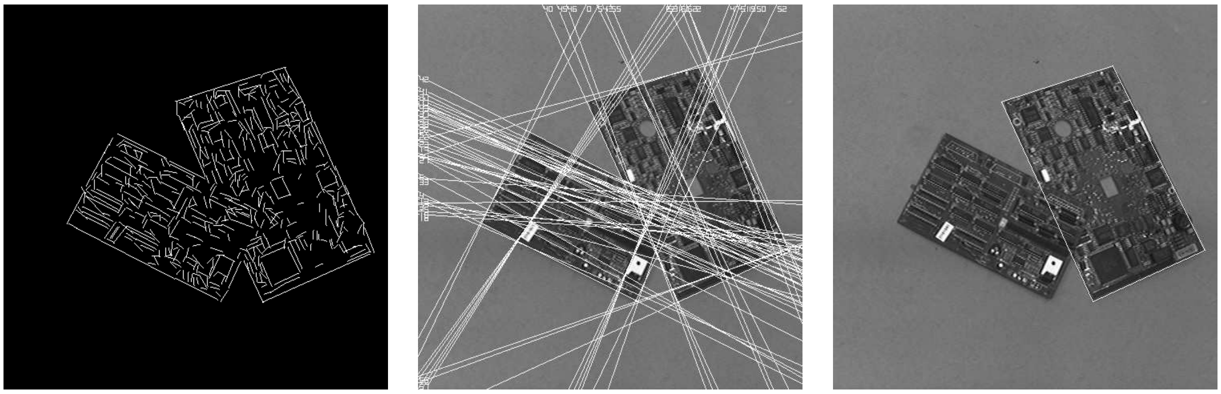
Es stellt sich die Frage, welche Bildprimitiva zugrundegelegt werden können, um letztlich ein 3D-Modell der Oberfläche zu berechnen. Insbesondere erfordert der Übergang von 2D auf 3D eine Rekonstruktion von Bildprimitiva aus mindestens zwei Ansichten, was wiederum die Lösung des Korrespondenzproblems voraussetzt. Es wird nun untersucht, inwiefern Ecken, Bildregionen und Grauwertkanten als grundlegende Bildprimitiva geeignet sind. Die Voraussetzung, daß die Kameras durch das Lochkameramodell beschreibbar sind, ermöglicht ein Stereomatching und eine Rekonstruktion mittels projektiver Geometrie. Von Spezialfällen abgesehen, werden Kanten und Punkte des Polyeders zu Kanten und Punkte in der Bildebene projiziert. Kanten und Punkte in der Bildebene können wiederum zu Kanten bzw. Punkten im dreidimensionalen rekonstruiert werden. Eine Seitenfläche läßt sich am effizientesten durch die den Polygonzug beschreibende Folge von 3D-Punkten repräsentieren. Es wäre somit naheliegend, Ecken in Stereobildern zu detektieren, um dann die 3D-Koordinaten der korrespondieren Ecken zu rekonstruieren. Sowohl für die Rekonstruktion von Punkten als auch für die Berechnung von Punktkorrespondenzen existieren in der Literatur Verfahren (z. B. [9]). Eckendetektoren haben jedoch einen relativ kleinen Einzugsbereich und sind daher sensitiv gegenüber Rauschen. Außerdem können adjazente Punkte, d. h. die eine Kante eines Polygons definierenden Punkte, ohne weitere Informationen nicht bestimmt werden. Eine auf Bildregionen basierende Lokalisation von Oberflächen, bei der Bildregionen auf Basis der sie charakterisierenden Merkmale gruppiert werden und deren Rand das umschließende Polygon bildet, setzt voraus, daß die Merkmale innerhalb des Bildes einer Seitenfläche nicht zu stark variieren. Diese Voraussetzung ist jedoch nicht erfüllt, da sich weitere Objekte auf der Oberfläche befinden. Diese Überlegungen führen zu einer grauwertkantenbasierten Lokalisation der Objektoberflächen. Bevor jedoch eine Rekonstruktion stattfinden kann, muß eine Grauwertkante geometrisch geeignet repräsentiert werden. In den von Zhang und Faugeras [42, 9] vorgestellten Verfahren werden Grauwertkanten durch Segmente approximiert. Als Alternative bietet sich eine globale Detektion von Geraden durch Hough-Transformation an.

Im folgenden sollen nun die Vor- und Nachteile von Segmenten und Geraden bezüglich Robustheit bei der Detektion, Lösung des Korrespondenzproblems und der letztendlichen Gewinnung der 3D-Objektberandung kurz diskutiert werden. Die Szene in Abbildung 5.1 dient als praktisches Beispiel. Abbildung 5.1(a) zeigt das Bild der extrahierten Segmente[1], Abbildung 5.1(b) das Bild mit den detektierten Geraden. Die Bildgeraden wurden durch das auf der Hough-Transformation basierende Verfahren detektiert, welches in Kapitel 5.1 beschrieben wird. In Abbildung 5.1(c) ist die Berandung der verdeckten Platine als Beispiel für eine zu extrahierende Berandung dargestellt.

Das Extraktionsverfahren für Segmente muß robust gegenüber durch Rauschen oder lokale Beleuchtungsänderungen entstehende Unterbrechungen der Grauwertkanten sein, andererseits sehr genau an den Endpunkten der Grauwertkante. In Abbildung 5.1(a) ist erkennbar, daß sich erhebliche Lücken und Überschneidungen an den Berandungen des Objektes ergeben, auch wegen den teilweisen Überdeckungen der Objekte. Die Hough-Transformation zur Detektion von Geraden dagegen ist ein globales Verfahren, somit robust gegen Unterbrechungen im Verlauf von Grauwertkanten. Die Berandung eines Objektes ergibt sich aus einer Folge von Segmenten bzw. Geraden. Im Falle der Segmente ergeben sich aufgrund der Lücken und Überschneidungen erhebliche Probleme bei der Bestimmung adjazenter Segmente.

Auch bezüglich des für die 3D-Rekonstruktion notwendigen Matchings in mindestens zwei Ansichten überwiegen bei Segmenten die Nachteile. Werden nur Anfangs- und Endpunkte zweier Segmente gematcht, wird das Matchingverfahren vergleichsweise instabil bei Fehlkorrespondenzen. Das Matching von mehreren, auf den Segmenten liegenden Punkten führt bei zu kurzen Segmenten aus gleichem Grund zu Problemen. Da Geraden das Resultat re-

[1]Als Segmenteoperator diente das Programm mit dem (fälschlichen) Namen `vpolygon` des Bildverarbeitungssystems `KHOROS`.



(a) Segmente (b) Geraden (c) Objektberandung

Abbildung 5.1: Ergebnis eines Segmenteoperators, Ergebnis der Geradendetektion durch Hough-Transformation, Berandung der überdeckten Platine lativ vieler Kantenpixel sind, läßt sich robuster matchen. Erfolgt die 3D-Rekonstruktion der Segmente durch die Rekonstruktion der Anfangs- und Endpunkte, werden nur zwei Ansichten benötigt. Eine Rekonstruktion der Geraden kann jedoch im allgemeinen nur aus mindestens drei Ansichten erfolgen.

Offensichtlich überwiegen die Vorteile für Geraden als Bildprimitiva, also wurde ein geradenbasiertes Objekterkennungsverfahren entwickelt.

5.1 Detektion von Geraden

5.1.1 Die Hough-Transformation zur Detektion von Geraden

Die Hough-Transformation ist eine Methode, mit der sich Objekte im Bild detektieren lassen, die sich durch Parameter und einer impliziten Funktion auf Bildraum und Parameterraum beschreiben lassen. Die Grundidee dieses Verfahrens geht auf Hough [13] zurück. Es transformiert das Problem der globalen Suche von Bildobjekten in das einfacher zu lösende Problem der Maximumsuche.

Seien N_y und N_x die Anzahl der Zeilen bzw. Spalten der Grauwertbilder, so ist $G = \{0, \ldots, N_y - 1\} \times \{0, \ldots, N_x - 1\}$ der Definitionsbereich des Bildes, also die Menge der Bildkoordinaten. Die Detektion der Geraden basiert auf Pixel, an denen ein bestimmtes Prädikat erfüllt ist, z. B. das Vorhandensein einer Grauwertkante. Die Menge dieser Koordinaten sei mit E bezeichnet. Der Parameterraum P enthält die Parametervektoren $(\phi, d) \in \mathbb{R}$, die sich gemäß der Parameterdarstellung einer Geraden im $\mathbb{R}^2$

$$g(x, y, \phi, d) = (x - x_{org}) \cos(\phi) + (y - y_{org}) \sin(\phi) - d = 0 \tag{5.1}$$

beschreiben lassen. Der Parameter ϕ ist der Winkel zwischen der x-Achse und dem Normaleneinheitsvektor $\boldsymbol{n} = (\cos\phi, \sin\phi)^T$, der Abstand der Geraden zum Ursprung $(x_{org}, y_{org}) = (\frac{N_x}{2}, \frac{N_y}{2})$ ist $|d|$. Aufgrund der Wahl des Ursprungs gilt $-b \leq d \leq b$, wobei $b = \frac{1}{2}\|(N_x, N_y)\|$ die halbe Länge der Bilddiagonalen ist. Somit ist $P = \{(\phi, d) | \phi \in [-\frac{\pi}{2}, \frac{\pi}{2}[, -b \leq d \leq b\}$ der kleinste Parameterraum, der alle Geraden im Bild beschreiben kann. Dieser wird durch eine Matrix der Dimension $N_d \times N_\phi$ diskretisiert.

Die Hough-Transformierte $H : P \to \mathbb{N}_0$ ist definiert durch

$$H(\phi, d) = \sum_{(x,y) \in E, (\phi,d) \in P} h(x, y, \phi, d) \tag{5.2}$$

mit der Funktion $h : E \times P \to \{0, 1\}$,

$$h(x, y, \phi, d) = \begin{cases} 1 & \text{, falls } f(x, y, \phi, d) = 0 \\ 0 & \text{, sonst} \end{cases} \tag{5.3}$$

Für jedes $(x, y) \in E$ findet ein Inkrement bei den Parametern (ϕ, d) statt, die eine mögliche Gerade durch (x, y) beschreiben.

Sind beispielsweise (x_1, y_1) und (x_2, y_2) zwei Bildpunkte einer Geraden, die durch (ϕ^*, d^*) definiert ist, so gilt für die Hough-Transformierte $H(\phi^*, d^*) = 2$, da beide Punkte auf dieser Geraden liegen. Da es offensichtlich keine andere Gerade mit dieser Eigenschaft gibt, gilt $H(\phi^*, d^*) < 2$ für alle $(\phi, d) \neq (\phi^*, d^*)$.

Die wesentlichen Eigenschaften der Hough-Transformation sind:

1. Sie ist global, sie erfolgt ohne Einbeziehung der Umgebung eines Bildpunktes. Daher kann auch eine Bildgerade detektiert werden, deren Pixel nicht zusammenhängend sind. Diese Eigenschaft ist der Grund für die hohe Robustheit des Verfahrens

2. Der Wert eines Maximums ist die Anzahl der Bildpunkte auf der Geraden, was eine Selektion von Geraden ermöglicht.

Die Abbildung H wird durch ein diskretes Akkumulatorfeld $\mathcal{H}$ realisiert. Mit

$$\phi_i = -\frac{\pi}{2} + i \cdot \frac{\pi}{N_\phi} \qquad \text{und} \qquad d_j = -b + j \cdot \frac{2b}{N_d} \tag{5.4}$$

bilden $\Phi = \{\phi_i \,|\, i = 0, \ldots N_\phi - 1\}$ und $D = \{d_i \,|\, i = 0, \ldots N_d - 1\}$ den Parameterraum der detektierbaren Geraden. Unter Verwendung von (5.1) ergibt sich der Algorithmus für die Standard-Hough-Transformation.

initialisiere $\mathcal{H}$
for all $(x, y) \in E$ **do**
    for all $\phi_i \in \Phi$ **do**
        $d_j := \mathrm{rd}((x - x_{org})\sin(\phi_i) + (y - y_{org})\cos(\phi_i))$
        $\mathcal{H}(\phi_i, d_j) := \mathcal{H}(\phi_i, d_j) + 1$
    end for
end for

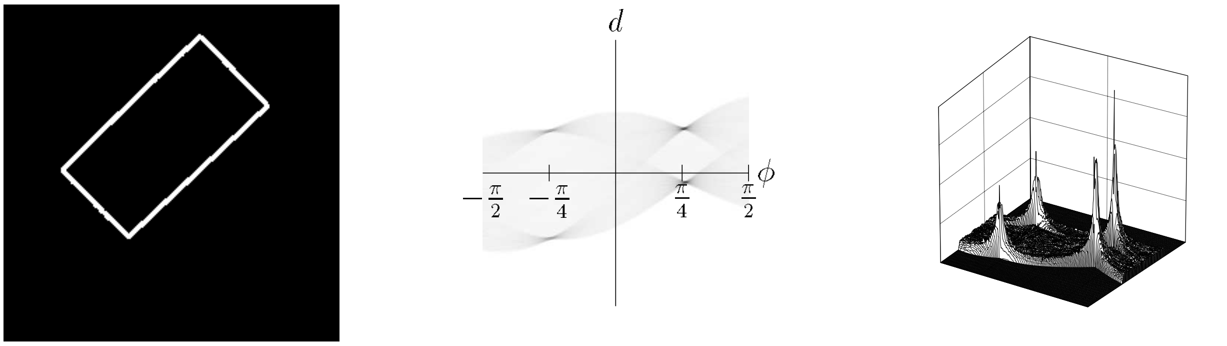
Algorithmus 1: Standard-Hough-Transformation

Abbildung 5.2 zeigt die Hough-Transformation eines Binärbildes. Es ergeben sich 4 deutliche Peaks. Aufgrund der Polarform ist das Akkumulatorfeld $\mathcal{H}$ zyklisch gespiegelt, wie in der 2D-Darstellung deutlich wird. Die dreidimensionale Graphik zeigt auch die Beziehung zwischen der Höhe der Peaks und der Anzahl der Pixel auf einer Geraden im Bild.

Zwecks Einsparung von Rechenzeit bei der Transformation wird für sin und cos jeweils eine look-up-table

$$LUT_{\sin}(i) = \sin(\phi_i), \quad LUT_{\cos}(i) = \sin(\phi_i) \quad i = 0, \ldots N_\phi - 1 \tag{5.5}$$

generiert.



(a) (b) (c)

Abbildung 5.2: Hough-Transformation eines Binärbildes mit 4 Geraden

5.1.2 Orientierungsselektive, schnelle Hough-Transformation

Die im vorhergehenden Abschnitt beschriebene Hough-Transformation ist sehr rechenintensiv, da der Akkumulator für jede Orientierung inkrementiert werden muß, also pro Bildpunkt N_ϕ-mal. Steht die Orientierung der Kante zur Verfügung, so braucht nur eine Akkumulatorzelle inkrementiert werden. Die Transformation wird als orientierungsselektive Hough-Transformation bezeichnet [14]. Um eine größere Fehlertoleranz zuzulassen, empfiehlt sich jedoch eine Inkrementierung innerhalb einer kleinen Umgebung um die detektierte Orientierung unter Berücksichtigung der Zyklizität. Da der Normalenvektor der Geraden dem Gradienten an der Grauwertkante entspricht, ist die Verwendung des Sobeloperators als Kantendetektor naheliegend.

Der Sobeloperator setzt sich aus den beiden Filtermasken

$$S^h = \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} * \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{5.6}$$

$$S^v = \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \frac{1}{8}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{5.7}$$

zusammen, die den symmetrischen Differenzenquotienten aus dem senkrecht zur Ableitungsrichtung geglätteten Bild berechnen. Sind s^h und s^v die Bildmatrizen, die sich durch Anwendung der Filter S^h und S^v ergeben, so ist die Antwort s des Sobeloperators punktweise definiert durch

$$s_{i,j} = \sqrt{(s^h_{i,j})^2 + (s^v_{i,j})^2} \tag{5.8}$$

Die Orientierung der Geraden im Bildpunkt ergibt sich dann aus

$$\phi_{i,j} = \begin{cases} \operatorname{atan}(\frac{s^v_{i,j}}{s^h_{i,j}}) & \text{, falls } s^h_{i,j} \neq 0 \\ -\frac{\pi}{2} & \text{, sonst} \end{cases} \tag{5.9}$$

Durch die 8-bit-Quantisierung der Grauwerte im ganzzahligen Intervall $[0, 255]$ nehmen ∇_x und ∇_y durch die Faltung ganzzahlige Werte im Intervall $[-127, 127]$ an. Dadurch kann eine look-up-table

$$LUT_\phi : [-127, 127] \times [-127, 127] \rightarrow [-\frac{\pi}{2}, \frac{\pi}{2}[; (s^h_{i,j}, s^v_{i,j}) \mapsto \phi_{i,j} \tag{5.10}$$

generiert werden, was den Rechenaufwand bei der Bildverarbeitung erheblich reduziert.

Die Hough-Transformation wird auf die Pixel angewendet, deren Sobelantwort einen Schwellwert θ_{Sobel} überschreitet. Daher kann auch die Berechnung der Wurzel in (5.8) aus Effizienzgründen entfallen.

Neben dem geringeren Rechenaufwand hat die orientierungsselektive Hough-Transformation gegenüber der Standard-Hough-Transformation den Vorteil, daß wesentlich schärfere, und somit besser voneinander trennbare Peaks entstehen, was durch einen Vergleich der Abbildungen 5.2(b) und 5.3(a) bestätigt wird.

5.1.3 Bestimmung der Maxima im Hough-Raum

Die Bestimmung der lokalen Maxima im Akkumulator ist ein schlecht gestelltes Problem und wird in der Literatur ausgiebig behandelt [26, 27, 37]. Als robustes und schnelles Verfahren hat sich die Anwendung eines Schwellwertoperators auf den Akkumulator mit anschließender Suche der Maxima innerhalb der entstandenen Zusammenhangskomponenten erwiesen.

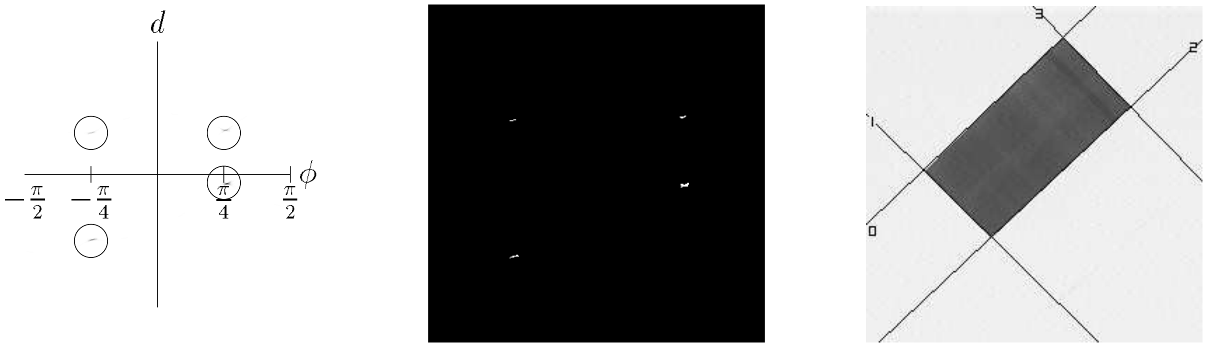
$$c_{i,j} = \begin{cases} 1 & \text{, falls } \mathcal{H}_{i,j} \geq \theta_{\text{Hough}} \\ 0 & \text{, sonst} \end{cases} \tag{5.11}$$

Der Schwellwert θ_{Hough} gibt die Mindestanzahl der Kantenpunkte an, die für eine Gerade votiert haben müssen. Somit läßt sich auf die Sensitivität der Geradendetektion Einfluß nehmen.

Bei der Bestimmung des Maximums innerhalb einer Komponente muß berücksichtigt werden, daß das Akkumulatorfeld zyklisch gespiegelt ist. Der Zusammenhang basiert auf der 8-Nachbarschaft der Pixel. Das Ergebnis der Maximasuche ist eine Menge

$$M = \{(\phi^*_1, d^*_1), \ldots, (\phi^*_{n-1}, d^*_{n-1})\} \tag{5.12}$$

Abbildung 5.3 zeigt die Zusammenhangskomponenten, wie sie sich durch Anwendung des Schwellwertoperators auf das Akkumulatorfeld in Abb. 5.3(a) ergeben. Dieses ergibt sich durch Anwendung der orientierungsselektiven Hough-Transformation auf das Binärbild in Abbildung 5.2(a). Durch die orientierungsselektive Hough-Transformation ergeben sich schärfere Peaks und somit auch bei niedrigen Schwellwert kleine Zusammenhangskomponenten. Die Geraden in Abbildung 5.3(c) entsprechen den Maxima in den Komponenten.



(a) (b) (c)

Abbildung 5.3: Akkumulatorfeld, Binärbild der Komponenten und Bild mit detektierten Geraden

5.1.4 Strategien zur Verbesserung der Detektion

5.1.4.1 Ausdünnung des Kantenbildes

Das binäre Kantenbild E entsteht durch Anwendung eines globalen Schwellwertoperators auf das Sobelbild. Da eine Kante kein sprunghafter Anstieg der Grauwerte, sondern relativ glatt ist, weist das Binärbild relativ breite Kurven auf. Gewünscht ist jedoch ein gesetztes Pixel an den Stellen, an denen der Gradient auf der Kante maximal ist. Eine Ausdünnung des Sobelbildes nach einem in [15] vorgestellten Verfahren führt dazu, daß im binarisierten Bild nur an den Stellen Pixel gesetzt sind, an denen der Betrag des Gradienten ein lokales Maximum besitzt.

Bei diesem Verfahren werden die Orientierungsvektoren 4 verschiedenen Sektoren zugeordnet, wie es in Abbildung 5.4 dargestellt ist. Die Unterteilung der Sektoren korrespondiert mit der 8-Nachbarschaft eines Pixels. Zu jedem Pixel im Sobelbild wird geprüft, ob die mit dem Sektor korrespondierenden Pixel aus der Nachbarschaft des Aufpunktes echt größere Sobelantworten aufweisen. Wenn dies zutrifft, ist der Aufpunkt kein lokales Maximum. Er wird auf den Wert 0 zurückgesetzt und liegt somit immer unterhalb der Schwelle. Andern-

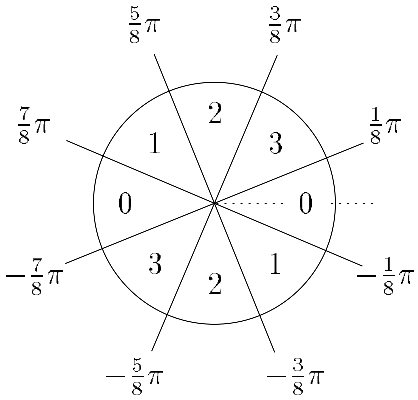
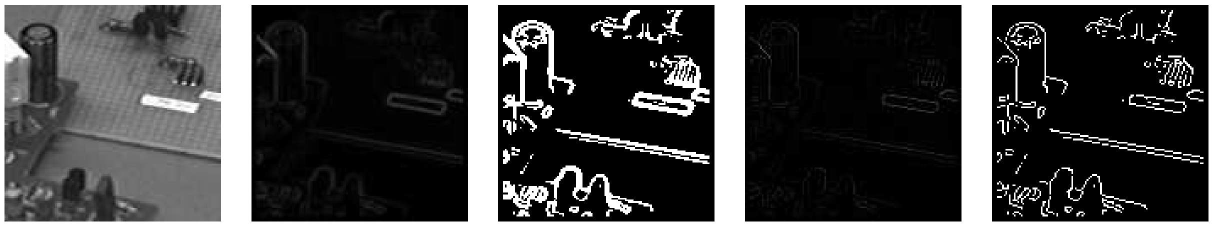


Abbildung 5.4: Partitionierung der möglichen Richtungen des Gradienten in 4 verschiedenen Sektoren (nach [15])

falls bleibt der Wert des Sobeloperators im Aufpunkt.

$$e_{i,j} = \begin{cases} 0 & \text{, falls sector}(s_{i,j}) = 0 \wedge (s_{i,j-1} > s_{i,j} \vee s_{i,j+1} > s_{i,j}) \\ 0 & \text{, falls sector}(s_{i,j}) = 1 \wedge (s_{i-1,j-1} > s_{i,j} \vee s_{i+1,j+1} > s_{i,j}) \\ 0 & \text{, falls sector}(s_{i,j}) = 2 \wedge (s_{i-1,j} > s_{i,j} \vee s_{i+1,j} > s_{i,j}) \\ 0 & \text{, falls sector}(s_{i,j}) = 3 \wedge (s_{i+1,j-1} > s_{i,j} \vee s_{i-1,j+1} > s_{i,j}) \\ s_{i,j} & \text{, sonst} \end{cases} \tag{5.13}$$

Abbildung 5.5 zeigt die Wirkung des Ausdünnungsoperators auf das Bild in Abb. 5.5(a). In Abb. 5.5(b) ist das Sobelbild dargestellt, in Abb. 5.5(d) die Ausdünnung des Sobelbildes. Abbildung 5.5(c) und 5.5(e) zeigen die jeweiligen Binärbilder bei gleichem Schwellwert. Durch die Ausdünnung ergibt sich eine Lokalisation einer Grauwertkante von der Breite eines Pixels.

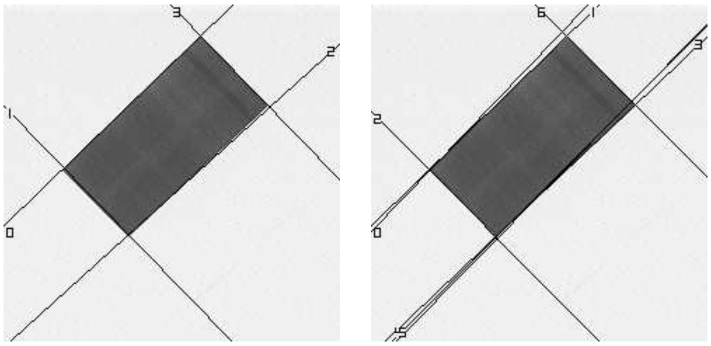


(a) (b) (c) (d) (e)

Abbildung 5.5: Wirkung des Ausdünnungsoperators auf die Kantendetektion

5.1.4.2 Auflösung des Hough-Raumes

Die Wahl der Auflösung bei Diskretisierung des Parameterraumes hat wesentlichen Einfluß auf die Detektion und Lokalisation der Geraden. Ist die Auflösung zu gering, wird der Fehler bei der Lokalisation zu groß. Die Geraden in Abbildung 5.6(a) weichen stärker von den Kanten des Rechtecks ab als in Abbildung 5.3(c), bei der die Auflösung des Parameterraumes $N_d \times N_\phi = 362 \times 360$ beträgt. Andererseits zerfallen die Peaks in mehrere dicht nebeneinanderliegende lokale Maxima, wenn die Auflösung des Parameterraumes zu hoch ist (Abb. 5.6(b)). Der gleiche Effekt tritt auf, wenn N_d gegenüber N_ϕ zu groß ist. Die Steigung der für ein Kantenpixel durchlaufenen sinusförmigen Kurve im Akkumulatorfeld kann dann über dem Wert 1 liegen, so daß sich beim Inkrementieren Lücken ergeben. Es empfiehlt sich daher, N_d nicht größer als N_ϕ zu wählen.



(a) $N_d = 90$, $N_\psi = 90$ (b) $N_d = 724$, $N_\psi = 724$

Abbildung 5.6: Wirkung der Auflösung des Parameterraumes auf die Detektion der Geraden

5.1.5 Erweiterung der Hough-Transformation

Aus den detektierten Geraden allein können die Berandungen eines Objektes nicht eindeutig bestimmt werden. Ebenso ist ein auf der geometrischen Beschreibung von Geraden basierendes Stereomatching nicht möglich, worauf in Kapitel 5.2 noch näher eingegangen wird. Daher muß zur Lösung des Korrespondenzproblems auf die Grauwertinformation, d. h. Pixel und Umgebungen von Pixeln zurückgegriffen werden. Um auf die Grauwerte und Umgebungen von Pixeln zugreifen zu können, die zu einer detektierten Gerade gehören, müssen die Positionen der Pixel gespeichert werden. Dieses kann während der Votierung geschehen. Die Speicherung erfolgt in einem zweidimensionalem Feld $\mathcal{P}$ von Mengen, welches die gleiche Größe wie das Akkumulatorfeld hat und dessen Komponenten als eindimensionale Felder (*arrays*) realisiert sind. Die Anzahl der Votierungen für eine Gerade stimmt somit mit der Anzahl der gespeicherten Positionen überein: $\mathcal{H}(\phi, d) = |\mathcal{P}(\phi, d)|$. Beträgt die Größe des Grauwertbildes (und somit auch des Kantenbildes) $N_y \times N_x$ Pixel, so ist der Ort (i, j) eines Pixels $g_{i,j}$ eindeutig bestimmt durch $p(i, j) = i \cdot N_x + j$, so daß anstelle eines zweidimensionalen Paares nur der eindimensionale Index p gespeichert werden braucht, was den Speicheraufwand halbiert. Somit ergibt sich insgesamt der erweiterte, orientierungsbasierte Algorithmus (siehe folgenden Algorithmus 2).

initialisiere $\mathcal{H}$
initialisiere $\mathcal{P}$
for all $(x, y) \in E$ **do**
  for all $\phi_i \in [-\epsilon + \psi_{y,x}, \psi_{y,x} + \epsilon[_{zyklisch} \cap \Phi$ **do**
    $d := (x - x_{org}) \sin(\phi_i) + (y - y_{org}) \cos(\phi_i)$
    $\mathcal{H}(\phi, d) := \mathcal{H}(\phi, d) + 1$
    $\mathcal{P}(\phi, d)[\mathcal{H}(\phi, d)] := p(i, j)$
  end for
end for

Algorithmus 2: Die erweiterte, orientierungsbasierte Hough-Transformation

5.1.6 Berechnung der Geraden

Jedem Maximum $(\phi, d) \in M$ des Akkumulators entspricht eine Gerade im Bild. Es werden zwei Darstellungen der 2D-Geraden benötigt:

1. Darstellung als projektiver Vektor:
   Die Punkt-Gleichung einer Geraden lautet $\tilde{\boldsymbol{m}} \cdot \tilde{\boldsymbol{l}} = 0$, wobei $\tilde{\boldsymbol{l}}$ der die Gerade repräsentierende projektive Vektor ist und $\tilde{\boldsymbol{m}}$ der eines beliebigen Punktes in der projektiven Ebene $\mathcal{P}^2$. Mit Gleichung (5.1) folgt
   $$\tilde{\boldsymbol{l}} = \begin{pmatrix} \cos(\phi) \\ \sin(\phi) \\ -d - x_{org} \cdot \cos(\phi) - y_{org} \cdot \sin(\phi) \end{pmatrix} \tag{5.14}$$
   Diese Darstellung wird für die Rekonstruktion der Geraden benötigt (Kapitel 5.3).

2. In Punkt-Richtungs-Form:
   Die Gerade wird durch durch einen Aufpunkt $\boldsymbol{a}$ und einen Richtungsvektor $\boldsymbol{u}$ beschrieben:
   $$\boldsymbol{g}(\lambda) = \boldsymbol{a} + \lambda \cdot \boldsymbol{u} \tag{5.15}$$
   Der Richtungsvektor $\boldsymbol{u}$ steht senkrecht auf dem Normalenvektor $\boldsymbol{n}$:
   $$\boldsymbol{u} = (-\sin(\phi), \cos(\phi))^T \tag{5.16}$$
   Der Aufpunkt wird definiert durch
   $$\boldsymbol{a} = (x_{org}, y_{org})^T + d \cdot \boldsymbol{n} = (x_{org}, y_{org})^T + d \cdot (\cos(\phi), \sin(\phi))^T \tag{5.17}$$
   Diese Darstellung dient der Berechnung von Schnittpunkten zweier 2D-Geraden (Kapitel 5.6.1).

Das Ergebnis der Detektion sind also zwei Mengen von Geraden für beide Repräsentationen:
$$\{\tilde{\boldsymbol{l}}_0, \ldots, \tilde{\boldsymbol{l}}_{n-1}\} \quad \text{und} \quad \{{}^{2\mathrm{D}}\boldsymbol{g}_0, \ldots, {}^{2\mathrm{D}}\boldsymbol{g}_{n-1}\} \tag{5.18}$$

5.2 Korrespondenzen von Geraden, Stereomatching

Um geometrische Objekte wie Punkte oder Geraden rekonstruieren zu können, müssen die korrespondierenden Bildpunkte oder Bildgeraden gefunden werden. Das Ergebnis der bisherigen Bildverarbeitung sind die geometrischen Daten der Bildgeraden in den Kamerabildern. Erschwerend für die Lösung des Korrespondenzproblems ist, daß die realen 3D-Geraden beliebig im Raum liegen können. Somit gibt es keine speziellen geometrischen Voraussetzungen, die in ein Matchingverfahren einfließen könnten.

Im Falle des Matchings von Punkten reduziert sich die Suche des zu einem Bildpunkt $\boldsymbol{m}^{(1)}$ in der Bildebene $\mathcal{R}^{(1)}$ korrespondierenden Bildpunktes $\boldsymbol{m}^{(2)}$ in der Bildebene $\mathcal{R}^{(2)}$ durch Ausnutzung der epipolaren Bedingung (*epipolar constraint* [9]) auf ein eindimensionales Suchproblem, da ein Kandidat für $\boldsymbol{m}^{(1)}$ auf einer speziellen Geraden, der sogenannten Epipolaren, liegt. Eine für projektive Geraden analoge Eigenschaft gibt es nicht. Daher erfolgt das Matching auf Basis der Punkte auf den Geraden, die während der Hough-Transformation in $\mathcal{P}$ festgehalten wurden. Unter Einbeziehung des Grauwertprofils (Templatematching, SSD) ergibt sich ein Matchingverfahren für Stereobilder und trinokulare Bilder.

5.2.1 Die epipolare Geometrie

Sei ein Punkt $\boldsymbol{m}^{(1)}$ im Bild der linken Kamera gegeben, der die Projektion eines a priori unbekannten Weltpunktes mit den euklidischen Koordinaten $\boldsymbol{M}$ ist. Alle Punkte auf der Geraden durch das optische Zentrum $\boldsymbol{C}^{(1)}$ und den Punkt $\boldsymbol{M}$ werden auf den Bildpunkt $\boldsymbol{m}^{(1)}$ projiziert (siehe Abb. 5.7). Die Projektion dieser Geraden in die Bildebene der rechten Kamera ergibt eine Gerade im rechten Bild. Diese Gerade wird als Epipolare bezeichnet. Da der Punkt $\boldsymbol{M}$ unbekannt ist, betrachtet man den Punkt im Unendlichen auf der Geraden durch $\boldsymbol{C}^{(1)}$ und $\boldsymbol{M}$. Dieser sei mit $\tilde{\boldsymbol{D}}$ bezeichnet. Es gilt

$$\tilde{\boldsymbol{m}}^{(1)} = \tilde{\boldsymbol{P}}^{(1)}\tilde{\boldsymbol{D}} = \begin{pmatrix} \boldsymbol{P}^{(1)} & \tilde{\boldsymbol{p}} \end{pmatrix} \begin{pmatrix} \boldsymbol{D} \\ 0 \end{pmatrix} \tag{5.19}$$

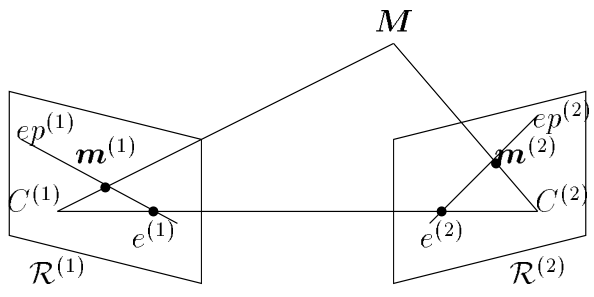


Abbildung 5.7: Die epipolare Geometrie

und somit

$$\boldsymbol{D} = \boldsymbol{P}^{(1)-1}\tilde{\boldsymbol{m}}^{(1)} \tag{5.20}$$

Die Epipolare ist durch den Epipol $\tilde{\boldsymbol{e}}^{(2)}$, der die Projektion des optischen Zentrums $\tilde{\boldsymbol{C}}^{(1)}$ auf die Bildebene $\mathcal{R}^{(2)}$ ist, und der Projektion $\tilde{\boldsymbol{d}}^{(2)}$ des Punktes $\tilde{\boldsymbol{D}}$ auf $\mathcal{R}^{(2)}$ gegeben. Sie verläuft durch die Punkte $\tilde{\boldsymbol{e}}^{(2)}$ und $\tilde{\boldsymbol{d}}^{(2)}$, der die Epipolare darstellende projektive Vektor ist somit $\tilde{\boldsymbol{e}}^{(2)} \times \tilde{\boldsymbol{d}}^{(2)}$. Die Punktgleichung der Gerade lautet also

$$\tilde{\boldsymbol{m}}^{(2)T}(\tilde{\boldsymbol{e}}^{(2)} \times \tilde{\boldsymbol{d}}^{(2)}) = 0 \tag{5.21}$$

Mit $\tilde{\boldsymbol{d}}^{(2)} = \tilde{\boldsymbol{P}}^{(2)}\tilde{\boldsymbol{D}}$, Gleichung (5.20) und der das Kreuzprodukt mit $\tilde{\boldsymbol{e}}^{(2)} = \tilde{\boldsymbol{P}}^{(2)}\tilde{\boldsymbol{C}}^{(1)}$ beschreibenden Matrix

$$\boldsymbol{E}^{(2)} := \begin{pmatrix} 0 & -e_3^{(2)} & e_2^{(2)} \\ e_3^{(2)} & 0 & e_1^{(2)} \\ -e_2^{(2)} & -e_1^{(2)} & 0 \end{pmatrix} \tag{5.22}$$

gilt

$$\tilde{\boldsymbol{e}}^{(2)} \times \tilde{\boldsymbol{d}}^{(2)} = \boldsymbol{E}^{(2)}\boldsymbol{P}^{(2)}\boldsymbol{P}^{(1)-1}\tilde{\boldsymbol{m}}^{(1)} \tag{5.23}$$

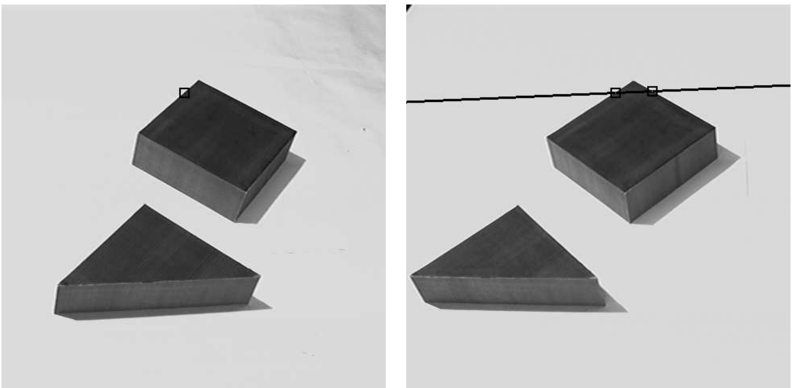
Die Matrix $\boldsymbol{F} := \boldsymbol{E}^{(2)}\boldsymbol{P}^{(2)}\boldsymbol{P}^{(1)-1}$ wird *Fundamentalmatrix* genannt. Zwei korrespondierende Punkte erfüllen somit die *Longuet-Higgins-Gleichung* [9]

$$\tilde{\boldsymbol{m}}^{(2)T}\boldsymbol{F}\tilde{\boldsymbol{m}}^{(1)} = 0 \tag{5.24}$$

Abbildung 5.8 zeigt ein Stereobildpaar. Der Punkt $\boldsymbol{m}^{(1)}$ liegt in der Mitte des durch ein Quadrat markierten Bildausschnitts im linken Bild (Abb. 5.8(a)). Die Epipolare $ep^{(2)}$ im rechten Bild, auf der die zu $\boldsymbol{m}^{(1)}$ korrespondierenden Punkte liegen können, ist in Abbildung 5.8(b) dargestellt.

5.2.2 Matching auf photometrischer Basis

Die epipolare Bedingung reicht nicht aus, um den korrespondierenden Punkt eindeutig zu bestimmen. Selbst bei beidseitiger Ausnutzung von Gleichung (5.24) gibt es Fälle, in denen das Korrespondenzproblem für Punkte ein eindimensionales Suchproblem bleibt. Hinzu kommt, daß bei digitalen Bildern Gleichung (5.24) nicht direkt anwendbar ist, da die Pixelkoordinaten diskret sind. Die Suche des Kandidaten erfolgt daher in einer kleinen Umgebung im Bild, in der die Epipolare liegt. Die „Breite“ dieses Streifens hängt insbesondere von der Genauigkeit der Kalibrierung ab. Da also geometrische Eigenschaften



(a) Bild der linken Kamera $\mathcal{R}^{(1)}$ (b) Bild der rechten Kamera $\mathcal{R}^{(2)}$

Abbildung 5.8: Punkt im Zentrum des quadratischen Bildausschnitts auf der Kante eines Objektes im linken Bild, Epipolare dieses Punktes im rechten Bild

allein nicht zur Lösung des Korrespondenzproblems ausreichen, müssen weitere, photometrische Merkmale herangezogen werden. Als zuverlässiges Kriterium hat sich die Abweichung der Grauwertprofile um die zu testenden Bildpunkte erwiesen. Dieses Verfahren wird als Template-Matching bezeichnet. Ballard [3] gibt verschiedene Distanzmaße an; die Auswahl eines geeigneten Maßes hängt von der Beschaffenheit der verwendeten Kameras, den Beleuchtungsverhältnissen und den projektiven Eigenschaften der Stereoanordnung ab.

Es wird von folgenden Voraussetzungen ausgegangen:

1. Die Kameras sind in gleicher Weise konstruiert, die Brennweite der Objektive ist ebenfalls gleich.
2. Die Ausleuchtung ist relativ homogen und die Objekte besitzen bezüglich beider Kameras dieselben Reflektanzeigenschaften.
3. Die Winkeldifferenz der Orientierungen zweier korrespondierender Grauwertkanten ist so gering, daß die Grauwertprofile in einer Umgebung ähnlich genug sind.

Unter diesen Voraussetzungen genügt als Ähnlichkeitsmaß die Summe der quadratischen Differenzen (SSD) zweier gleichgroßer Teilbilder $h^{(1)}$ und $h^{(2)}$ der Größe $b \times b$:

$$d(h^{(1)}, h^{(2)}) = \sum_{k=0,l=0}^{b} (h^{(1)}_{k,l} - h^{(2)}_{k,l})^2 \tag{5.25}$$

Damit $h^{(1)}$ und $h^{(2)}$ bezüglich ihrer Zentren symmetrisch sind, ist b ungerade. Sind $(i^{(1)}, j^{(1)})$ und $(i^{(2)}, j^{(2)})$ die Koordinaten der Zentren in den Bildern $h^{(1)}$ und $g^{(2)}$, so gilt

$$h^{(1)}_{k,l} = g^{(1)}_{i^{(1)}-\frac{b}{2}+k,\, j^{(1)}-\frac{b}{2}+l} \qquad \text{und} \qquad h^{(2)}_{k,l} = g^{(2)}_{i^{(2)}-\frac{b}{2}+k,\, j^{(2)}-\frac{b}{2}+l} \tag{5.26}$$

Somit läßt sich eine Metrik

$$d_b((i^{(1)}, j^{(1)}), (i^{(2)}, j^{(2)})) \tag{5.27}$$

definieren, welche die Ähnlichkeit der Grauwertprofile um die Punkte $(i^{(1)}, j^{(1)})$ und $(i^{(2)}, j^{(2)})$ in den Bildern $g^{(1)}$ bzw. $g^{(2)}$ mißt.

Die Quadrate in Abbildung 5.8 markieren Teilbilder. Offensichtlich paßt das Teilbild auf der linken Kante des Objektes im rechten Bild besser zum Teilbild im linken Bild als das Teilbild auf der rechten Kante. Somit läßt sich in diesem Beispiel das Korrespondenzproblem eindeutig lösen. In komplexeren Szenen können jedoch trotz Kombination von epipolarer Geometrie und Template-Matching falsche Korrespondenzen auftreten, daher ist ein Votierungsverfahren Teil des Stereo-Matchings. Dieses Votierungsverfahren ist insbesondere Gegenstand des nun folgenden Abschnitts.

5.2.3 Verbindung beider Ansätze zu einem Matchingverfahren

In diesem Kapitel soll nun das im Rahmen dieser Diplomarbeit entwickelte Matchingverfahren zur Lösung des Korrespondenzproblems für Geraden vorgestellt werden. In dieses Verfahren fließen die epipolare Bedingung und das Template-Matching ein, sowie ein Votierungsverfahren zwecks Erhöhung der Robustheit.

Zur Rekonstruktion von Geraden werden (mindestens) 3 korrespondierende Bildgeraden von 3 Ansichten benötigt. Daher müssen Tripel korrespondierender Bildgeraden gefunden werden. Diese ergeben sich aus dem Matching zweier Bildpaare, wobei das eine Bildpaar durch die erste und zweite Ansicht, das andere durch die zweite und dritte Ansicht gegeben ist. In Abschnitt 5.2.3.1 wird das Matchingverfahren für 2 Ansichten beschrieben, in Abschnitt 5.2.3.2 die Berechnung von Korrespondenztripeln aus Korrespondenzpaaren von Geraden der zwei Bildpaare.

5.2.3.1 Matching von Geraden eines Bildpaares

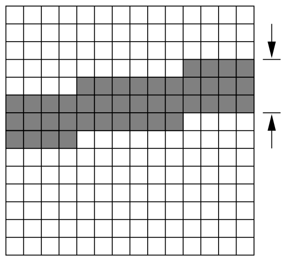
Im folgenden werden zwei Ansichten betrachtet, die durch a und b indiziert seien. Es werden die Pixel in das Matching einbezogen, die auf einer Geraden liegen. Sowohl bei der Standard-Hough-Transformation als auch bei der orientierungsselektiven Hough-Transformation kann ein Pixel zu mehr als einer Geraden gehören. Bei der orientierungsselektiven Hough-Transformation kann dieser Fall auftreten, wenn zwecks Fehlertoleranz auch in einer Umgebung um die detektierte Orientierung votiert wird.

In jedem Bildpunkt wird festgehalten, welche Bildgerade durch diesen hindurchführt. Die Koordinaten der Kantenpixel einer Geraden sind in der entsprechenden Komponente des zweidimensionalen Feldes $\mathcal{P}$ gespeichert. Sei

$$z_{i,j}^{(a)} = \{\iota_0^{(a)}, \ldots, \iota_{n_{i,j}^{(a)}-1}^{(a)}\} \tag{5.28}$$

die Menge von Indizes der Geraden durch den Bildpunkt (i, j) des a-ten Bildes. Entsprechend sei $z_{i,j}^{(b)}$ für jeden Bildpunkt (i, j) des b-ten Bildes definiert.

Beschreibt (i, j) ein Pixel in einem Bild, so erfolgt eine Suche unter den Pixeln im anderen Bild, die in einer Umgebung um die Epipolare des Punktes (i, j) liegen. Die Epipolare ist durch Gleichung (5.23) gegeben, mit der projektiven Darstellung $\tilde{\boldsymbol{m}} = (i, j, 1)^T$ des Bildpunktes (i, j).



Der Anfangs- und Endpunkt des Scanbereiches ist durch den Schnitt der Epipolaren mit dem Bildrand gegeben. Aufgrund der Geometrie der verwendeten Kameras liegt die Steigung der Epipolaren im Intervall $[-1, 1]$. Somit läßt sich die „Breite“ des Scanbereiches sinnvollerweise durch die Anzahl der Pixel in der Vertikalen festlegen (siehe nebenstehende Abbildung), was den Scanalgorithmus vereinfacht. Um die Notation im folgenden zu vereinfachen, sei der Scanbereich durch die Menge $\mathrm{Epi}(i, j)$ repräsentiert. Nun werden die Mengen

$$e_{i,j}^{(a)} = \begin{cases} \emptyset & \text{, falls } z_{i,j}^{(a)} = \emptyset \\ \{((s,t), \iota^{(b)}) \,|\, (s,t) \in \mathrm{Epi}(i,j) \wedge \iota^{(b)} \in z_{s,t}^{(b)}\} & \text{, falls } z_{i,j}^{(a)} \neq \emptyset \end{cases} \tag{5.29}$$

$$f_{i,j}^{(a)} = \begin{cases} \emptyset & \text{, falls } z_{i,j}^{(a)} = \emptyset \\ \{((s,t), \iota^{(b)}) \,|\, (s,t) \in N \times N \wedge (i,j) \in \mathrm{Epi}(s,t) \wedge \iota^{(b)} \in z_{s,t}^{(b)}\} & \text{, falls } z_{i,j}^{(a)} \neq \emptyset \end{cases} \tag{5.30}$$

berechnet. Die Menge $e_{i,j}^{(a)}$ beschreibt die Bildpunkte im b-ten Bild, die sich in dem durch die Epipolare des im a-ten Bild liegenden Bildpunktes (i, j) gegebenen Scanbereich befinden. Die Menge $f_{i,j}^{(a)}$ beschreibt die Bildpunkte im b-ten Bild, deren Scanbereich den im a-ten Bild liegenden Bildpunkt (i, j) trifft. Dabei werden nur Pixel einbezogen, die auf einer detektierten Gerade liegen. Da die Indizes der entsprechenden Geraden des weiteren relevant sind, bestehen beide Mengen aus Paaren, die sich aus der Kombination von Position und Geradenindizes ergeben.

Unter der Voraussetzung, daß die Orientierungen zweier korrespondierender 2D-Geraden im Stereobildpaar nur relativ wenig voneinander abweichen, kann, um die Mächtigkeit der Mengen $e_{i,j}^{(a)}$ und $f_{i,j}^{(a)}$ zu reduzieren, ein Zusatzkriterium formuliert werden. Es werden nur Paare von Geraden betrachtet, deren Orientierungsdifferenz unterhalb einer Schwelle liegt. Ist (i, j) ein Punkt auf einer Gerade $g_\kappa^{(a)}$, so wird das Tupel $((s,t), \iota_b)$ dann zu $e_{i,j}^{(a)}$ bzw. $f_{i,j}^{(a)}$ hinzugefügt, wenn

$$|\phi_\kappa^{(a)} - \phi_\iota^{(b)}|_{zyklisch} \leq \theta \tag{5.31}$$

Die Winkel $\phi_\kappa^{(a)}$ und $\phi_\iota^{(b)}$ sind die aus der Maximasuche im Akkumulatorfeld der beiden Ansichten resultierenden Winkel (Gleichung (5.12)). Der Schnitt

$$s_{i,j}^{(a)} = e_{i,j}^{(a)} \cap f_{i,j}^{(a)} \tag{5.32}$$

realisiert die beidseitige Ausnutzung der epipolaren Bedingung. In $s_{i,j}^{(a)}$ sind die Pixel (mit Geradenzugehörigkeit) gespeichert, die geometrisch mit (i,j) korrespondieren können. Durch Anwendung des SSD-Verfahrens (Gleichung (5.27) läßt sich die Ähnlichkeit der Grauwertumgebung von (i,j) mit den in $s_{i,j}^{(a)}$ enthaltenen Punkten vergleichen. Die Wahrscheinlichkeit, das zwei (oder mehrere) Punkte aus $s_{i,j}^{(a)}$ denselben minimalen Abstand zu (i,j) im Sinne der definierten Metrik besitzen, ist sehr gering. Tritt dieser Fall dennoch auf, kann einer dieser Punkte zufällig ausgewählt werden. Auf diese Weise läßt sich genau ein Punkt aus $s_{i,j}^{(a)}$ bestimmen, der am besten zum Bildpunkt (i,j) im a-ten Bild paßt und somit auch die Gerade, die zu (i,j) paßt.

Trotz der Ausnutzung der epipolaren Bedingung und des Template-Matchings ergeben sich falsche Korrespondenzen. Die Ursachen sind Rauschen, inhomogene Lichtverhältnisse sowie Verdeckungen. Das den Grauwertbildern aufgeprägte Rauschen führt zu einer fehlerbehafteten Orientierung an den Grauwertkanten. Ist für ein Kantenpixel, durch das eine detektierte Gerade hindurchführt, der Fehler zu groß, so votierte dieser Punkt bei der orientierungsbasierten Hough-Transformation nicht für diese Gerade, folglich wird dieser Punkt nicht in das Matchingverfahren einbezogen. Auf diese Weise können Lücken auf einer Objektkante entstehen und es kommt zur Bildung von Punktkorrespondenzen mit Punkten auf nicht korrespondierenden Geraden. Inhomogene Lichtverhältnisse sowie durch die verschiedenen Blickrichtungen hervorgerufene Verdeckungen führen ebenso zum Nichtvorhandensein korrespondierender Kantenpixel.

Ein Matching von Geraden aufgrund nur eines Paares korrespondierender Punkte würde also zu einer hohen Anzahl falscher Korrespondenzen von Geraden führen. Führt man jedoch diese Suche des korrespondierenden Punktes für jeden Bildpunkt auf einer Geraden im a-ten Bild durch, so läßt sich sich durch Votierung für jede Gerade im b-ten Bild ermitteln, wieviele Punktkorrespondenzen existieren. Die Gerade mit den meisten Punktkorrespondenzen ist dann die korrespondierende Gerade. Da die Kantenpixel der Geraden bei der Hough-Transformation in $\mathcal{P}$ gespeichert wurden, läßt sich die Votierung sehr effizient durchführen. Algorithmus 3 präzisiert das Votierungsverfahren. Das Akkumulatorfeld, dessen Dimension gleich der Anzahl der Geraden im b-ten Bild ist, sei mit $\mathcal{V}$ bezeichnet. Die Menge $K^{a\rightarrow b}$ ist die Relation, die man durch ein Matching der Geraden vom a-ten ins b-te Bild erhält.

Die Relation $K^{a\rightarrow b}$ ist eindeutig, d. h. zu jeder Geraden im a-ten Bild existiert genau eine Gerade im b-ten Bild. Diese Eigenschaft gilt jedoch in der Realität nicht immer. Abbildung 5.9 zeigt eine spezielle Stereoansicht einer Ecke eines Quaders. Die markierte Gerade im linken Bild hat zwei korrespondierende Geraden im rechten Bild. Es ist also ebenfalls ein Matching in umgekehrter Richtung notwendig. Dadurch erhält man die Menge $K^{b\rightarrow a}$. Die Menge von Paaren, welche die Korrespondenzen in korrekter Weise ausdrückt, erhält man durch Vereinigung der Relationen $K^{a\rightarrow b}$ und $K^{b\rightarrow a}$ unter Berücksichtigung, daß die Paare in $K^{b\rightarrow a}$ transponiert sind:

$$K^{a,b} := \{(\iota^{(a)}, \iota^{(b)}) \mid (\iota^{(a)}, \iota^{(b)}) \in K^{a\rightarrow b} \vee (\iota^{(b)}, \iota^{(a)}) \in K^{b\rightarrow a}\} \tag{5.33}$$

$K^{a\to b} := \emptyset$ // initialisiere Menge von Korrespondenzpaaren
for all $\iota \in \{0, \dots, n^{(a)} - 1\}$ **do**
  $\mathcal{V} := \mathbf{0}$ // initialisiere Akkumulatorfeld
  for all $(i,j) \in \mathcal{P}({}^{2\mathrm{D}}g_\iota^{(a)})$ **do**
    $\iota^{(b)} := \underset{((k,l),\iota^{(b)}) \in s_{i,j}^{(a)}}{\arg\min} d((i,j),(k,l))$ // Template-Matching
    $\mathcal{V}_{\iota^{(b)}} := \mathcal{V}_{\iota^{(b)}} + 1$
  end for
  $\iota_{\text{match}}^{(b)} = \arg\max \mathcal{V}_{i^{(b)}}$ // Index der dem Maximum von $\mathcal{V}$ entsprechenden Gerade
  $K^{a\to b} = K^{a\to b} \cup (\iota^{(a)}, \iota_{\text{match}}^{(b)})$
end for

Algorithmus 3: Votierungsverfahren

Die Vereinigung der beiden Relationen ist korrekt unter folgenden Voraussetzungen:

1. Zu einer Gerade in einem Bild wird auch die entsprechende Gerade im anderen Bild detektiert.
2. An einer Grauwertkante, die zu einer realen Objektkante gehört, wird genau eine Gerade detektiert.

Obwohl die Hough-Transformation ein sehr robustes Verfahren zur Detektion von Geraden ist, sind diese Voraussetzungen nicht immer erfüllt.

Fehlmatchings aufgrund der eben genannten Probleme lassen sich vermeiden, indem man statt der Vereinigung die Paare aus $K^{a\to b}$ und $K^{b\to a}$ schneidet; allerdings muß dann vorausgesetzt werden, daß Mehrdeutigkeiten wie in Abbildung 5.9 nicht auftreten:

$$K^{a,b} := \{(\iota^{(a)}, \iota^{(b)}) \mid (\iota^{(a)}, \iota^{(b)}) \in K^{a\to b} \wedge (\iota^{(b)}, \iota^{(a)}) \in K^{b\to a}\} \tag{5.34}$$

In den folgenden Beispielen wird $K^{a,b}$ gemäß (5.34) verwendet, da nur „große" Oberflächen detektiert werden sollen.

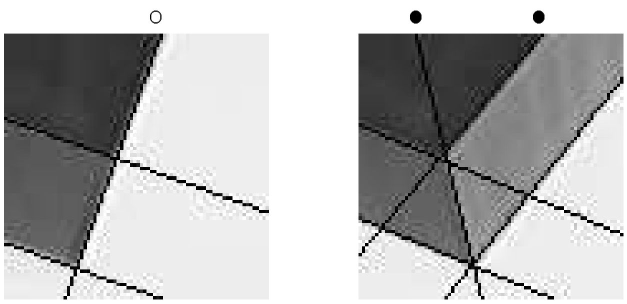


Abbildung 5.9: Die mit $\circ$ markierte Gerade in der linken Ansicht korrespondiert mit den zwei durch $\bullet$ markierten Geraden in der Ansicht der rechten Kamera

Abbildung 5.10 verdeutlicht zusammenfassend das Berechnungsschema für ein Bildpaar. Gegeben sind die Grauwertbilder $g^{(a)}$ und $g^{(b)}$, die Mengen $M^{(a)}$ und $M^{(b)}$, welche die Orientierungen der Geraden enthalten, sowie die Felder $\mathcal{P}^{(a)}$ und $\mathcal{P}^{(b)}$ mit den Mengen der Pixelpositionen für jede detektierte Gerade. Jede der 5 Ebenen gibt die Mengen an, die parallel berechnet werden können. Außerdem können die Mengen einer Ebene für jedes $(i, j) \in N \times N$ unabhängig voneinander berechnet werden. Das Verfahren ist also hochgradig parallelisierbar.

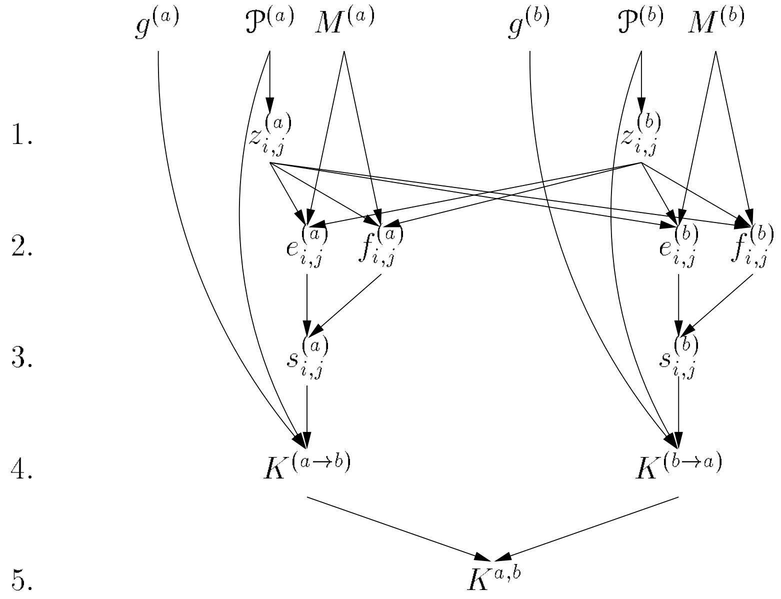


Abbildung 5.10: Zeitliches Berechnungsschema

5.2.3.2 Matching von Geraden eines Bildtripels

Das Ziel ist die Gewinnung der Korrespondenztripel der Geraden in 3 Ansichten. Nach dem im letzten Abschnitt angegebenen Verfahren werden die Mengen von Korrespondenzpaaren $K^{1,2}$ und $K^{2,3}$ berechnet. Zwei Paare $(\iota^{(1)}, \iota^{(2)}) \in K^{1,2}$ und $(\kappa^{(2)}, \iota^{(3)}) \in K^{2,3}$ lassen sich zu einem Tripel korrespondierender Bildgeraden verschmelzen, wenn $\iota^{(2)} = \kappa^{(2)}$. Somit erhält man die Menge

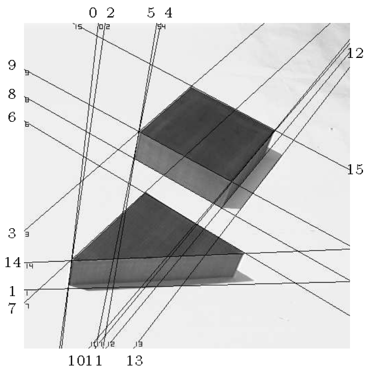
$$K^{1,2,3} = \{(\iota^{(1)}, \iota^{(2)}, \iota^{(3)}) \,|\, \exists(\iota^{(1)}, \iota^{(2)}) \in K^{1,2}, \exists(\kappa^{(2)}, \iota^{(3)}) \in K^{2,3} : \iota^{(2)} = \kappa^{(2)}\} \tag{5.35}$$

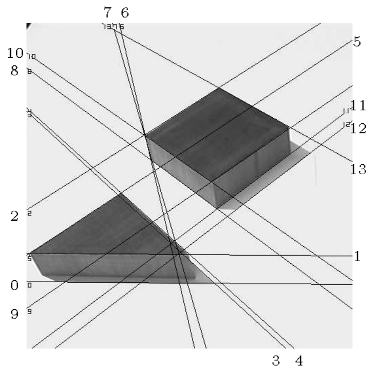
der Tripel von Indizes korrespondierender Bildgeraden. Für jedes Tripel wird dann durch das im nächsten Abschnitt beschriebene Rekonstruktionsverfahren eine 3D-Gerade berechnet. Um den Zusammenhang zwischen Tripeln und 3D-Geraden einfacher darstellen zu können, werden die Tripel mit einem Index versehen:

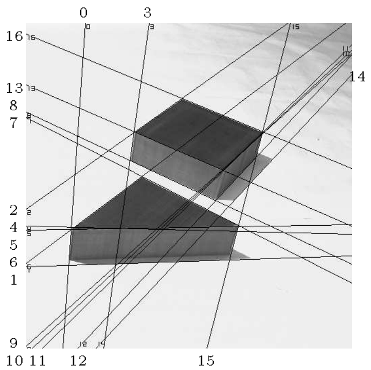
$$K^{1,2,3} = \{t_0, \ldots, t_{n-1}\} \qquad \text{mit} \quad n = |K^{1,2,3}| \tag{5.36}$$

5.2.4 Beispiel

Die Tabellen 5.2 und 5.3 zeigen das Ergebnis des Matchingverfahrens für eine einfache Szene, bei der in den drei Ansichten relativ wenige Geraden detektiert werden. Die Abbildungen Tab. 5.1(a)-5.1(c) zeigen die Grauwertbilder der drei Ansichten, in denen jeweils die detektierten Bildgeraden eingezeichnet sind. Die darunter befindlichen Abbildungen (Tab. 5.1(d)-5.1(f)) visualisieren die in $\mathcal{P}$ gespeicherten Kantenpunkte auf den detektierten Geraden. Diese Kantenpunkte decken eine Objektkante teilweise nur sehr lückenhaft ab. Die Konsequenz sind, wie in Kapitel 5.2.3.1 beschrieben wurde, fehlende oder falsche Punktkorrespondenzen während des Matchings der Geraden. Außerdem werden an einigen Objektkanten mehrere Geraden detektiert. Beispiele sind die Geraden $g_4^{(1)}$ und $g_5^{(1)}$ in der ersten Ansicht sowie die Geraden $g_6^{(2)}$ und $g_7^{(2)}$ in der zweiten Ansicht an der linken Kante des viereckigen Objektes. An der rechten Kante desselben Objektes wurden in der dritten Ansicht die Geraden $g_9^{(3)}$, $g_{10}^{(3)}$ und $g_{11}^{(3)}$ detektiert. Zu der Gerade $g_{15}^{(3)}$ in der dritten Ansicht existieren keine entsprechenden Geraden in den anderen Ansichten, gleiches gilt für $g_3^{(2)}$ in

<table>
<tr><th>1. Ansicht</th><th>2. Ansicht</th><th>3. Ansicht</th></tr>
<tr>
<td>

(a)</td>
<td>

(b)</td>
<td>

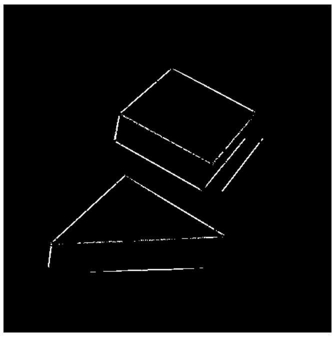
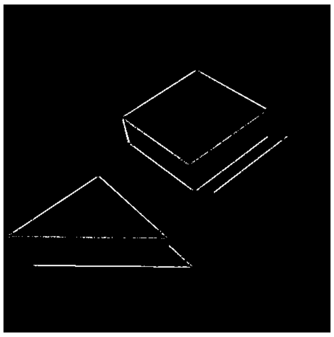
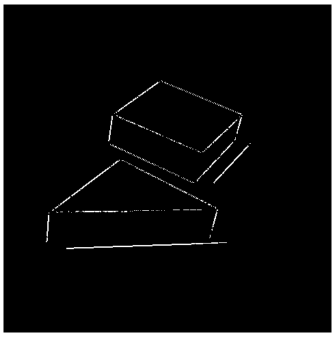
(c)</td>
</tr>
<tr>
<td>
(d)</td>
<td>
(e)</td>
<td>
(f)</td>
</tr>
</table>

Tabelle 5.1: Ansichten mit detektierten Geraden und Kantenpunkten

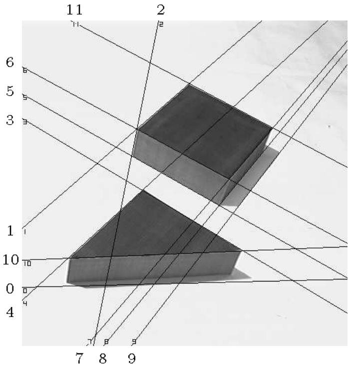
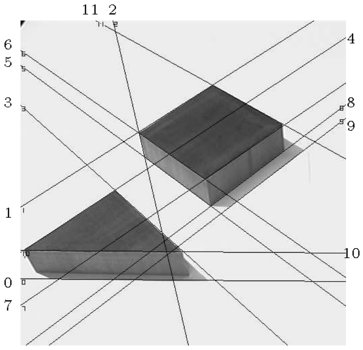
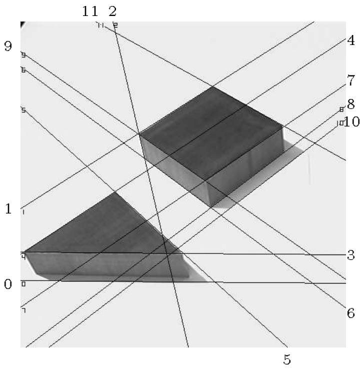
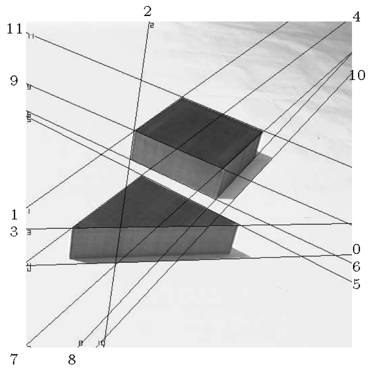
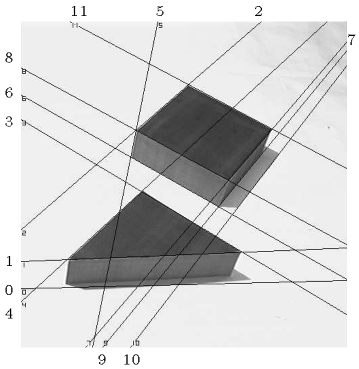
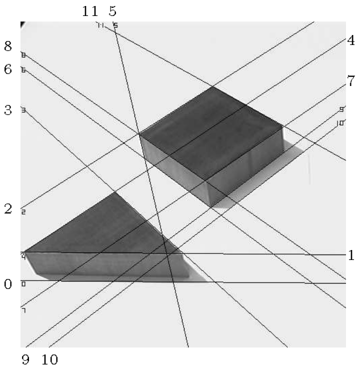
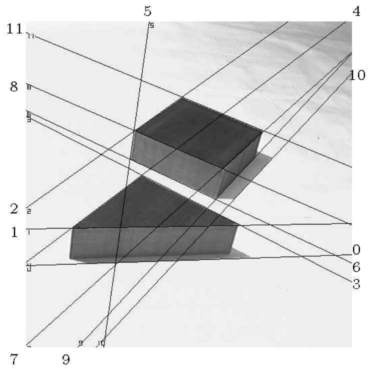
1. Ansicht	2. Ansicht	3. Ansicht
Korrespondenzen von 1. und 2. Ansicht		
 (a)	 (b)	
	Korrespondenzen von 2. und 3. Ansicht	
	 (c)	 (d)
Korrespondenzen über alle Ansichten		
 (e)	 (f)	 (g)

Tabelle 5.2: Darstellung der Korrespondenzen in den Ansichten

i	t_i
0	(1,0)
1	(3,2)
2	(4,6)
3	(6,4)
4	(7,5)
5	(8,8)
6	(9,10)
7	(10,9)
8	(12,11)
9	(13,12)
10	(14,1)
11	(15,13)

(a) $K^{1,2} = \{t_0, \ldots, t_{11}\}$

i	t_i
0	(0,1)
1	(2,2)
2	(6,3)
3	(1,5)
4	(5,6)
5	(3,7)
6	(8,8)
7	(9,9)
8	(11,12)
9	(10,13)
10	(12,14)
11	(13,16)

(b) $K^{2,3} = \{t_0, \ldots, t_{11}\}$

i	t_i
0	(1,0,1)
1	(14,1,5)
2	(3,2,2)
3	(6,4,7)
4	(7,5,6)
5	(4,6,3)
6	(8,8,8)
7	(10,11,12)
8	(9,10,13)
9	(12,11,12)
10	(13,12,14)
11	(15,13,16)

(c) $K^{1,2,3} = \{t_0, \ldots, t_{11}\}$

Tabelle 5.3: Korrespondenzpaare und -tripel

der zweiten Ansicht.

Dennoch liefert das Verfahren richtige Korrespondenzen von Geraden. Tabelle 5.3(a) enthält die Korrespondenzpaare der ersten und zweiten Ansicht. Die Komponenten der Paare sind die Indizes der Geraden in der ersten bzw. zweiten Ansicht. In Tabelle 5.2(a) und 5.2(b) markiert der Index i eines Paares t_i jeweils korrespondierende Geraden. Entsprechend zeigen Tabelle 5.3(b) und die Bilder in Tabelle 5.2(c) und 5.2(d) das Ergebnis für das Matching zwischen der zweiten und dritten Ansicht. Die Tripel in $K^{1,2,3}$ ergeben sich aus den Paaren mit gleicher Bildgerade in der zweiten Ansicht (Tab. 5.3(c), Tab. 5.2(e)-5.2(g)).

5.3 Rekonstruktion der Geraden

Gegenstand der Rekonstruktion von Geraden ist die Gewinnung der 3D-Geraden aus den korrespondierenden Bildgeraden in den Ansichten. Zur Rekonstruktion wird das in Faugeras [9] vorgeschlagene Verfahren verwendet.

Während ein Punkt und eine Ebene im projektiven Raum $\mathcal{P}^3$ durch ein Quadrupel definiert werden, deren Komponenten nicht alle gleich Null sind, läßt sich eine Gerade im projektiven Raum nur auf zwei kompliziertere Arten darstellen: entweder als Menge zweier linear abhängiger Punkte oder als Schnitt zweier Ebenen. Das Rekonstruktionsverfahren basiert auf der zuletzt genannten Möglichkeit. Um die Anzahl der gesuchten Parameter beider Ebenen möglichst klein zu halten, ist jedoch die Betrachtung von 3 Fällen notwendig.

1. Geraden, die nicht parallel zur xy-Ebene sind:

Die Punkte mit den Koordinaten (x, y, z) auf einer Geraden erfüllen für die gesuchten Parameter a, b, p, q das Gleichungssystem

$$\left\{ \begin{array}{l} x = az + p \\ y = bz + q \end{array} \right. \tag{5.37}$$

Die Punkt-Richtungs-Form dieser Geraden ist

$$\boldsymbol{g}(\lambda) = (p, q, 0)^T + \lambda(a, b, 1)^T \tag{5.38}$$

woraus deutlich wird, daß diese Gerade nicht parallel zur xy-Ebene sein kann. Die beiden Gleichungen von (5.37) beschreiben jeweils Ebenen L_1 und L_2, deren projektive Darstellungen

$$\tilde{\boldsymbol{L}}_1 = (1, 0, -a, -p)^T \qquad \text{und} \qquad \tilde{\boldsymbol{L}}_2 = (0, 1, -b, -q)^T \tag{5.39}$$

sind.

2. Geraden, die nicht parallel zur yz-Ebene sind:
   Die Punkte (x, y, z) auf einer Geraden erfüllen das Gleichungssystem

$$\left\{ \begin{array}{l} y = ax + p \\ z = bx + q \end{array} \right. \tag{5.40}$$

   Die Punkt-Richtungs-Form dieser Geraden ist

$$\boldsymbol{g}(\lambda) = (0, p, q)^T + \lambda(1, a, b)^T. \tag{5.41}$$

   Die Ebenen haben die projektiven Darstellungen

$$\tilde{\boldsymbol{L}}_1 = (-a, 1, 0, -p)^T \qquad \text{und} \qquad \tilde{\boldsymbol{L}}_2 = (-b, 0, 1, -q)^T \tag{5.42}$$

3. Geraden, die nicht parallel zur zx-Ebene sind:
   Die Punkte (x, y, z) erfüllen das Gleichungssystem

$$\left\{ \begin{array}{l} z = ay + p \\ x = by + q \end{array} \right. \tag{5.43}$$

   Die Punkt-Richtungs-Form dieser Geraden ist

$$\boldsymbol{g}(\lambda) = (q, 0, p)^T + \lambda(b, 1, a)^T. \tag{5.44}$$

   Die Ebenen haben die projektiven Darstellungen

$$\tilde{\boldsymbol{L}}_1 = (0, -a, 1, -p)^T \qquad \text{und} \qquad \tilde{\boldsymbol{L}}_2 = (1, -b, 0, -q)^T \tag{5.45}$$

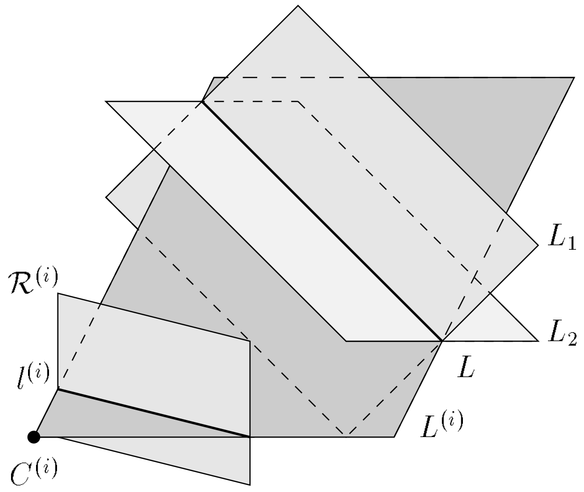


Abbildung 5.11: Bündel von Geraden

Es wird zunächst die Situation an einer Kamera betrachtet. Abbildung 5.11 veranschaulicht die nun folgenden Überlegungen. Ist $l^{(i)}$ die Gerade, die durch Projektion der 3D-Gerade L auf die Bildebene $\mathcal{R}^{(i)}$ entsteht, so gilt für alle Punkte m auf der projizierten Geraden die Gleichung $\tilde{\boldsymbol{l}}^{(i)}\tilde{\boldsymbol{m}} = 0$. Da m die Projektion eines Punktes M auf der Geraden L ist, gilt $\tilde{\boldsymbol{l}}^{(i)T}\tilde{\boldsymbol{P}}^{(i)}\tilde{\boldsymbol{M}} = 0$. Somit gilt $\tilde{\boldsymbol{P}}^{(i)T}\tilde{\boldsymbol{l}}^{(i)T}\tilde{\boldsymbol{M}} = 0$, was die Punkt-Ebenen-Gleichung der durch das optische Zentrum $C^{(i)}$ und der Geraden $l^{(i)}$ definierten Ebene $L^{(i)}$ ist. Also folgt für den $E^{(i)}$ repräsentierenden projektiven Vektor $\tilde{\boldsymbol{L}}^{(i)}$:

$$\tilde{\boldsymbol{L}}^{(i)} = \tilde{\boldsymbol{P}}^{(i)T}\tilde{\boldsymbol{l}}^{(i)} \tag{5.46}$$

Gesucht sind $\tilde{\boldsymbol{L}}_1$ und $\tilde{\boldsymbol{L}}_2$ gemäß (5.39), (5.42) oder (5.45), je nachdem, welche Darstellung geeignet ist. Es sei zunächst angenommen, daß sich die Gerade durch $\tilde{\boldsymbol{L}}_1$ und $\tilde{\boldsymbol{L}}_2$ in (5.39) repräsentieren läßt. Im folgenden wird die Matrix $\left(\tilde{\boldsymbol{L}}_1\,\tilde{\boldsymbol{L}}_2\,\tilde{\boldsymbol{L}}^{(i)}\right)$ betrachtet. Diese hat mit $\tilde{\boldsymbol{L}}^{(i)} = (\alpha^{(i)}, \beta^{(i)}, \gamma^{(i)}, \delta^{(i)})^T$ die Form

$$\begin{pmatrix} 1 & 0 & \alpha^{(i)} \\ 0 & 1 & \beta^{(i)} \\ -a & -b & \gamma^{(i)} \\ -p & -q & \delta^{(i)} \end{pmatrix} \tag{5.47}$$

Die durch $\tilde{\boldsymbol{L}}_1$, $\tilde{\boldsymbol{L}}_2$ und $\tilde{\boldsymbol{L}}^{(i)}$ dargestellten Ebenen L_1, L_2 und $L^{(i)}$ bilden ein Bündel. Nach dem Dualitätsprinzip ist dies äquivalent dazu, daß die durch $\tilde{\boldsymbol{L}}_1$, $\tilde{\boldsymbol{L}}_2$ und $\tilde{\boldsymbol{L}}^{(i)}$ dargestellten Punkte des $\mathcal{P}^3$ auf einer Geraden liegen, also linear abhängig sind. Daher ist der Rang der Matrix (5.47) kleiner oder gleich 2. Somit ist die Determinante jeder 3×3-Untermatrix von (5.47) gleich 0. Die 4 Determinanten sind

$$D_1 = a\alpha^{(i)} + b\beta^{(i)} + \gamma^{(i)} \tag{5.48}$$

$$D_2 = p\alpha^{(i)} + q\beta^{(i)} + \delta^{(i)} \tag{5.49}$$

$$D_3 = -b\delta^{(i)} + q\gamma^{(i)} + \alpha^{(i)}(aq - bp) \quad = qD_1 - bD_2 \tag{5.50}$$

$$D_4 = a\delta^{(i)} - p\gamma^{(i)} + \beta^{(i)}(aq - pb) \quad = -pD_1 + aD_2 \tag{5.51}$$

Da D_3 und D_4 Linearkombinationen von D_1 und D_2 sind, ist notwendig und hinreichend, daß D_1 und D_2 gleich 0 sind:

$$0 = a\alpha^{(i)} + b\beta^{(i)} + \gamma^{(i)} \tag{5.52}$$

$$0 = p\alpha^{(i)} + q\beta^{(i)} + \delta^{(i)} \tag{5.53}$$

Ist $\mathcal{L} = (a, b, p, q)^T$ der Vektor der zu schätzenden Parameter, so ist mit

$$\boldsymbol{A}^{(i)} = \begin{pmatrix} \alpha^{(i)} & \beta^{(i)} & 0 & 0 \\ 0 & 0 & \alpha^{(i)} & \beta^{(i)} \end{pmatrix} \quad \text{und} \quad \boldsymbol{b}^{(i)} = \begin{pmatrix} -\gamma^{(i)} \\ -\delta^{(i)} \end{pmatrix} \tag{5.54}$$

das Gleichungssystem (5.52) äquivalent zu

$$\boldsymbol{A}^{(i)}\mathcal{L} = \boldsymbol{b}^{(i)} \tag{5.55}$$

Für n Kameras ergibt sich mit

$$\mathcal{A} = \begin{pmatrix} \boldsymbol{A}^{(1)} \\ \vdots \\ \boldsymbol{A}^{(n)} \end{pmatrix} \quad \text{und} \quad \mathcal{B} = \begin{pmatrix} \boldsymbol{b}^{(1)} \\ \vdots \\ \boldsymbol{b}^{(n)} \end{pmatrix} \tag{5.56}$$

das Gleichungssystem

$$\mathcal{A}\mathcal{L} = \mathcal{B} \tag{5.57}$$

dessen Lösung im Sinne des kleinsten quadratischen Fehlers gegeben ist durch

$$\mathcal{L} = (\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T\mathcal{B} \tag{5.58}$$

Daraus wird dann die Punkt-Richtungs-Form der Geraden gemäß (5.44) bestimmt.

Damit das Gleichungssystem (5.57) lösbar ist, müssen $n \geq 3$ Kameras verwendet werden, bei denen die durch die optischen Zentren definierte 3D-Ebene die Bildebenen der Kameras nicht gleichzeitig schneidet. Werden nur 2 Kameras verwendet, so kann die 4×4-Matrix $\mathcal{A}$ in (5.56) singulär werden. Zur Erläuterung sei eine beliebige 3D-Ebene $L^{(1,2)}$, auf der die optischen Zentren der Kameras liegen und welche die Bildebenen $\mathcal{R}^{(1)}$ und $\mathcal{R}^{(2)}$ schneiden, gegeben (siehe Abbildung 5.12). Alle 3D-Geraden, die in der Ebene $L^{(1,2)}$ liegen, werden auf dieselben 2D-Geraden l_1 und l_2 in beiden Bildebenen $\mathcal{R}^{(1)}$ und $\mathcal{R}^{(2)}$ projiziert. Somit

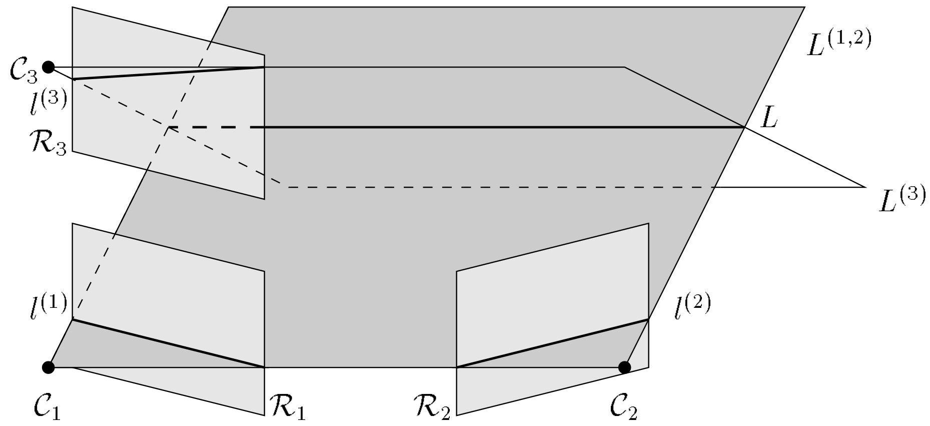


Abbildung 5.12: Motivation für drei Kameras

ergibt die Rückprojektion von l_1 und l_2 in den Raum keine durch $\mathcal{L}$ eindeutig bestimmte 3D-Gerade. Durch Hinzunahme der dritten Kamera erzeugt eine beliebige, in der Ebene $L^{(1,2)}$ befindliche 3D-Gerade L eine Projektion $l^{(3)}$ auf $\mathcal{R}^{(3)}$. Durch Rückprojektion der 2D-Gerade $l^{(3)}$ in den Raum ergibt sich eine Ebene $L^{(3)}$, die aufgrund der an die optischen Zentren der 3 Kameras gestellten Bedingungen nicht identisch ist. Die 3D-Gerade L ist dann der Schnitt der Ebenen $L^{(1,2)}$ und $L^{(3)}$. Die Matrix $\mathcal{A}$ ist also regulär.

Im folgenden seien noch kurz die Ergebnisse für den 2. und 3. Fall zusammengefaßt. Für Geraden, die nicht parallel zur yz-Ebene verlaufen, ergeben sich anstelle von (5.48)-(5.51) die Unterdeterminanten

$$D_1 = a\beta^{(i)} + b\gamma^{(i)} + \alpha^{(i)} \tag{5.59}$$
$$D_2 = -aq\beta^{(i)} + b(\delta^{(i)} + pb) - \alpha^{(i)}q \qquad = -qD_1 + bD_4 \tag{5.60}$$
$$D_3 = -a(\delta^{(i)} + q\gamma^{(i)}) + bp\gamma^{(i)} + p\alpha^{(i)} \qquad = pD_1 - aD_4 \tag{5.61}$$
$$D_4 = \delta^{(i)} + q\gamma^{(i)} + p\beta^{(i)} \tag{5.62}$$

Anstelle von (5.54) ergibt sich

$$\boldsymbol{A}^{(i)} = \begin{pmatrix} \beta^{(i)} & \gamma^{(i)} & 0 & 0 \\ 0 & 0 & \beta^{(i)} & \gamma^{(i)} \end{pmatrix} \qquad \text{und} \qquad \boldsymbol{b}^{(i)} = \begin{pmatrix} -\alpha^{(i)} \\ -\delta^{(i)} \end{pmatrix} \tag{5.63}$$

Die Punkt-Richtungs-Form ist durch (5.41) gegeben.

Für Geraden, die nicht parallel zur zx-Ebene verlaufen, ergibt sich

$$D_1 = a\gamma^{(i)} + \beta^{(i)} + b\alpha^{(i)} \tag{5.64}$$
$$D_2 = a\delta^{(i)} - p\beta^{(i)} + aq\alpha^{(i)} - bp\alpha^{(i)} \qquad = -pD_1 - aD_3 \tag{5.65}$$
$$D_3 = -\delta^{(i)} - p\gamma^{(i)} - q\alpha^{(i)} \tag{5.66}$$
$$D_4 = -aq\gamma^{(i)} + b(\delta^{(i)} + p\gamma^{(i)}) - q\beta^{(i)} \qquad = -qD_1 + bD_3 \tag{5.67}$$

sowie

$$\boldsymbol{A}^{(i)} = \begin{pmatrix} \delta^{(i)} & \alpha^{(i)} & 0 & 0 \\ 0 & 0 & \delta^{(i)} & \alpha^{(i)} \end{pmatrix} \qquad \text{und} \qquad \boldsymbol{b}^{(i)} = \begin{pmatrix} -\beta^{(i)} \\ -\delta^{(i)} \end{pmatrix} \tag{5.68}$$

Die Punkt-Richtungs-Form ist durch (5.44) gegeben.

Für jedes Tripel $(i^{(1)}, i^{(2)}, i^{(3)}) \in K^{1,2,3}$ wird also aus $\tilde{\boldsymbol{l}}^{(1)}_{i^{(1)}}$, $\tilde{\boldsymbol{l}}^{(2)}_{i^{(2)}}$ und $\tilde{\boldsymbol{l}}^{(3)}_{i^{(3)}}$ eine 3D-Gerade in Punkt-Richtungsform rekonstruiert. Da nicht a priori festgelegt werden kann, welche der 3 Fälle zutrifft, müssen für alle Fälle die Matrizen $\mathcal{A}$ und $\mathcal{B}$ in Gleichung (5.56) berechnet werden. Die 3D-Gerade läßt sich in einer Darstellung genau dann repräsentieren, wenn das Gleichungssystem (5.57) lösbar ist. Unter den Darstellungen wird die Lösung $\mathcal{L}$ gewählt, für den der Fehler $\|\mathcal{A}\mathcal{L} - \mathcal{B}\|$ minimal ist.

Das Ergebnis der Rekonstruktion der Geraden ist also eine Menge

$$^{3\mathrm{D}}L = \{\boldsymbol{g}_0, \dots, \boldsymbol{g}_{n-1}\} \tag{5.69}$$

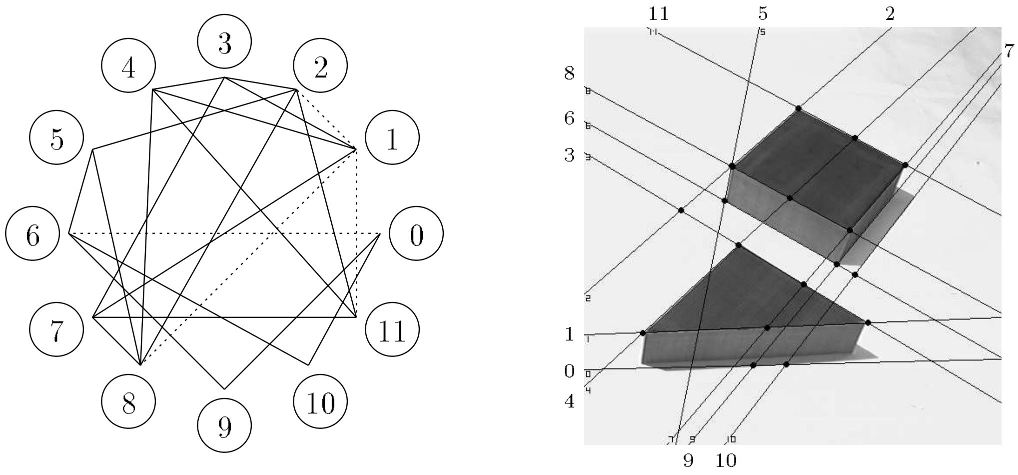
von Geraden in Punkt-Richtungs-Form.

5.4 Geradenbasierte Beschreibung der Szene durch Graphen

Nachdem die 3D-Geraden rekonstruiert wurden, sind sich schneidende 3D-Geraden von Interesse. Jeder Eckpunkt eines Objektes ist – sofern die Berandungsgeraden detektiert wurden – der Schnittpunkt zweier 3D-Geraden. Die für die Objekterkennung relevante, umgekehrte Richtung gilt nicht. Im allgemeinen gibt es Schnittpunkte, die nicht Eckpunkte eines Objektes sind (siehe z. B. Abb. 5.13(b)). Es gilt, diese Schnittpunkte im weiteren Verlauf des Verfahrens auszuschließen. Zunächst ist jedoch nicht notwendig, die Schnittpunkte explizit zu berechnen, es genügt die Information, welche Geraden durch einen Schnittpunkt in Beziehung stehen. Daher ist die Repräsentation der Szene durch Graphen naheliegend, dessen Knotenmenge der Menge von Geraden und dessen Kantenmenge Paare sich schneidender Geraden entspricht.

Ein ungerichteter, endlicher Graph G ist definiert als ein Paar (V, K), wobei V eine endliche Menge ist und $E \subseteq \{\{v, u\} \mid \{u, v\} \subseteq V\}$. Die Menge V ist die Menge der Knoten und K die Menge der Kanten.

Der eine Szene beschreibende Graph ist schlingenfrei, V enthält nur zweielementige Mengen. Es ist $V = \{0, \dots, n-1\}$, wobei mit $i \in V$ die durch das Tripel $t_i \in K^{1,2,3}$ beschriebene 3D-Gerade ist. Abbildung 5.13 zeigt das Bild der Beispielszene mit dem dazugehörigen Graphen.



(a) (b)

Abbildung 5.13: Szenengraph zu den aus den drei Ansichten der Beispielszene rekonstruierten Geraden. Das Bild zeigt die erste Ansicht. Die Kanten des Graphen, denen außerhalb des Bildes liegende Schnittpunkte entsprechen, sind gestrichelt dargestellt.

5.5 Berechnung des Graphen

Der Extraktion der polygonalen Oberflächen geht insbesondere die Suche von Kreisen bestimmter Länge im Graphen voraus. Diese Operation ist sehr teuer und hängt insbesondere von der Anzahl der Kanten im Graphen ab. Daher ist es sinnvoll, nur bestimmte Geradenkonfigurationen zuzulassen, sowie 3D-Schnittpunkte auszuschließen, zu denen keine Projektion in den Ansichten existiert.

5.5.1 Konfigurationen von Geraden

Zwei 3D-Geraden können auf zwei verschiedene Arten in Beziehung stehen, entweder es existiert ein Schnittpunkt oder nicht. Da die zu erkennenden Objektoberflächen jedoch meistens spezielle, geometrische Eigenschaften haben, ist es sinnvoll, nur solche Geradenbeziehungen zu betrachten, die die Berandungsgeraden des gesuchten Objektes auch aufweisen. Durch Einfluß dieses a priori Wissens läßt sich der 3D-Graph erheblich reduzieren. Weist das Objekt beispielsweise nur rechtwinklige Ecken auf, so braucht man nur Schnittpunkte zuzulassen, an denen der Winkel 90° beträgt. Setzt sich das Objekt z. B. aus Paaren paralleler Geraden zusammen, so brauchen nur Geraden betrachtet werden, die diese Eigenschaft bezüglich anderer Geraden besitzen. Alle Entscheidungen müssen innerhalb einer Fehlertoleranz erfolgen. Für Abstände bezeichne θ_d die zulässige absolute Abweichung, für Winkel sei entsprechend $\theta_\angle$ definiert.

Es seien $\boldsymbol{g}_1(\lambda) = \boldsymbol{a}_1 + \boldsymbol{u}_1$ und $\boldsymbol{g}_2(\lambda) = \boldsymbol{a}_2 + \boldsymbol{u}_2$ zwei 3D-Geraden in Punkt-Richtungs-Form.

1. *Abstand der Geraden*:
   Sind $\boldsymbol{u}_1$ und $\boldsymbol{u}_2$ linear unabhängig, so ist der Abstand der beiden Geraden gleich dem Abstand der durch

   $$\boldsymbol{g}_i(\lambda, \mu) = \boldsymbol{a}_i + \lambda \boldsymbol{u}_1 + \mu \boldsymbol{u}_2, \qquad i \in \{1, 2\} \tag{5.70}$$

   definierten Ebenen. Die Hesse-Normalenformen der Ebenen sind

   $$d_i = \boldsymbol{n}\boldsymbol{x}, \qquad i \in \{1, 2\} \tag{5.71}$$

   mit dem Normaleneinheitsvektor

   $$\boldsymbol{n} = \frac{\boldsymbol{u}_1 \times \boldsymbol{u}_2}{\|\boldsymbol{u}_1 \times \boldsymbol{u}_2\|} \tag{5.72}$$

   Der Abstand der Ebenen – und somit der Geraden – erhält man aus

   $$d(\boldsymbol{g}_1, \boldsymbol{g}_2) = |d_1 - d_2| = |\boldsymbol{n}\boldsymbol{a}_1 - \boldsymbol{n}\boldsymbol{a}_2| \tag{5.73}$$

   Sind $\boldsymbol{u}_1$ und $\boldsymbol{u}_2$ linear abhängig, so liefert (5.73) den Abstand der parallelen Geraden für

   $$\boldsymbol{n} = \begin{cases} \frac{\boldsymbol{u}_1 \times ((\boldsymbol{a}_2 - \boldsymbol{a}_1) \times \boldsymbol{u}_1)}{\|(\boldsymbol{a}_2 - \boldsymbol{a}_1) \times \boldsymbol{u}_1\|} & \text{, falls } \|\boldsymbol{u}_1 \times ((\boldsymbol{a}_2 - \boldsymbol{a}_1) \times \boldsymbol{u}_1)\| \neq 0 \\ \boldsymbol{0} & \text{, sonst} \end{cases} \tag{5.74}$$

   Der erste Fall liegt vor, wenn die parallelen Geraden sich nicht schneiden und somit in einer Ebene liegen. Der Vektor $\boldsymbol{n}$ liegt dann in dieser Ebene und senkrecht zu den Richtungsvektoren der beiden Geraden.

2. *Winkel zwischen den Geraden*: Der Winkel zwischen beiden Geraden ist gegeben durch

   $$\angle(\boldsymbol{g}_1, \boldsymbol{g}_2) = |\arcsin(\|\boldsymbol{u}_1 \times \boldsymbol{u}_2\|)| \tag{5.75}$$

Die Geraden $\boldsymbol{g}_1$ und $\boldsymbol{g}_2$ schneiden sich, falls

$$d(\boldsymbol{g}_1, \boldsymbol{g}_2) \leq \theta_d \tag{5.76}$$

Sie besitzen den Winkel α, falls

$$|\angle(\boldsymbol{g}_1, \boldsymbol{g}_2) - \alpha| \leq \theta_\angle \tag{5.77}$$

für die zwei a priori festzulegenden Toleranzschwellen θ_d und $\theta_\angle$.

Die Kodierung des a priori Wissens für Geradenkonfigurationen geschieht folgendermaßen:

1. *Konfiguration sich schneidender Geraden*: Die das Objekt charakterisierenden Winkel sind durch die Menge

$$A = \{\alpha_1, \dots, \alpha_n\} \tag{5.78}$$

gegeben. Um auch beliebige Winkel zuzulassen, sei ein spezielles Element „beliebig“ eingeführt: $\alpha_i \in \mathbb{R} \cup \{\text{beliebig}\}$.

2. *Konfiguration sich nicht schneidender Geraden*: Es werden nur Geraden betrachtet, die mit einer anderen Gerade durch bestimmte Winkel und Abstände in Beziehung stehen. Die Winkel und Abstände der Geraden sind durch die Menge

$$B = \{(\beta_1, d_1), \dots, (\beta_m, d_m)\} \tag{5.79}$$

festgelegt, mit $\beta_i, d_i \in \mathbb{R} \cup \{\text{beliebig}\}$.

Der Szenengraph (V, K) sei durch seine Adjazenzmatrix repräsentiert, die mit $\boldsymbol{S}$ bezeichnet sei. Gemäß dem ersten Kriterium ergibt sich für die Komponenten von $\boldsymbol{S}$:

$$S_{i\,j} := \begin{cases} 1 & \text{, falls } d(\boldsymbol{g}_i, \boldsymbol{g}_j) \le \theta_d \wedge \exists \alpha \in A : \alpha = \text{beliebig} \vee |\angle(\boldsymbol{g}_i, \boldsymbol{g}_j) - \alpha| \le \theta_\angle \\ 0 & \text{, sonst} \end{cases} \tag{5.80}$$

für alle Geraden $\boldsymbol{g}_{i_1}, \boldsymbol{g}_{i_2} \in {}^{3\mathrm{D}}L$. Danach werden Geraden entfernt, die mit keiner anderen Geraden in der durch (5.79) bestimmten Beziehung stehen:

$$\begin{aligned} S_i := \boldsymbol{0} \quad & \text{falls } \forall (\alpha, d) \in B \neg \exists \boldsymbol{g}_j \in {}^{3\mathrm{D}}L : \\ & (\alpha = \text{beliebig} \vee |\angle(\boldsymbol{g}_i, \boldsymbol{g}_j) - \beta| \le \theta_\angle) \wedge (d = \text{beliebig} \vee |d(\boldsymbol{g}_i, \boldsymbol{g}_j) - d| \le \theta_d) \end{aligned} \tag{5.81}$$

Dabei bezeichnet S_i sowohl die i-te Zeile als auch die i-te Spalte der Adjazenzmatrix $\boldsymbol{S}$.

5.5.2 Verifikation der Schnittpunkte in den Ansichten

Der Szenengraph läßt sich weiter reduzieren, indem die Kanten entfernt werden, denen außerhalb des Bildes liegende Schnittpunkte entsprechen. Sind ${}^{2\mathrm{D}}\boldsymbol{g}_i$ und ${}^{2\mathrm{D}}\boldsymbol{g}_j$ die Bildgeraden einer Ansicht, so erhält man den Schnittpunkt der beiden Geraden durch Gleichsetzen der Punkt-Richtungs-Formen ${}^{2\mathrm{D}}\boldsymbol{g}_i(\lambda_{i\,j}) = {}^{2\mathrm{D}}\boldsymbol{g}_j(\lambda_{j\,i})$ (siehe (5.18)). Daraus folgt unmittelbar $\begin{pmatrix} \boldsymbol{u}_i & -\boldsymbol{u}_j \end{pmatrix} \begin{pmatrix} \lambda_{i\,j} \\ \lambda_{j\,i} \end{pmatrix} = \boldsymbol{a}_i - \boldsymbol{a}_j$ und somit

$$\begin{pmatrix} \lambda_{i\,j} \\ \lambda_{j\,i} \end{pmatrix} = \begin{pmatrix} \boldsymbol{u}_i & -\boldsymbol{u}_j \end{pmatrix}^{-1} (\boldsymbol{a}_i - \boldsymbol{a}_j) = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}^{-1} (\boldsymbol{a}_i - \boldsymbol{a}_j) = \frac{1}{\alpha\delta - \beta\gamma} \begin{pmatrix} \delta & -\beta \\ -\gamma & \alpha \end{pmatrix} \tag{5.82}$$

falls die Matrix $\begin{pmatrix} \boldsymbol{u}_i & -\boldsymbol{u}_j \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$ invertierbar ist.

Jede Gerade hat zwei Schnitte mit dem Rand des Bildes. Definiert man 4 Geraden für den oberen, unteren, linken und rechten Rand, so schneidet eine Bildgerade zwei dieser Bildgeraden. Die $\nu_{i\,k_1}$ und $\nu_{i\,k_2}$, die sich anstelle von $\lambda_{i\,j}$ und $\lambda_{j\,i}$ aus (5.82) ergeben, haben die Eigenschaft, daß

$$\nu_{i\,k_1} \leq \lambda_{i\,j} \leq \nu_{i\,k_2} \quad \text{für alle} \quad j \in \{0, \dots, n-1\} \tag{5.83}$$

Aufgrund dieses Kriteriums läßt sich bestimmen, welcher Schnittpunkt im Bild liegt. Es werden nur Schnitte betrachtet, die innerhalb des Bildes liegen: $\lambda_{i\,j}$ und $\lambda_{j\,i}$ sind undefiniert, falls die Determinante $\alpha\delta - \beta\gamma$ in (5.82) Null ist oder außerhalb des Intervalls $[\nu_{i\,k_1}, \nu_{i\,k_2}]$ liegen. Die Lösungen lassen sich in der Matrix

$$\boldsymbol{\Lambda}^{(a)} = \begin{pmatrix} \kappa_{0\,0}^{(a)} & \dots & \kappa_{0\,3}^{(a)} & \mu_{0\,0}^{(a)} & \dots & \mu_{0\,n}^{(a)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \kappa_{3\,0}^{(a)} & \dots & \kappa_{3\,3}^{(a)} & \mu_{3\,0}^{(a)} & \dots & \mu_{1\,n}^{(a)} \\ \nu_{0\,0}^{(a)} & \dots & \nu_{0\,3}^{(a)} & \lambda_{0\,0}^{(a)} & \dots & \lambda_{0\,n}^{(a)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \nu_{n\,0}^{(a)} & \dots & \nu_{n\,3}^{(a)} & \lambda_{n\,0}^{(a)} & \dots & \lambda_{n\,n}^{(a)} \end{pmatrix} \tag{5.84}$$

zusammenfassen. Es beschreibt $\kappa_{i\,j}$ den Schnitt der i-ten Bildrandgeraden mit der j-ten Bildrandgeraden, $\mu_{i\,j}$ den Schnitt der i-ten Bildrandgeraden mit der j-ten Bildgeraden, $\nu_{i\,j}$ den Schnitt der i-ten Bildgeraden mit der j-ten Bildrandgeraden, $\lambda_{i\,j}$ den Schnitt der i-ten Bildgeraden mit der j-ten Bildgeraden. Sind $\boldsymbol{t}_i = (\iota_i^{(1)}, \iota_i^{(2)}, \iota_i^{(3)})$ und $\boldsymbol{t}_j = (\iota_j^{(1)}, \iota_j^{(2)}, \iota_j^{(3)})$ die durch das Matchingverfahren (Kapitel 5.2.3.2) gewonnenen Tripel der Indizes zweier 3D-Geraden $\boldsymbol{g}_i$ und $\boldsymbol{g}_j$, so wird

$$S_{i\,j} := \begin{cases} 1 & \text{, falls } S_{i\,j} = 1 \,\wedge\, \lambda^{(a)}_{\iota_i^{(a)}\,\iota_j^{(a)}} \text{ ist definiert für } a = 1, \dots, 3 \\ 0 & \text{, sonst} \end{cases} \tag{5.85}$$

gesetzt und somit außerhalb einer Ansicht liegende Schnittpunkte entfernt.

5.5.3 Kreise im Szenengraphen

Die Objektoberfläche, zu dem sich das Kamerasystem ausrichten soll, ist durch die polygonale Berandung charakterisiert. Dieses Polygon wird repräsentiert durch eine Folge von 3D-Punkten $\boldsymbol{p}_0, \dots, \boldsymbol{p}_{n_P-1}$. Den Punkten dieses Polygons entsprechen Kanten des Szenengraphen. Zwei aufeinanderfolgende Polygonpunkte definieren eine Kante, die einer Gerade, also einem Knoten des Szenengraphen, entspricht. Somit geht der Berechnung der Polygone die Suche von Kreisen im Graphen voraus.

Ein Kreis der Länge m in einem Graphen ist üblicherweise definiert als ein geschlossener Kantenzug[2]

$$v_0, k_1, v_1, \ldots, k_m, v_m \tag{5.86}$$

mit $v_0, \ldots, v_m \in V$ und $k_1, \ldots, k_m \in K$, bei dem die Knoten paarweise verschieden sind.

Da der die Szene beschreibende Graph keine speziellen Eigenschaften hat, welche die Komplexität des Suchalgorithmus verringern könnte, ist eine erschöpfende Suche im Graphen notwendig. Von jedem Knoten ausgehend muß ein Kreis bestimmter Länge gesucht werden. Daher haben die Anzahl der Knoten und die Grade der Knoten alleinigen Einfluß auf die Komplexität. Ist $n = |V|$ die Anzahl der Knoten des Graphen, so ist der Grad eines Knotens maximal n, bei einer fest gewählten Anzahl n_P von Punkten der Polygonzüge ergibt sich mit $m = n_P + 1$ eine Komplexität von $O(n \cdot n^{m-1}) = O(n^m) = O(n^{n_P+1})$. Der Algorithmus ist also von polynomialer Komplexität.

Die Suche eines Kreises erfolgt durch eine rekursive Funktion. Dabei werden der bisher gefundene Kantenzug, die aktuelle Länge und die Information, welche Knoten bereits durchlaufen wurden, lokal bei jedem Funktionsaufruf gespeichert.

Sind $v_0, v_1, \ldots, v_{m-1}$ die Knoten eines Kreises im Graphen, so sind ebenfalls

$$\begin{array}{l} v_1, v_2, \ldots, v_{m-1}, v_0 \\ v_2, v_3, \ldots, v_0, v_1 \\ \vdots \\ v_{n-1}, v_0, \ldots, v_{m-3}, v_{m-2} \end{array}$$

Pfade von Kreisen, die in Bezug auf die gegebene Problemstellung äquivalent sind. Da die Knoten des Graphen durch die Indizes der Geraden, also ganze, natürliche Zahlen, repräsentiert werden, wird durch die Bedingung

$$v_0 \leq v_i \qquad \forall i \in \{0, \ldots, m-1\} \tag{5.87}$$

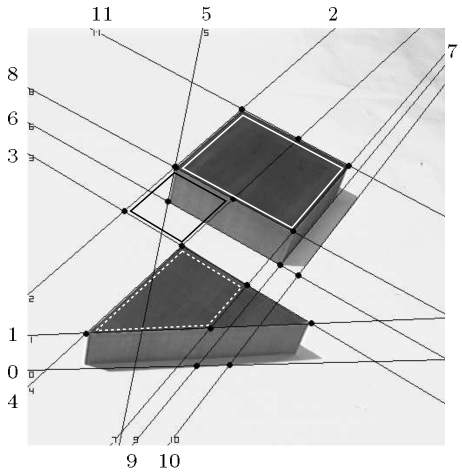
gewährleistet, daß genau ein Repräsentant der Äquivalenzklasse übrig bleibt. Außerdem ermöglicht das sequentielle Prüfen der Bedingung einen möglichst frühen Abbruch der Kreissuche.

Dennoch treten Berandungen noch doppelt auf, da nicht nur der Kreis $v_0, v_1, \ldots, v_m$, sondern auch der „gespiegelte“ Kreis $v_m, v_{m-1}, \ldots, v_0$ extrahiert wird. Auf dieses Problem wird später noch eingegangen.

Das Verfahren zur Suche von Kreisen im Graphen läßt sich parallelisieren, da die Kreise von jedem Knoten aus startend unabhängig voneinander berechnet werden können.

Abbildung 5.14 zeigt die in der Beispielszene gefundenen Kreise der Länge 4, ohne deren Spiegelungen.

[2]Ein Kantenzug ist eine Kantenfolge $k_1, k_2, \ldots, k_m$, bei der alle Kanten paarweise verschieden sind. Sind $\{v_0, \ldots, v_m\} \in V$ mit $k_i = (v_{i-1}, v_i)$, so heißt der Kantenzug geschlossen, falls $v_0 = v_m$.



0, 9, 6, 10, 0
1, 4, 3, 7, 1
1, 4, 8, 7, 1
1, 4, 11, 7, 1
2, 3, 4, 8, 2
2, 3, 4, 11, 2
2, 3, 7, 8, 2
2, 3, 7, 11, 2
2, 8, 4, 11, 2
2, 8, 7, 11, 2
3, 4, 8, 7, 3
3, 4, 11, 7, 3
4, 8, 7, 11, 4

Abbildung 5.14: Beispielszene und Kreise der Länge 4

5.6 Polygone

Nachdem die Kreise aus dem Graphen extrahiert wurden, müssen die durch die Kreise beschriebenen Polygone der Oberflächenberandungen berechnet werden. Ein Polygon besteht aus 3D-Punkten, von denen angenommen wird, daß sie in einer Ebene liegen.

Ein Polygon, das eine Berandung beschreibt, hat die Eigenschaft, daß sich seine Kanten nicht schneiden. Ein solches Polygon wird üblicherweise als einfaches Polygon bezeichnet. Die 2D-Projektion des planaren 3D-Polygons ist genau dann einfach, wenn das planare 3D-Polygon einfach ist. In der Bildebene läßt sich also bereits mit diesem Kriterium testen, ob die 3D-Konstruktion eine Objektberandung ist.

Für die weitere Objektmodellierung muß der Polygonzug eine bestimmte Orientierung aufweisen. Diese läßt sich durch die Orientierung der korrespondierenden 2D-Polygone festlegen, die in den Bildebenen der Ansichten gleich ist. Es werden nur Polygone selektiert, die im Koordinatensystem der Kameras in mathematisch positiver Orientierung, d. h. entgegen dem Uhrzeigersinn, orientiert sind. Auf diese Weise wird auch das im letzten Abschnitt angesprochene Problem, daß nicht nur ein Kreis, sondern auch seine Spiegelung vorliegt, gelöst.

Aus einem Kreis des Graphen lassen sich die 2D-Polygone berechnen, da mit den aufeinanderfolgenden Geradenkorrespondenzen auch die korrespondierenden Schnittpunkte bekannt sind. Durch Rekonstruktion der korrespondierenden 2D-Polygonpunkte erhält man die 3D-Punkte einer Objektoberfläche.

Die Berechnung eines 3D-Polygons aus einem Kreis geschieht also in folgenden Schritten:

1. Berechnung des 2D-Polygonzugs im linken oder rechten Bild.
2. Verifikation des 2D-Polygons:

2.1. Test, ob 2D-Polygon einfach ist. Wenn nicht, breche ab.
2.2. Test, ob das Polygon positiv orientiert ist. Wenn nicht, breche ab.

3. Berechnung des 3D-Polygons aus dem Pfad.

5.6.1 2D-Polygone

Sind $v_0, v_1, \ldots, v_m$ die Knoten eines Kreises im Graphen, so beschreiben $\boldsymbol{t}_{v_i} = (\iota_{v_i}^{(1)}, \iota_{v_i}^{(2)}, \iota_{v_i}^{(3)})$ und $\boldsymbol{t}_{v_{i+1}} = (\iota_{v_{i+1}}^{(1)}, \iota_{v_{i+1}}^{(2)}, \iota_{v_{i+1}}^{(3)})$ für $i = 0, \ldots, m-1$ die adjazenten Geraden in den drei Ansichten. Aus den Schnittpunkten ergeben sich die Eckpunkte ${}^{2\mathrm{D}}\boldsymbol{p}_i^{(a)}$, $i = 0, \ldots, m-1$ der Polygone in der jeweiligen Ansicht. Mit der abkürzenden Schreibweise $k = \iota_{v_i}^{(a)}$ und $l = \iota_{v_{i+1}}^{(a)}$ erhält man den Schnittpunkt $\boldsymbol{s}_{k\,l}^{(a)}$ zwischen der k-ten Bildgeraden und der l-ten Bildgeraden in der a-ten Ansicht mit

$$\boldsymbol{s}_{k\,l}^{(a)} = \begin{cases} \boldsymbol{a}_k + \lambda_{k\,l}\boldsymbol{u}_k & \text{, falls } \lambda_{k\,l} \text{ definiert ist} \\ \text{undefiniert} & \text{, sonst} \end{cases} \tag{5.88}$$

wobei $\lambda_{i\,j}$ die Komponenten der Matrix $\Lambda^{(a)}$ in (5.84) sind. Ist der Schnittpunkt undefiniert, so liegt das Polygon nicht im Bild.

5.6.2 Fläche und Orientierung eines 2D-Polygons

Sei mit δ_i das durch $\boldsymbol{p}_i$, $\boldsymbol{p}_{i-1}$ und $\boldsymbol{p}_0$, beschriebene Dreieck bezeichnet, mit A_{δ_i} die Fläche, deren Vorzeichen sich aus der Orientierung des Dreiecks ergibt. Durch Induktion über die Anzahl n_P der Punkte eines einfachen Polygons läßt sich zeigen:

$$A_P = \sum_{i=2}^{n_P-1} A_{\delta_i} \tag{5.89}$$

$$= \frac{1}{2}\sum_{i=2}^{n_P-1} \det\left(\boldsymbol{p}_{i-1} - \boldsymbol{p}_0 \quad \boldsymbol{p}_i - \boldsymbol{p}_{i-1} \right) \tag{5.90}$$

Das Vorzeichen der Fläche A_P gibt die Orientierung des Polygons an. Ist $A_P > 0$, so ist das Polygon in mathematisch positiver Orientierung, d. h. entgegen dem Uhrzeigersinn orientiert; gilt $A_P < 0$, so ist das Polygon mit dem Uhrzeigersinn orientiert.

5.6.3 3D-Polygone

Aus den korrespondierenden 2D-Polygonen wird ein 3D-Polygon rekonstruiert. Dieses geschieht durch Rekonstruktion der Polygonpunkte.

5.6.3.1 Rekonstruktion eines 3D-Punktes

Zur Rekonstruktion eines Punktes werden $n \geq 2$ Bildebenen benötigt. Durch die Kamerakalibrierung erhält man die Projektionsmatrizen $\tilde{\boldsymbol{P}}^{(i)}$, $i \in \{1, \ldots, n\}$. Es seien $\tilde{\boldsymbol{m}}^{(i)} = (u^{(i)}, u^{(i)}, 1)^T$ die projektiven Darstellungen der korrespondierenden Punkte in den Bildebenen. Gesucht sind die euklidischen Koordinaten $\boldsymbol{M} = (x, y, z)^T$ des 3D-Punktes. Die projektiven Koordinaten sind $\tilde{\boldsymbol{M}} = (\boldsymbol{M}, 1)^T$. Aufgrund der Projektivität ergibt sich $\tilde{\boldsymbol{M}}$ aus der simultanen Lösung der Gleichungssysteme

$$\lambda^{(i)} \tilde{\boldsymbol{m}}^{(i)} = \tilde{\boldsymbol{P}}^{(i)} \tilde{\boldsymbol{M}} \qquad \text{für} \quad i \in \{1, \ldots, n\} \tag{5.91}$$

Diese Gleichungssysteme lassen sich schreiben als $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$ mit

$$\boldsymbol{A} = \begin{pmatrix} \tilde{\boldsymbol{Q}}_1^{(1)T} & -u^{(1)} & 0 & \ldots & 0 \\ \tilde{\boldsymbol{Q}}_2^{(1)T} & -v^{(1)} & 0 & \ldots & 0 \\ \tilde{\boldsymbol{Q}}_3^{(1)T} & 1 & 0 & \ldots & 0 \\ \tilde{\boldsymbol{Q}}_1^{(2)T} & 0 & -u^{(2)} & \ldots & 0 \\ \tilde{\boldsymbol{Q}}_2^{(2)T} & 0 & -v^{(2)} & \ldots & 0 \\ \tilde{\boldsymbol{Q}}_3^{(2)T} & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{Q}}_1^{(n)T} & 0 & 0 & \ldots & -u^{(n)} \\ \tilde{\boldsymbol{Q}}_2^{(n)T} & 0 & 0 & \ldots & -v^{(n)} \\ \tilde{\boldsymbol{Q}}_3^{(n)T} & 0 & 0 & \ldots & 1 \end{pmatrix} \qquad \boldsymbol{w} = \begin{pmatrix} x \\ y \\ z \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} \qquad \boldsymbol{b} = \begin{pmatrix} \tilde{\boldsymbol{m}}^{(1)} \\ \tilde{\boldsymbol{m}}^{(2)} \\ \vdots \\ \tilde{\boldsymbol{m}}^{(n)} \end{pmatrix} \tag{5.92}$$

Die Lösung dieses überbestimmten linearen Gleichungssystems im Sinne des kleinsten quadratischen Fehlers ist

$$\boldsymbol{w} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{b} \tag{5.93}$$

Der Lösungsvektor $\boldsymbol{w}$ enthält somit die Koordinaten des rekonstruierten 3D-Punktes.

5.6.4 Verifikation von Berandungspolygonen

Eine sich unmittelbar anschließende Berechnung und Identifikation der Oberflächen, die Gegenstand des folgenden Kapitels 5.7 sind, wäre unzureichend, da auch Berandungen erkannt werden können, die nicht real in der Szene vorhanden sind. Die Rekonstruktion der in Abbildung 5.14 durch ein durchgezogenes, schwarzes Polygon markierte Berandung hat offensichtlich eine Form, die sehr ähnlich zu der Form des rekonstruierten Polygons ist, welches durch ein durchgezogenes, weißes Polygon markiert ist. Erfolgt in diesem Beispiel die Erkennung größeninvariant, so würde fälschlicherweise die Rekonstruktion der schwarz markierten Berandung erkannt. Dieses Problem wird durch eine Verifikation des Polygons in einer oder mehreren Ansichten gelöst. Als Kriterium dient das Verhältnis zwischen der Anzahl von Kantenpunkten im Grauwertbild auf dem Polygonzug und der Länge des Polygonzugs. Es werden nur die Oberflächen aus den Polygonen berechnet, für die dieses Verhältnis groß genug ist.

5.7 Modellierung der Oberflächen

An der obersten Stufe der Abstraktionshierarchie steht das Modell der polygonalen, ebenen Objektoberflächen. Das wesentliche Merkmal dieser Oberfläche ist die zweidimensionale Form. Die Form ist das positions-, rotations- und skalierungsinvariante Merkmal eines Objektes, daher muß ein Verfahren, das zwei Formen unterscheidet, auch diese Invarianten besitzen.

Zur Beschreibung der Form hat sich der Ansatz von Arkin et al. [2] als geeignet erwiesen. Die Kontur wird durch eine Funktion repräsentiert, die des weiteren als Konturfunktion bezeichnet wird. Eine auf der L_p-Norm basierende Metrik ermöglicht, die Ähnlichkeit zweier Konturfunktionen und somit zweier Formen zu messen.

Zur Unterscheidung zweier Oberflächen kann, je nach Komplexität der Szene, das Hinzuziehen der Größe der Oberfläche als weiteres Merkmal notwendig sein. Daher wird ein Ähnlichkeitsmaß für Oberflächen definiert, welches sich auf die Metrik der Form und den Flächeninhalt stützt. Da zur Lokalisation des Kamerasystems die Lage und die Orientierung der Objektoberfläche benötigt werden, ist der Flächenschwerpunkt und der Normalenvektor der Fläche ebenfalls Bestandteil des Modells (siehe Abb. 5.15).

Die Berechnung des Modells aus einem 3D-Polygon erfolgt in drei wesentlichen Schritten:

1. Berechnung der optimalen Ebene, in der die Punkte des 3D-Polygons liegen
2. Projektion des 3D-Polygons auf die Ebene

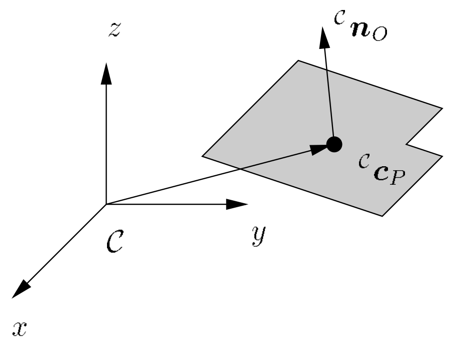


Abbildung 5.15: Flächenschwerpunkt und Normalenvektor des Oberflächenmodells

3. Berechnung der Modellparameter des resultierenden 2D-Polygons (Aufpunkt, Normalenvektor, Fläche, Form)

5.7.1 Optimale Ebene

Die Schätzung der optimalen Ebene erfolgt mittels Hauptachsentransformation. Die 3D-Polygonpunkte $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{n_P-1}$ werden als Realisierungen einer dreidimensionalen Zufallsvariablen $\boldsymbol{X}$ aufgefaßt. Die Basisvektoren des Oberflächenkoordinatensystems sind die Eigenvektoren, die zu den 2 größten Eigenwerten der Kovarianzmatrix von $\boldsymbol{X}$ gehören. Die Kovarianzmatrix sei mit $\boldsymbol{A}$ bezeichnet. Sie ist definiert durch

$$A_{ij} = E\{(X_i - E\{X_i\})(X_j - E\{X_j\})\} \tag{5.94}$$

und es gilt

$$\boldsymbol{A} = \frac{1}{n_P} \sum_{j=0}^{n_P-1} (\boldsymbol{p}_j - \boldsymbol{\mu}) \cdot (\boldsymbol{p}_j - \boldsymbol{\mu})^T \tag{5.95}$$

mit dem Erwartungswert

$$\boldsymbol{\mu} = E\{\boldsymbol{X}\} = \frac{1}{n_p} \sum_{i=0}^{n_P-1} \boldsymbol{p}_i \tag{5.96}$$

Da die Menge der 3D-Polygonpunkte der Schnitt mindestens 3 verschiedener 3D-Geraden ist, sind die Punkte des Polygons nicht kollinear. Somit ist die Ordnung von $\boldsymbol{A}$ größer oder gleich 2. Da $\boldsymbol{A}$ symmetrisch und positiv semidefinit ist[3], existieren somit 2 reelle Eigenwerte

$$\lambda_1 > \lambda_2 > 0 \tag{5.97}$$

wobei λ_1 der größte Eigenwert der Matrix $\boldsymbol{A}$ ist, mit den entsprechenden Eigenvektoren $\boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathbb{R}^3$:

$$\boldsymbol{A}\boldsymbol{q}_k = \lambda_k \boldsymbol{q}_k \quad k \in \{1, 2\} \tag{5.98}$$

Die Vektoren $\boldsymbol{q}_1$ und $\boldsymbol{q}_2$ bilden ein 2D-Koordinatensystem mit Ursprung $\boldsymbol{\mu}$, das die optimale Ebene

$$\boldsymbol{e}(\alpha, \beta) = \boldsymbol{\mu} + \alpha \boldsymbol{q}_1 + \beta \boldsymbol{q}_2 \tag{5.99}$$

beschreibt.

Die Projektion der 3D-Polygonpunkte auf diese Ebene ist die Projektion der mit $-\boldsymbol{\mu}$ translatierten Punkte auf $\boldsymbol{q}_1$ und $\boldsymbol{q}_2$:

$$^{\mathcal{O}}\boldsymbol{p}_k = \begin{pmatrix} \boldsymbol{q}_1{}^T \\ \boldsymbol{q}_2{}^T \end{pmatrix} (\boldsymbol{p}_k - \boldsymbol{\mu}) \quad , i = 0, \ldots, n_P - 1 \tag{5.100}$$

[3] siehe Anhang A

5.7.2 Der Flächenschwerpunkt eines Polygons

Durch Induktion läßt sich zeigen, daß der Flächenschwerpunkt eines 2D-Polygons gegeben ist durch

$$ {}^{\mathcal{O}}\boldsymbol{c}_P = \frac{1}{A_P}\sum_{i=2}^{n_P-1} A_{\delta_i}\cdot \boldsymbol{c}_{\delta_i} \tag{5.101}$$

$$ = \frac{1}{A_P}\sum_{i=2}^{n_P-1} \frac{A_{\delta_i}}{3}({}^{\mathcal{O}}\boldsymbol{p}_i + {}^{\mathcal{O}}\boldsymbol{p}_{i-1} + {}^{\mathcal{O}}\boldsymbol{p}_0) \tag{5.102}$$

wobei A_P die orientierte Fläche des Polygons, δ_i das durch ${}^{\mathcal{O}}\boldsymbol{p}_i$, ${}^{\mathcal{O}}\boldsymbol{p}_{i-1}$ und ${}^{\mathcal{O}}\boldsymbol{p}_0$, beschriebene Dreieck, A_{δ_i} dessen orientierte Fläche und $\boldsymbol{c}_{\delta_i}$ den Schwerpunkt des Dreiecks bezeichnen.

Den 3D-Flächenschwerpunkt ergibt sich aus

$$ {}^{\mathcal{C}}\boldsymbol{c}_P = \boldsymbol{\mu} + \begin{pmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 \end{pmatrix} \cdot {}^{\mathcal{O}}\boldsymbol{c}_P \tag{5.103}$$

mit $\boldsymbol{\mu}$ in (5.96) und $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ in (5.98).

5.7.3 Die Konturfunktion

In diesem Kapitel soll kurz die Konturfunktion vorgestellt werden [2]. Es folgen eine informale Definition der Funktion, ihre im Hinblick auf die Repräsentation der Form wesentlichen Eigenschaften, sowie die für den Vergleich zweier Formen benötigte Metrik.

Sei O ein Punkt auf der Kontur eines zweidimensionalen Objektes A mit der Länge l_A. Wandert man von O ausgehend zu einem Punkt P auf der Kontur derart, daß sich auf der linken Seite der Kontur das Innere des Objektes befindet, so hat man eine mit $w \in [0, l_A]$ bezeichnete Weglänge zurückgelegt. Man ordnet zunächst der Weglänge w den Winkel im Bogenmaß zu, den die Tangente in P mit der Horizontalen bildet, unter Berücksichtigung der bereits durchgeführten Umläufe. Durch eine Parametrisierung $[0,1] \to [0, l_A]; \sigma \mapsto \sigma \cdot l_A$ erhält man nun die Konturfunktion $\Theta_{A,O} : [0,1] \to \mathbb{R}$.

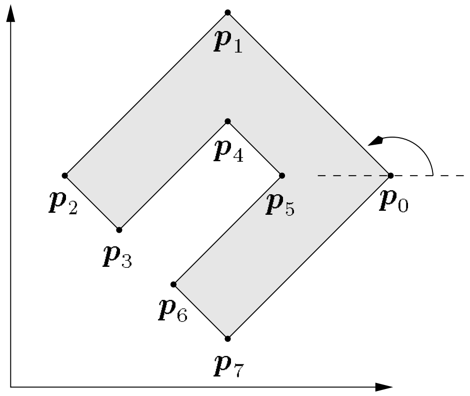
Abbildung 5.16(a) stellt die Konturfunktion zum Objekt in Abbildung 5.16(b) graphisch dar. Die Stützstellen $\sigma_0, \ldots, \sigma_8 \in [0,1]$ ergeben sich aus der Parametrisierung des durch die Polygonpunkte $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_7$ festgelegten Weges.

Die Parametrisierung impliziert Invarianz gegenüber Skalierung.

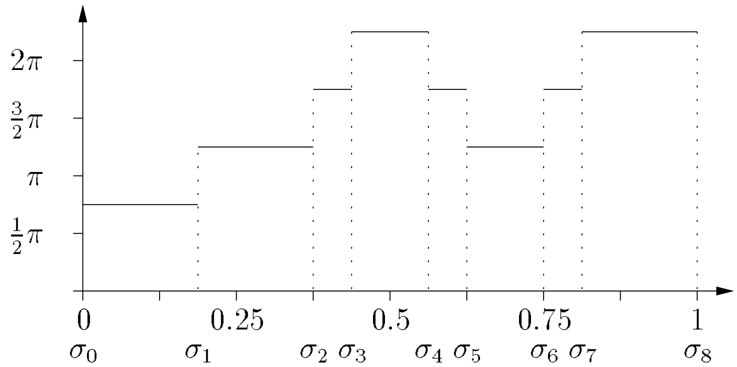
Die Konturfunktion ist periodisch im folgenden Sinne:

$$ \Theta_{A,O}(s+1) = \Theta_{A,O}(s) + 2\pi \tag{5.104}$$

Die Konturfunktion $\Theta_{A,O}$ ist bestimmt durch die Wahl des Anfangspunktes O, so daß zwei gleichen Objekten bei Wahl verschiedener Anfangspunkte verschiedene Konturfunktionen



(a) Polygon



(b) Konturfunktion

Abbildung 5.16: Polygon mit entsprechender Konturfunktion

zugeordnet sind:
Sind Θ_{A,O_1} und Θ_{A,O_2} zwei Konturfunktionen eines Objektes A und w die Weglänge von O_1 nach O_2, so gilt:

$$\Theta_{A,O_1}(s+t) = \Theta_{A,O_2}(s) \qquad \text{mit} \quad t = \frac{w}{l_A}. \tag{5.105}$$

Ein weiterer Freiheitsgrad besteht in der Drehung des Objektes A um einen beliebigen Punkt Q auf der Kontur: Geht das Objekt A durch Drehung um einen Winkel θ um einen beliebigen Punkt Q in das Objekt A' über, so folgt:

$$\Theta_{A,O}(s) + \theta = \Theta_{A',O}(s), \tag{5.106}$$

mit einem auf der Kontur festgehaltenen Anfangspunkt O.

Durch Translation, Rotation und Skalierung des Objektes A erhält man also Konturfunktionen, die alle die gleiche Form beschreiben. Daher bilden die durch Konturfunktionen beschriebenen Formen eine Äquivalenzrelation und

$$\Theta_A = \{\Theta_{A',O'} \,|\, \exists\, t, \theta \in \mathrm{IR}\, \forall s \in [0,1] : \Theta_{A',O'}(s+t) + \theta = \Theta_{A,O}(s)\} \tag{5.107}$$

ist die Äquivalenzklasse der Form A.

Seien A und B Formen, so ist der Abstand $d_2(A,B)$ definiert durch

$$d_2(A,B) = \left(\min_{\theta \in \mathrm{IR}, t \in [0,1]} D_2^{A,B}(t,\theta) \right)^{\frac{1}{2}} \tag{5.108}$$

mit

$$D_2^{A,B}(t,\theta) = \int_0^1 |\Theta_A(s+t) - \Theta_B(s) + \theta|^2 \, ds. \tag{5.109}$$

Der Beweis, daß d_2 eine Metrik ist, ist ebenfalls in [2] erbracht.

Sei die Kontur eines Objektes A durch das einfache Polygon gegeben, das durch die in die Ebene transformierten Scheitel $\{{}^{\mathcal{O}}\boldsymbol{p}_0, \ldots, {}^{\mathcal{O}}\boldsymbol{p}_{n_P-1}\}$ (siehe Gl. (5.100)) definiert wird.

Die Konturfunktion Θ_A wird repräsentiert durch

$$\overline{\Theta}_{A,O_A} = \{(\sigma_0, \alpha_0), \ldots, (\sigma_{n_A}, \alpha_{n_A})\} \quad , n_A = n_P + 1 \tag{5.110}$$

wobei σ_i die Stützstellen und α_i die Winkel im Bogenmaß der in $\boldsymbol{p}_i$ beginnenden Konturabschnitte sind.

In [2] wird ebenfalls ein effizienter Algorithmus angegeben, der den Ähnlichkeitsabstand zweier Konturrepräsentationen $\overline{\Theta}_{A,O_A}$ und $\overline{\Theta}_{B,O_B}$ berechnet. Dieser ist von der Komplexität $O(n_A n_B \log_2(n_A n_B))$. Insbesondere kann somit auch der Abstand $d_2(A,B)$ zweier Formen A und B berechnet werden, deren Konturpolygone eine unterschiedliche Anzahl von Punkten besitzen.

5.7.4 Das Oberflächenmodell

Die polygonale Oberfläche ist modelliert durch

- den Aufpunkt ${}^{\mathcal{C}}\boldsymbol{c}_P$ aus Gleichung (5.103)
- den Normalenvektor

$$\boldsymbol{n}_O = \boldsymbol{q}_1 \times \boldsymbol{q}_2 \tag{5.111}$$

  mit $\boldsymbol{q}_1$ und $\boldsymbol{q}_2$ gemäß Gleichung (5.98)
- die Fläche $F_O := |A_P|$ mit A_P aus Kap. 5.6.2

- die Form, dazu gehören:
  - die Repräsentation der Konturfunktion in (5.110)
  - die normierte Fläche

$$F_S = \frac{F_O}{l_O^2} \tag{5.112}$$

mit der Fläche F_O und der Konturlänge l_O des Bildobjektes, von dem die Form stammt. Dieses Merkmal beschreibt die Rundheit einer Form.

5.7.5 Das Ähnlichkeitsmaß

Um die Ähnlichkeit zweier Oberflächenmodelle vergleichen zu können, muß neben der Form auch die Größe einbezogen werden. Das Maß für die Ähnlichkeit zweier Oberflächen sei definiert durch

$$d_O(O_1, O_2) = \omega_S \cdot d_2(S(O_1), S(O_2)) + \omega_O \cdot |\frac{F_{O_1} - F_{O_2}}{F_{O_1} + F_{O_2}}| \tag{5.113}$$

wobei $S(O)$ die Konturfunktion eines Objektes bezeichnet. Die Faktoren $\omega_S, \omega_O \in \mathbb{R}_{\geq 0}$ dienen zur Wichtung von Formabstand und Ähnlichkeit der Größe. Die Metrik $|\frac{F_{O_1} - F_{O_2}}{F_{O_1} + F_{O_2}}|$ hat gegenüber der Metrik $|F_{O_1} - F_{O_2}|$ den Vorteil, daß die Ähnlichkeit zweier Flächen nicht von deren absoluter Größe abhängig ist. Sie wird ausschließlich bestimmt durch die relative Größe der beiden Flächen, was eher der Realität entspricht. Ist $r = \frac{F_{O_1}}{F_{O_2}}$, so gilt

$$|\frac{F_{O_1} - F_{O_2}}{F_{O_1} + F_{O_2}}| = |\frac{r - 1}{r + 1}| \tag{5.114}$$

Die aus der Szene extrahierten Oberflächen werden mit einem Modell, das in einer Datenbank enthalten ist, durch das Ähnlichkeitsmaß in Gleichung (5.113) verglichen. Es wird die Objektoberfläche als Zielobjekt bestimmt, welches zu dem gewählten Objektmodell der Datenbank den kleinsten Abstand hat und unterhalb einer Schwelle θ_O liegt. Durch den Vergleich mit den Modellen der Datenbank lassen sich die aus den Bilddaten extrahierten Oberflächen also identifizieren.

Eine geeignete Schwelle ergibt sich aus dem maximal zulässigen Verhältnis der Fächen zweier Oberflächen O_1 und O_2 mit Gleichung (5.114), dem maximal zulässigen Abstand $d_2(S(O_1), S(O_2))$ der Formen und den Wichtungsfaktoren. Für zwei intuitiv ähnliche Formen gilt in etwa $d_2(O_1, O_2) \lesssim 0.4$, was durch Versuche ermittelt wurde.

Die Ausrichtung des Kamerasystems zum Zielobjekt wird im folgenden Kapitel beschrieben.

Kapitel 6

Ausrichtung des Kamerasystems zur Oberfläche

Inhalt dieses Kapitels ist die Berechnung der neuen Manipulatordaten zur Bewegung des Kamerasystems in eine neue Position und Orientierung derart, daß sich die Objektoberfläche in einer bestimmten Lage innerhalb des Kamerakoordinatensystems befindet. Diese neue Lage muß durch den Vektor $\boldsymbol{p}$, der die Koordinaten des Flächenschwerpunktes im neuen Kamerakoordinatensystem $\mathcal{C}_2$ und den Vektor $\boldsymbol{b}$, der die Blickrichtung im Kamerakoordinatensystem angibt, a priori festgelegt werden.

Zunächst muß die Abbildung $C : \mathcal{C}_1 \rightarrow \mathcal{C}_2$ bestimmt werden. Mittels der aus der Hand-Auge-Kalibrierung gewonnenen Abbildung lassen sich dann aus den aktuellen Roboterdaten die neuen Roboterdaten berechnen. Gesucht ist also die Matrix

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{R}_C & \boldsymbol{t}_C \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{6.1}$$

Zur Veranschaulichung der nun folgenden Berechnungen dient Abbildung 6.1.

Die neue Orientierung wird erreicht, indem der Vektor ${}^{\mathcal{C}_1}\boldsymbol{n}_O$ auf den Vektor

$${}^{\mathcal{C}_1}\boldsymbol{w} = \begin{cases} -\boldsymbol{b} & \text{, falls } {}^{\mathcal{C}_1}\boldsymbol{n}_O{}^T \cdot \boldsymbol{b} < 0 \\ \boldsymbol{b} & \text{, sonst} \end{cases} \tag{6.2}$$

rotiert wird. Dieses geschieht durch eine Rotation um den auf ${}^{\mathcal{C}_1}\boldsymbol{n}_O$ und ${}^{\mathcal{C}_1}\boldsymbol{w}$ senkrecht stehenden Vektor

$$\boldsymbol{u}_r = {}^{\mathcal{C}_1}\boldsymbol{n}_O \times {}^{\mathcal{C}_1}\boldsymbol{w} \tag{6.3}$$

um den Winkel ψ. Es gilt, da $\|{}^{\mathcal{C}_1}\boldsymbol{n}_O\| = \|{}^{\mathcal{C}_1}\boldsymbol{w}\| = 1$:

$$\sin(\psi) = \frac{\|{}^{\mathcal{C}_1}\boldsymbol{n}_O \times {}^{\mathcal{C}_1}\boldsymbol{w}\|}{\|{}^{\mathcal{C}_1}\boldsymbol{n}_O\| \cdot \|{}^{\mathcal{C}_1}\boldsymbol{w}\|} = \|\boldsymbol{u}_r\| \tag{6.4}$$

und somit

$$\psi = \arcsin(\|\boldsymbol{u}_r\|) \in [-\frac{\pi}{2}, \frac{\pi}{2}] \tag{6.5}$$

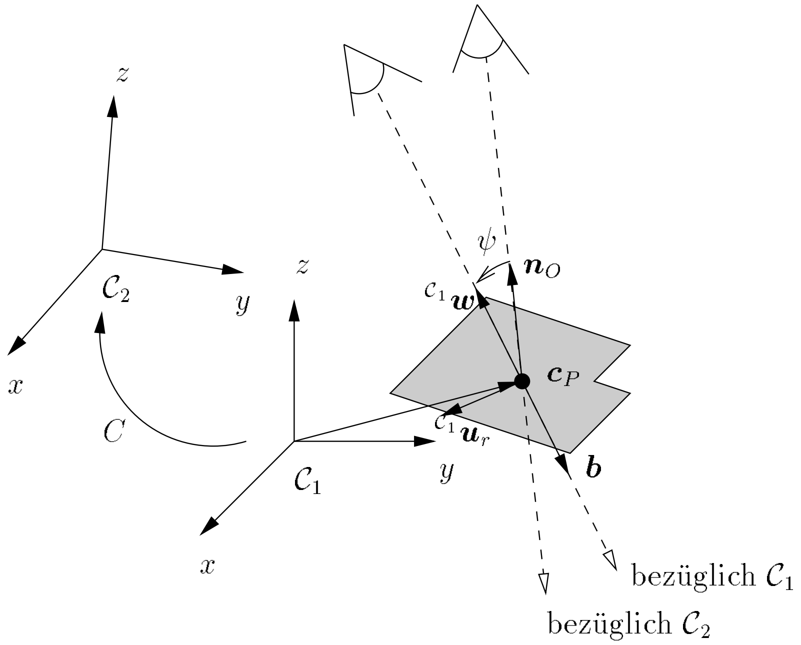


Abbildung 6.1: Lage des Oberflächenmodells bezüglich der Kamerakoordinatensysteme vor ($\mathcal{C}_1$) und nach ($\mathcal{C}_2$) der Bewegung

Nach [9] gilt mit

$$\boldsymbol{u} = \frac{\boldsymbol{u}_r}{\|\boldsymbol{u}_r\|} \tag{6.6}$$

und

$$s = \cos(\frac{\psi}{2}) \tag{6.7a}$$

$$\boldsymbol{v} = \sin(\frac{\psi}{2}) \cdot \boldsymbol{u} \tag{6.7b}$$

für den Rotationsanteil[1]:

$$\boldsymbol{R}_C = \begin{pmatrix} s^2 + \boldsymbol{v}_0^2 - \boldsymbol{v}_1^2 - \boldsymbol{v}_2^2 & 2(\boldsymbol{v}_0\boldsymbol{v}_1 - s\boldsymbol{v}_2) & 2(\boldsymbol{v}_0\boldsymbol{v}_2 + s\boldsymbol{v}_1) \\ 2(\boldsymbol{v}_0\boldsymbol{v}_1 + s\boldsymbol{v}_2) & s^2 - \boldsymbol{v}_0^2 + \boldsymbol{v}_1^2 - \boldsymbol{v}_2^2 & 2(\boldsymbol{v}_1\boldsymbol{v}_2 - s\boldsymbol{v}_0) \\ 2(\boldsymbol{v}_0\boldsymbol{v}_2 - s\boldsymbol{v}_1) & 2(\boldsymbol{v}_1\boldsymbol{v}_2 + s\boldsymbol{v}_0) & s^2 - \boldsymbol{v}_0^2 - \boldsymbol{v}_1^2 + \boldsymbol{v}_2^2 \end{pmatrix} \tag{6.8}$$

Die Koordinaten eines beliebigen Punktes $\boldsymbol{q}$ werden durch die Bewegung von $\mathcal{C}_1$ nach $\mathcal{C}_2$ folgendermaßen transformiert: ${}^{\mathcal{C}_2}\boldsymbol{q} = \boldsymbol{R}_C{}^{\mathcal{C}_1}\boldsymbol{q} + \boldsymbol{t}_C$. Durch die Korrekturbewegung soll erreicht

[1] Die Matrix $\boldsymbol{R}_C$ folgt aus der Beschreibung von Rotationen mittels Quaternionen: Für $\boldsymbol{x} \in \mathbb{R}^3$ gilt $(0, \boldsymbol{R}_C\boldsymbol{x}) = \boldsymbol{q} \times (0, \boldsymbol{x}) \times \overline{\boldsymbol{q}}$ mit $\boldsymbol{q} = (s, \boldsymbol{v})$, $\overline{\boldsymbol{q}} = (s, -\boldsymbol{v})$, wobei $\times$ das Quaternionenprodukt ist. Siehe dazu [9, Kap. 5.5.2].

werden, daß der transformierte Flächenschwerpunkt ${}^{\mathcal{C}_2}\boldsymbol{c}_P$ mit dem Punkt $\boldsymbol{p}$ übereinstimmt. Somit folgt:

$$\boldsymbol{t}_C = {}^{\mathcal{C}_2}\boldsymbol{p} - \boldsymbol{R}_C {}^{\mathcal{C}_1}\boldsymbol{c}_P \tag{6.9}$$

Nun folgt die Berechnung der Matrix $\widehat{\boldsymbol{W}}_2$, welche die neue Lage des Toolkoordinatensystems $\mathcal{T}_2$ im Basiskoordinatensystem $\mathcal{B}$ des Roboters beschreibt. Es gilt, wie Abbildung 6.2 verdeutlicht:

$$\begin{aligned} \boldsymbol{W}_2 &= \boldsymbol{T}\boldsymbol{W}_1 && (6.10)\\ &= \boldsymbol{X}^{-1}\boldsymbol{C}\boldsymbol{X}\boldsymbol{W}_1 && (6.11) \end{aligned}$$

Die Lage des Koordinatensystems $\mathcal{T}_2$ bezüglich $\mathcal{B}$ ergibt sich durch Inversion von $\boldsymbol{W}_2$:

$$\widehat{\boldsymbol{W}}_2 = \boldsymbol{W}_2{}^{-1} = (\boldsymbol{X}^{-1}\boldsymbol{C}\boldsymbol{X}\boldsymbol{W}_1)^{-1} = \boldsymbol{W}_1{}^{-1}\boldsymbol{X}^{-1}\boldsymbol{C}^{-1}\boldsymbol{X} \tag{6.12}$$

Es folgt:

$$\begin{aligned} &\begin{pmatrix} \widehat{\boldsymbol{R}}_{W_2} & \widehat{\boldsymbol{t}}_{W_2} \\ \boldsymbol{0} & 1 \end{pmatrix} && (6.13)\\ &= \begin{pmatrix} \widehat{\boldsymbol{R}}_{W_1} & \widehat{\boldsymbol{t}}_{W_1} \\ \boldsymbol{0} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_X{}^T & -\boldsymbol{R}_X{}^T\boldsymbol{t}_X \\ \boldsymbol{0} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_C{}^T & -\boldsymbol{R}_C{}^T\boldsymbol{t}_C \\ \boldsymbol{0} & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_X & \boldsymbol{t}_X \\ \boldsymbol{0} & 1 \end{pmatrix} && (6.14)\\ &= \begin{pmatrix} \widehat{\boldsymbol{R}}_{W_1}\boldsymbol{R}_X{}^T\boldsymbol{R}_C{}^T\boldsymbol{R}_X & \widehat{\boldsymbol{R}}_{W_1}\boldsymbol{R}_X{}^T(\boldsymbol{R}_C{}^T(\boldsymbol{t}_X - \boldsymbol{t}_C) - \boldsymbol{t}_X) + \widehat{\boldsymbol{t}}_{W_1} \\ \boldsymbol{0} & 1 \end{pmatrix} && (6.15) \end{aligned}$$

Die Orientierungsmatrix $\widehat{\boldsymbol{R}}_{W_1}$ und der Translationsvektor $\widehat{\boldsymbol{t}}_{W_1}$ ergeben sich aus den Daten, die von der Peripherie des Roboterarmes ausgelesen werden, wobei die Berechnung der Orientierungsmatrix $\widehat{\boldsymbol{R}}_{W_1}$ aus den ZYZ-Eulerwinkeln nach Bemerkung 1 (Kap. 3.2) erfolgt. Mit Bemerkung 2 ergeben die neuen ZYZ-Eulerwinkel α, β und γ aus der Matrix $\widehat{\boldsymbol{R}}_{W_2}$. Insgesamt ergeben sich somit die Roboterdaten $(\widehat{\boldsymbol{t}}_2, \alpha, \beta, \gamma)$ zum Anfahren der neuen Toolkonfiguration.

$$\begin{tikzcd}
 & \mathcal{T}_1 \arrow[r, bend left, "\boldsymbol{X}"] \arrow[dd, bend left, "\boldsymbol{T}"] & \mathcal{C}_1 \arrow[dd, bend left, "\boldsymbol{C}"] \\
\mathcal{B} \arrow[ur, bend left, "\boldsymbol{W}_1"] \arrow[dr, bend left, "\boldsymbol{W}_2"] & & \\
 & \mathcal{T}_2 \arrow[r, bend left, "\boldsymbol{X}"] & \mathcal{C}_2
\end{tikzcd}$$

Abbildung 6.2: Diagramm der Transformationen zwischen den Koordinatensystemen

Kapitel 7

Testergebnisse des ersten Teils

In diesem Kapitel werden die Testergebnisse des für die Erkennung der polygonalen, planaren Objektoberfläche und die Bewegung des Kamerasystems durch den Roboter relevanten Teilsystems vorgestellt.

7.1 Kalibrierung

7.1.1 Kamerakalibrierung

Wie bereits in Kapitel 4.1 beschrieben wurde, liegen der Schätzung der Projektionsmatrix einer Kamera n Paare $(\boldsymbol{M}_i, \boldsymbol{m}_i)$, $i = 1, \dots, n$ von Weltkoordinaten im 3D-Kamerakoordinatensystem $\mathcal{C}$ und korrespondierenden Bildkoordinaten in einer Bildebene $\mathcal{R}$ zugrunde, die man durch das Abfahren eines Punktegitters mit dem Roboterarm und Verarbeitung des Bildes des Kalibrierungsmusters erhält. Das kreisförmige Kalibrierungsmuster hat einen Durchmesser von 100 mm. Den Gitterpunkten sind Punkte im Kamerakoordinatensystem $\mathcal{C}$ zugeordnet, es ergibt sich somit ein Punktegitter im Koordinatensystem $\mathcal{C}$.

Die Kalibrierung erfolgte durch $n = 75$ Punktepaare. Das Punktegitter ist gegeben durch

$$\boldsymbol{M}_{i,j,k} = i \cdot \begin{pmatrix} 70 \\ 0 \\ 0 \end{pmatrix} + j \cdot \begin{pmatrix} 0 \\ 70 \\ 0 \end{pmatrix} + k \cdot \begin{pmatrix} 0 \\ 0 \\ 60 \end{pmatrix} \qquad i, j \in \{-2, -1, \dots, 2\}, k \in \{-1, 0, 1\} \tag{7.1}$$

Für die optischen Zentren der 3 Kameras, d. h. Stereokameras und die sich durch Anfahren ergebende dritte Kamera, erhält man aus Gleichung (4.4) beispielsweise (in Millimetern):

$$\boldsymbol{c}_1 = \begin{pmatrix} -42.26 \\ 193.37 \\ -749.35 \end{pmatrix} \quad \boldsymbol{c}_2 = \begin{pmatrix} -41.23 \\ -187.08 \\ -746.72 \end{pmatrix} \quad \boldsymbol{c}_3 = \begin{pmatrix} 98.67 \\ 194.24 \\ -777.17 \end{pmatrix} \tag{7.2}$$

Daraus ergeben sich folgende Abstände der optischen Zentren (*baselines*):

$$\overline{\boldsymbol{c}_1\boldsymbol{c}_2} = 380.46\,\mathrm{mm} \qquad \overline{\boldsymbol{c}_1\boldsymbol{c}_3} = 143.65\,\mathrm{mm} \qquad \overline{\boldsymbol{c}_2\boldsymbol{c}_3} = 407.31\,\mathrm{mm} \tag{7.3}$$

Die optischen Zentren und deren Abstände charakterisieren den Versuchsaufbau.

Zur Ermittlung der Fehler wird das Gitter $m = 10$ mal abgefahren und jeweils kalibriert. Für jeden Gitterpunkt betrachten wir zunächst die Standardabweichung

$$\sigma = \sqrt{\frac{1}{m}\sum_{l=1}^{m}(\boldsymbol{m}_l - \boldsymbol{\mu}_B)^2} \qquad \text{mit} \quad \boldsymbol{\mu}_B = \frac{1}{m}\sum_{i=l}^{m}\boldsymbol{m}_l \tag{7.4}$$

der Bildgeradenschnittpunkte $\boldsymbol{m}_l$. Abbildung 7.1 visualisiert die Ergebnisse. Sie zeigt die Standardabweichung in Abhängigkeit vom Mittelwert $\boldsymbol{\mu}_B$. Es wird deutlich, daß die Schnittpunkte an den Bildecken eine größere Abweichung aufweisen als im übrigen Bild. Die maximale Standardabweichung beträgt 1.4 Pixel, insgesamt jedoch nur 0.4 Pixel. Die höhere Abweichung in den Ecken ist darauf zurückzuführen, daß das Kalibrierungsmuster nicht vollständig abgebildet wird und die Geradendetektion auf kurzen Bildkanten instabiler ist.

Es wird nun verifiziert, daß sich die Projektion der Punkte im Raum auf die Bildebene einer Kamera durch eine lineare projektive Abbildung beschreiben läßt. Dazu betrachten wir die Wurzel des mittleren quadratischen Fehlers zwischen den projizierten 3D-Punkten und $\boldsymbol{\mu}_B$ pro Gitterpunkt $\boldsymbol{M}$:

$$\text{rmse} = \sqrt{\frac{1}{m}\sum_{l=1}^{m}((\frac{x_l}{s_l}, \frac{y_l}{s_l})^T - \boldsymbol{\mu}_B)^2} \quad \text{mit} \quad (x_l, y_l, s_l)^T = \tilde{\boldsymbol{P}}_l\tilde{\boldsymbol{M}}_l \tag{7.5}$$

wobei $\tilde{\boldsymbol{M}}_l = \begin{pmatrix} \boldsymbol{M}_l & 1 \end{pmatrix}^T$ sich aus den Roboterkoordinaten des angefahrenen Gitterpunktes ergibt. Der maximale Fehler beträgt 3.3 Pixel und tritt an einer Ecke des Bildes auf (siehe Abb. 7.2). Er ist auf die hohe Standardabweichung des Schnittpunktes zurückzuführen. Insgesamt ergibt sich ein Fehler von 1.3 Pixel. Dieser Wert bestimmt auch die Breite des Scanbereiches beim Stereomatching (Kap. 5.2.3.1). Sie sollte also mindestens 3 Pixel betragen.

7.1.2 Hand-Auge-Kalibrierung

Die Genauigkeit der Schätzung des Translationsanteils $\boldsymbol{t}_X$ der Matrix $\boldsymbol{X}$ gemäß Gleichung (4.13), welche die Beziehung zwischen dem Toolkoordinatensystem $\mathcal{T}$ und dem Kamerakoordinatensystem $\mathcal{C}$ festlegt, läßt sich praktisch nicht genau überprüfen, da der Ursprung des Toolkoordinatensystems im Innern des Endeffektors liegt. Zur Hand-Auge-Kalibrierung wurden 63 Punkte angefahren, wobei die Orientierung des Toolkoordinatensystems variiert wurde. Für die euklidische Vektornorm des Translationsanteils $\boldsymbol{t}_X$ ergab sich beispielsweise $\|\boldsymbol{t}_X\| = 828$ mm, während per Hand etwa 825 mm gemessen wurden. Da der Hand-Auge-Kalibrierung rekonstruierte Punkte zugrunde liegen, fließt der Fehler der Rekonstruktion von Punkten in die Kalibrierung ein. Die Rekonstruktionsfehler sind Gegenstand des folgenden Abschnitts.

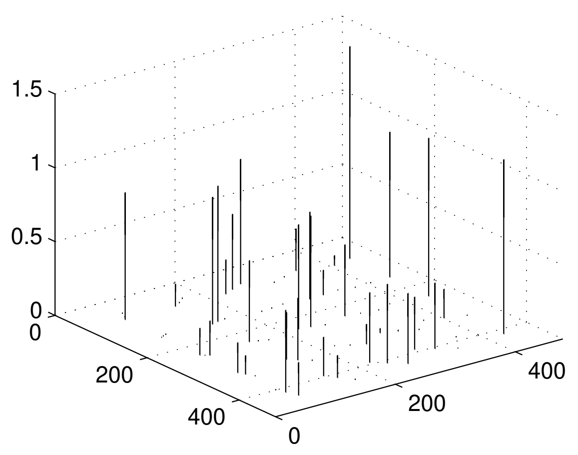


Abbildung 7.1: Standardabweichung der Schnittpunkte im Bildkoordinatensystem einer Kamera (in Pixel)

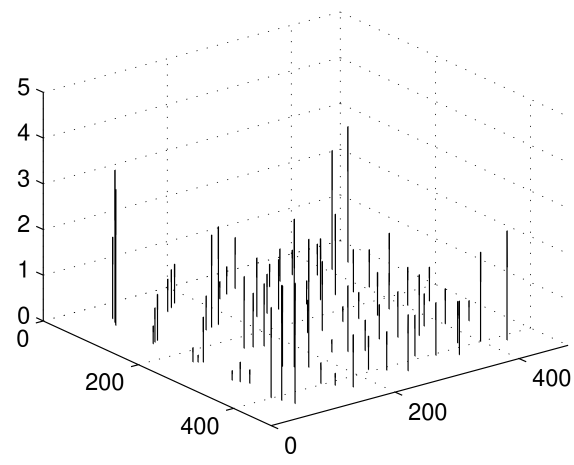


Abbildung 7.2: Fehler zwischen projiziertem Weltpunkt und mittlerem Schnittpunkt (in Pixel)

7.2 Rekonstruktion von Punkten und Geraden

Dieser Abschnitt beschreibt die Testergebnisse für die Rekonstruktion von Punkten und von Geraden. Dazu wurde wieder das Kalibrierungsmuster mit einem Durchmesser von 100 mm eingesetzt und ein Gitter $m = 10$ mal abgefahren, so daß sich das Punktegitter (7.1) im Kamerakoordinatensystem ergibt. Die Rekonstruktion der Punkte erfolgte aus zwei Ansichten, die Rekonstruktion der Geraden aus den notwendigen drei Ansichten.

Im Falle der Rekonstruktion von Punkten betrachten wir die Standardabweichung

$$\sigma = \sqrt{\frac{1}{m}\sum_{l=1}^{m}(\boldsymbol{p}_l - \boldsymbol{\mu}_p)^2} \qquad \text{mit} \quad \boldsymbol{\mu}_p = \frac{1}{m}\sum_{l=1}^{m}\boldsymbol{p}_l \tag{7.6}$$

wobei $\boldsymbol{p}_l$ einen rekonstruierten Punkt bezeichnet, sowie den Fehler

$$\text{rmse} = \sqrt{\frac{1}{m}\sum_{l=1}^{m}(\boldsymbol{p}_l - \boldsymbol{M}_{i,j,k})^2} \tag{7.7}$$

für jeden Gitterpunkt. Die maximale Standardabweichung beträgt 6.58 mm, der entsprechende Gitterpunkt liegt am Rand. Für die Gitterpunkte insgesamt ergibt sich eine Standardabweichung von 1.41 mm. Der maximale Fehler beträgt 10.1 mm, insgesamt 3.3 mm. Die Länge der Pfeile in Abbildung 7.3 ist der Fehler der Rekonstruktion an den jeweiligen Gitterpunkten.

Da bei der Rekonstruktion von Geraden nicht nachgeprüft werden kann, inwieweit die rekonstruierte Gerade mit der realen Geraden übereinstimmt, wird der Rekonstruktionsfehler

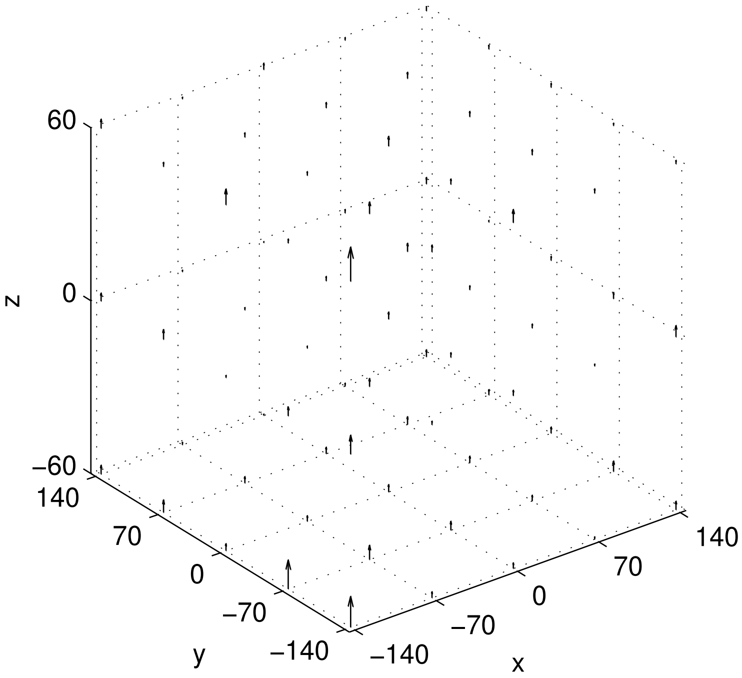


Abbildung 7.3: Fehler der Punktrekonstruktion in den Gitterpunkten

über den Abstand der zwei rekonstruierten Geraden bestimmt. Während sich die realen Geraden des Testmusters schneiden, existiert mit hoher Wahrscheinlichkeit kein Schnittpunkt der rekonstruierten Geraden. Bezeichnet d den Abstand der rekonstruierten Geraden, so wird der Fehler rmse $= \sqrt{\frac{1}{m}\sum_{l=1}^{m} d_l}$ an jedem Gitterpunkt betrachtet. Abbildung 7.4 stellt diese Fehler graphisch dar. Auffällig ist der maximale Fehler von 61 mm, der auf eine schlechte Geradendetektion zurückzuführen ist. Insgesamt liegt der Fehler deutlich unter 10 mm. An dieser Stelle sei betont, daß die Rekonstruktion von Geraden das Matching von Geraden über drei Ansichten voraussetzt. Für die Rekonstruktion des Schnittpunktes der zwei Bildgeraden ist die Bestimmung der Korrespondenzen in zwei Ansichten nicht notwendig, dennoch wurden die Korrespondenzen der Geraden in zwei Ansichten testhalber berechnet. Die Ergebnisse zeigen also insbesondere auch die Korrektheit der Matchingverfahren aus Kapitel 5.2.

7.3 Korrekturbewegung des Kamerasystems

Das Ziel des ersten Teilsystems ist die Bewegung des Kamerasystems derart, daß die Objektoberfläche unter einer durch den Vektor $\boldsymbol{b}$ definierten Blickrichtung erscheint und sich in einer durch $\boldsymbol{p}$ definierten Position im Kamerakoordinatensystem befindet (Kap. 6). In die Berechnung des neuen Roboterzustandes fließen die Fehler der Rekonstruktionen und

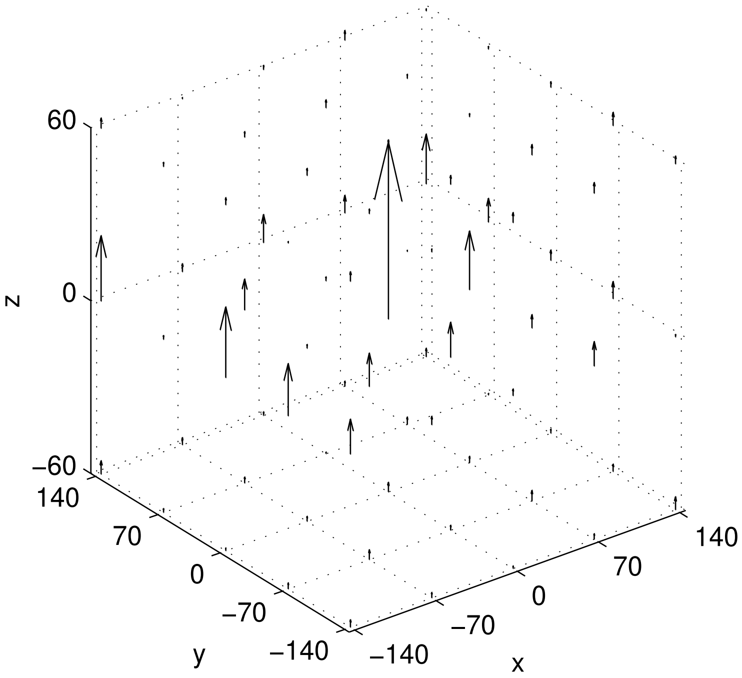


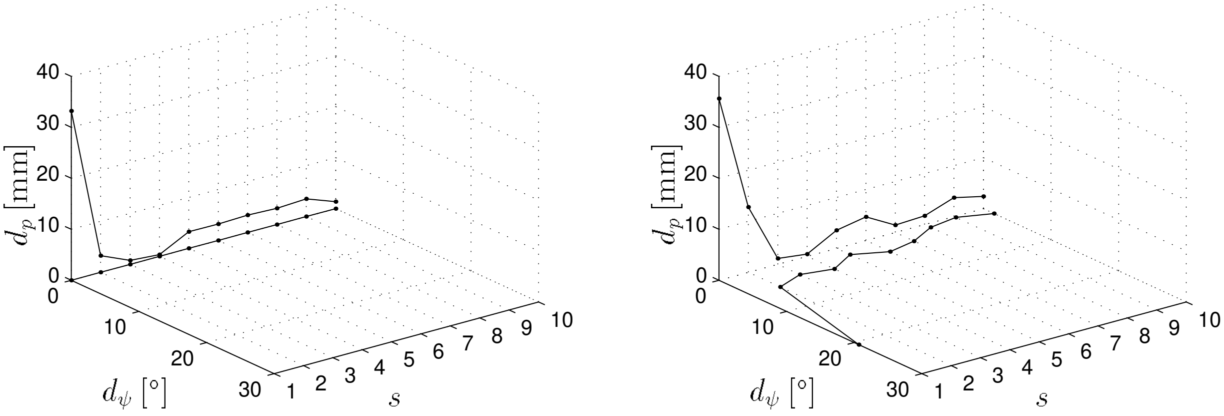
Abbildung 7.4: Abstand der rekonstruierten Geraden in den Gitterpunkten

der Hand-Auge-Kalibrierung ein. Um die Genauigkeit der Korrekturbewegung beurteilen zu können, werden Wahrnehmung und Aktion in einem Zyklus durchgeführt. Dazu wird zunächst ein einfaches Testmuster verwendet, ein schwarzes Quadrat mit einer Seitenlänge von 120 mm. Da die Detektion auf kürzeren Kanten instabiler ist, erfolgt ein weiterer Test mit einem Quadrat, dessen Seitenlänge 60 mm beträgt. Diese werden derart auf der Arbeitsfläche des Roboters positioniert, daß der Abstand der Flächenschwerpunkte vor und nach der Bewegung ca. 30 mm beträgt. Zum Test der Orientierungskorrektur wird der initiale Zustand des Roboters derart gewählt, daß die Blickrichtung und der Normalenvektor der quadratischen Flächen um 20° voneinander abweichen. Es werden die Fehler

$$d_p = \|\boldsymbol{p} - \boldsymbol{c}_P\| \qquad \text{und} \qquad d_\psi = |\psi| \tag{7.8}$$

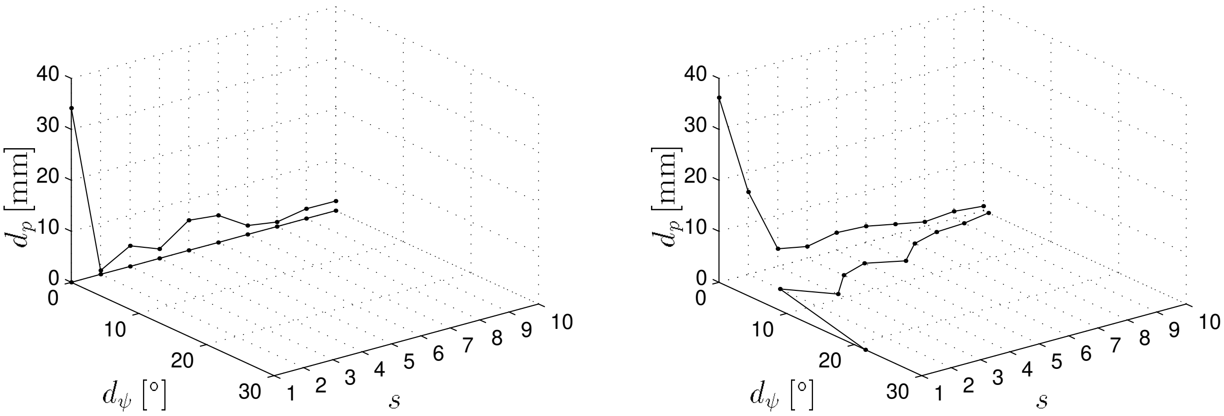
vor jeder Bewegung betrachtet, wobei $\boldsymbol{c}_P$ der Oberflächenschwerpunkt gemäß (5.103) und ψ der Korrekturwinkel in (6.5) ist. Da nicht genau bekannt ist, wie groß der Fehler bei der Hand-Auge-Kalibrierung ist, werden jeweils zwei Tests durchgeführt, ein Test ohne Korrektur der Orientierung, d. h. es wird in den Gleichungen (6.7) $\psi = 0$ gesetzt, und ein Test mit Korrektur. Da die Orientierung des Toolkoordinatensystems und des Kamerakoordinatensystems identisch sind, ist die Korrekturbewegung im Falle unveränderter Orientierung unabhängig von der relativen Lage dieser beiden Koordinatensysteme. Nur bei Korrektur der Orientierung hat also der Fehler der Hand-Auge-Kalibrierung Einfluß.

Die Abbildungen 7.5 und 7.6 zeigen die Fehlerverläufe $d_p(s)$ und $d_\psi(s)$ für die Korrekturschritte $s = 1, \ldots, 10$ jeweils für beide Testmuster. Bei fester Orientierung (Abb. 7.5(a) und



(a) ohne Korrektur der Orientierung (b) mit Korrektur der Orientierung

Abbildung 7.5: Fehler bei großem Testmuster

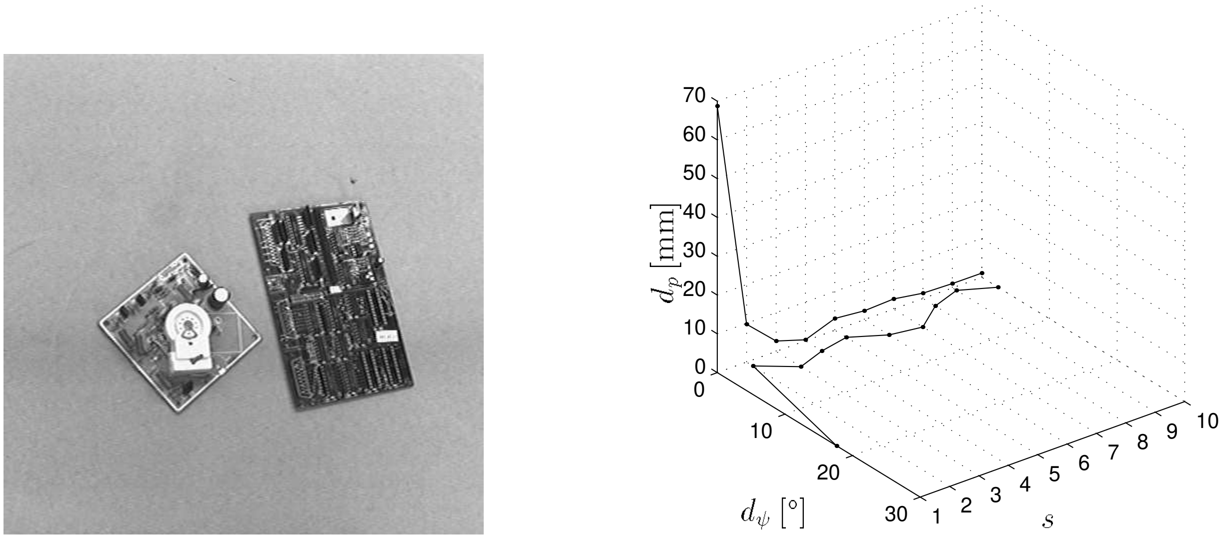


(a) ohne Korrektur der Orientierung (b) mit Korrektur der Orientierung

Abbildung 7.6: Fehler bei kleinem Testmuster

7.6(a)) ist der Fehler d_p bereits vor dem zweiten Korrekturschritt sehr klein und schwankt in dem durch den Fehler der Punktrekonstruktion gegebenen Rahmen. Beim kleineren Testmuster treten infolge der instabileren Geradendetektion etwas größere Schwankungen auf. Ohne Korrektur der Orientierung ist also nur ein Korrekturschritt erforderlich. Wird die Orientierung korrigiert (Abb. 7.5(b) u. 7.6(b)), so ist der Winkelfehler d_ψ vor dem zweiten Korrekturschritt zwar klein, aufgrund des Fehlers der Hand-Auge-Kalibrierung ist der Abstand d_p jedoch noch relativ hoch (13 mm bzw. 16 mm). Er wird erst im nachfolgenden Schritt korrigiert, da sich die Orientierung für $s = 2, \ldots, 10$ nicht wesentlich ändert. Mit Korrektur der Orientierung sind also zwei Schritte erforderlich, um das Kamerasystem in die gewünschte Position und Orientierung zum Objekt zu bewegen.

Abschließend soll das Verhalten an einer Szene mit zwei Platinen getestet werden. Abbil-



(a) Bild der ersten Kamera (b) Fehler

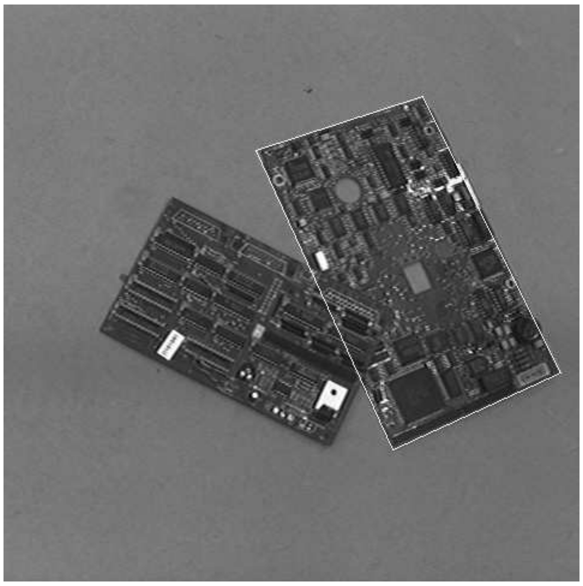
Abbildung 7.7: Test der Korrekturbewegung an einer realen Szene

dung 7.7(a) zeigt die Szene aus der Sicht der linken Kamera. Der Objekterkennung lag das in Kapitel 5.7.4 beschriebene Modell der großen, rechteckigen Platine zugrunde, deren Seitenlängen 100 mm und 160 mm betragen. Die große Platine wird stabil erkannt, der Fehler d_p liegt nach dem zweiten Korrekturschritt unterhalb von 3 mm, der Winkelfehler d_ϕ unterhalb von 5°. Die Abweichung von der realen Orientierung der Platinenoberfläche wird von der Steigung der Grauwertkanten an der Berandung der Objekte beeinflußt. Existieren an den Objektberandungen beispielsweise Schatten, so ist die Detektion der Bildgeraden ungenauer, so daß sich höhere Rekonstruktionsfehler ergeben. Je kleiner die Objekte sind, desto größer ist dann der Orientierungsfehler.

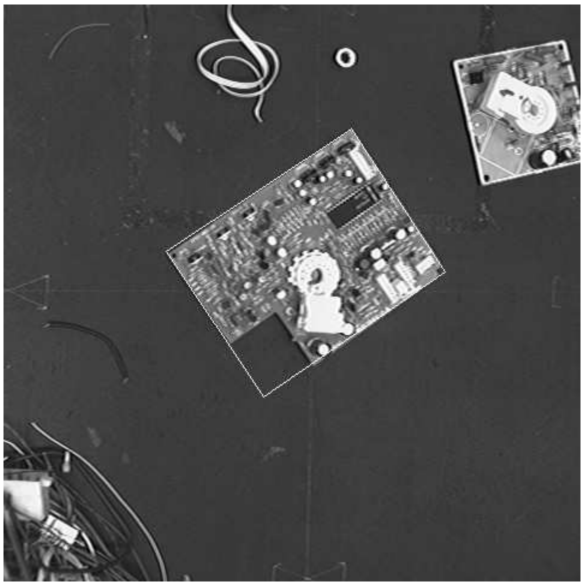
7.4 Objekterkennung

In diesem Kapitel sollen am Beispiel realer, komplexer Szenen einige Eigenschaften des Erkennungsverfahrens hervorgehoben werden. Die in Bezug auf die Zielstellung sowohl relevanten als auch irrelevanten Objekte der Szenen heben sich durch ihre Inhomogenitäten der Grauwerte von den bisher überwiegend künstlichen Testmustern und -objekten ab.

Der Objektextraktion liegt eine auf der Hough-Transformation basierende Detektion von Bildgeraden zugrunde. Wie bereits erwähnt, ist dieses Verfahren global. Folglich braucht die Kontur eines Objektes, welches lokalisiert werden soll, nicht vollständig als orientierte Grauwertkante in den Bildern der Kameras vorhanden sein. Somit ist die Objekterkennung nicht nur robust gegenüber Rauschen, sondern auch bei teilweiser Okklusion oder



(a) Okklusion



(b) Grobe Objektmodellierung

Abbildung 7.8: Erkennung von Objekten bei unvollständiger Kontur in den Ansichten

unpräziser Modellierung der 3D-Oberflächen möglich. Ursachen für Okklusionen sind das Übereinanderliegen der Objekte in der Szene sowie Verdeckungen aufgrund der Kameraperspektiven. Die Erkennung eines Objektes setzt insbesondere ein korrektes Matching der Berandungsgeraden in den Ansichten voraus. Das in Kapitel 5.2 beschriebene Verfahren des punktweisen Matchings von Geraden ermöglicht wegen der Votierung (Algorithmus 3) auch bei teilweisen Okklusionen die Berechnung der korrekten Geradenkorrespondenzen. Beispielsweise wird in der in Abbildung 7.8(a) dargestellten Szene die größere der beiden Platinen trotz der Okklusion erkannt. Die erkannten Objekte wurden durch die Projektionen der 3D-Oberflächenberandung in die abgebildete Ansicht markiert.

Liegt ein Modell eines Objektes vor, so stellt sich bezüglich der Wahl eines geeigneten Erkennungsverfahrens die Frage, wie detailliert eine mit dem vorgegebenen Modell vergleichbare Objektrepräsentation sein muß. Je detaillierter eine Objektbeschreibung durch ein Modell ist, desto aufwendiger und instabiler werden die Verfahren. Von Interesse ist also eine möglichst grobe Objektmodellierung. Abbildung 7.8(b) zeigt das Ergebnis des in dieser Arbeit vorgestellten Verfahrens bei Vorgabe des auf dem äußeren Rechteck basierenden Oberflächenmodells der erkannten Platine.

Abbildung 7.9, bei der zwecks besserer Darstellung der den Oberflächenmodellen entsprechenden Objektberandungen die Grauwerte halbiert wurden, zeigt ein Beispiel für die Identifikation zweier Objektoberflächen. Die Oberflächenrepräsentationen der in der Mitte befindlichen Platinen wurden durch das Verfahren extrahiert. Durch Vergleich der Oberflächen mit den vorgegebenen Modellen der beiden Platinen durch das Distanzmaß

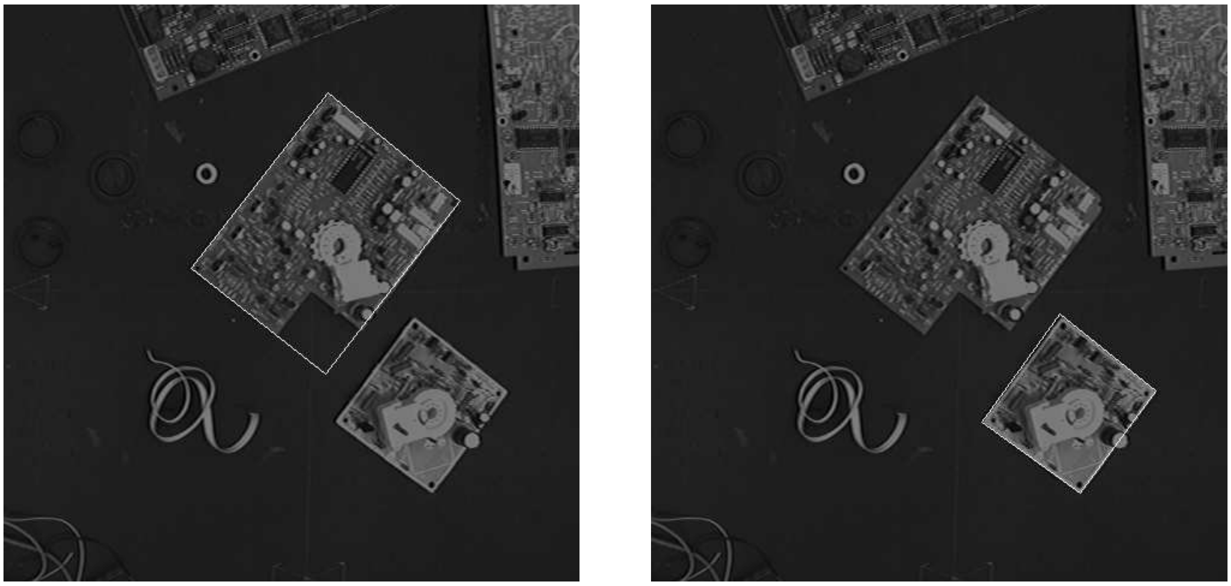
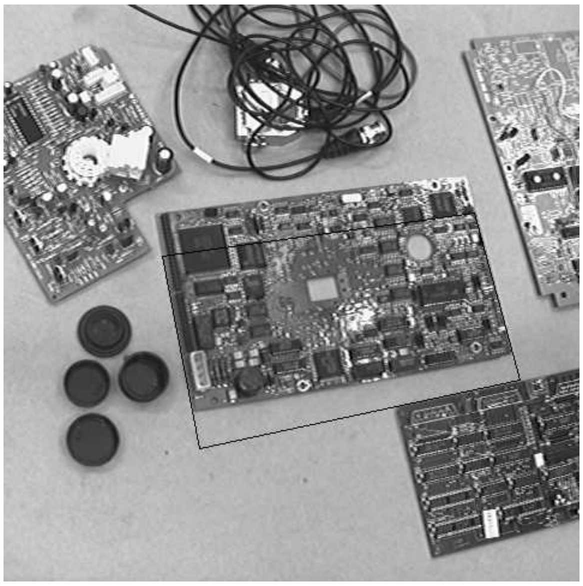


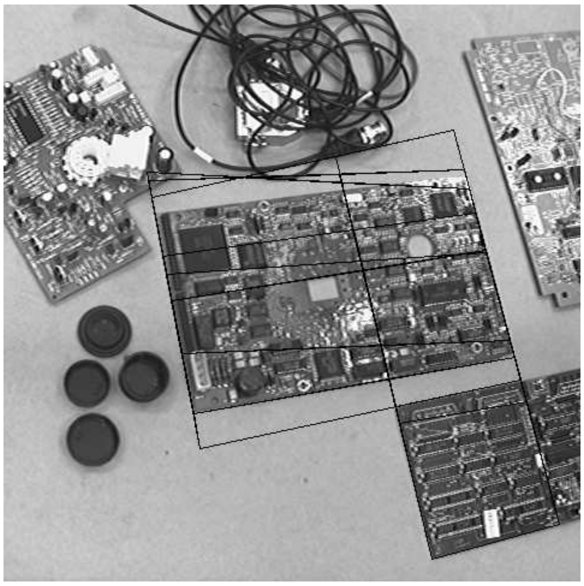
Abbildung 7.9: Identifikation zweier Objekte in derselben Szene

(Gleichung (5.113)) lassen sich die beiden Objekte voneinander unterscheiden und jeweils eindeutig bestimmen.

Es sollen an dieser Stelle auch Grenzen des Erkennungsverfahrens aufgezeigt werden. Abbildung 7.10(a) zeigt das Ergebnis der Oberflächenlokalisation für die dargestellte Szene. Es ist erkennbar, daß nicht die gewünschte 3D-Oberfläche lokalisiert wurde, sondern eine bezüglich der äußeren Kontur der Platine sehr ähnliche. Das Hauptproblem liegt bei komplexen Szenen in der Geradendetektion, und zwar in der Suche der lokalen Maxima im Parameterraum. Die Globalität der Hough-Transformation impliziert zwar Robustheit, jedoch führt das Vorhandensein vieler verschieden orientierter Grauwertkanten im Bild zu diffusen, breiten Peaks, so daß die Bildgeraden ungenau oder gar nicht detektiert werden. Letzterer Fall tritt ein, wenn Peaks im Akkumulatorfeld verschmelzen. Ein weiterer Grund für die ungenauere Detektion von Bildgeraden liegt darin, daß die Grauwertkanten in realen Szenen wesentlich flacher sind als bei den „künstlichen“ Testmustern. Um flache Kanten besser detektieren zu können, wäre es sinnvoll, neben dem Sobeloperator auch andere orientierungsselektive Kantendetektoren zu testen (siehe z. B. [14, Kap. 11.5.3]). Der Sobeloperator zeichnet sich durch die geringe Maskengrößen der horizontalen und der vertikalen Sobelmasken aus, was einerseits zu einem geringen Rechenaufwand führt, andererseits zu einer erhöhten Sensitivität gegenüber Rauschen. Somit sind auch die Rekonstruktionsfehler bei Bildern realer Szenen etwas höher.
Der Grund für die Fehldetektion in Abbildung 7.10(a) liegt darin, daß die obere Berandungsgerade der mittleren Platine in einer Ansicht nicht detektiert wurde. Abbildung 7.11(a) zeigt die detektierten Geraden im Bild der linken Kamera. Es ist erkennbar, daß die obere Berandungsgerade der mittleren Platine fehlt. Somit wird die Oberflächenrepräsentation bestimmt, die zum vorgegebenen Modell der Platine am ähnlichsten ist (siehe



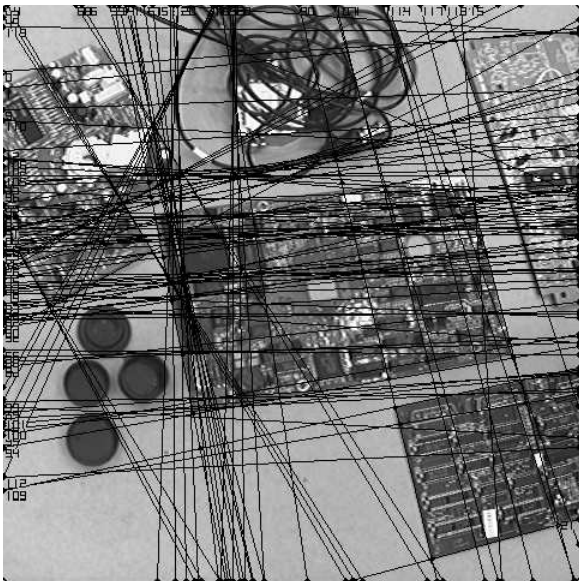
(a) das zum Modell ähnlichste Objekt



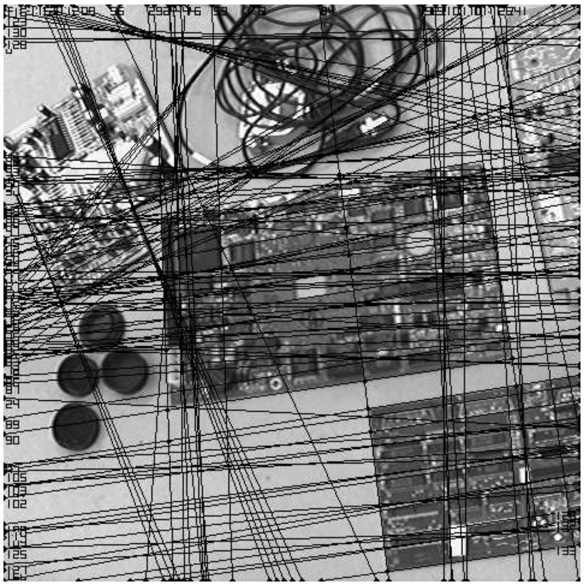
(b) Menge zum Modell ähnlicher Objekte

Abbildung 7.10: Ergebnis für weitere Testszene

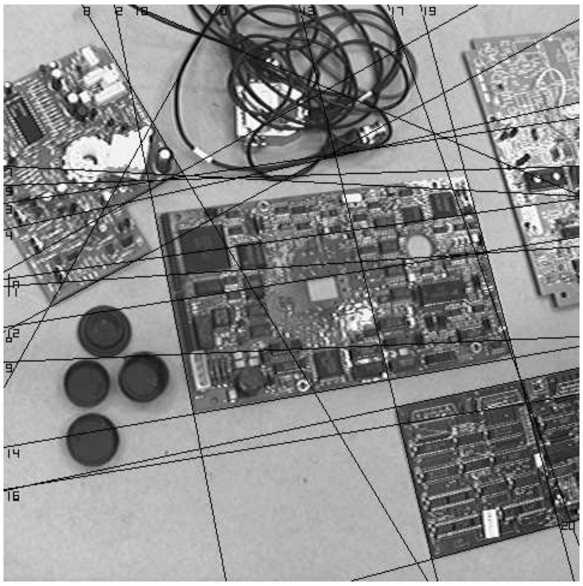
Abb. 7.10(a)). Abbildung 7.10(b) zeigt die zum Modell innerhalb einer Schwelle ähnlichen, lokalisierten Oberflächen. Offensichtlich wurden die der unteren Berandung der mittleren Platine entsprechenden Bildgeraden in den drei Ansichten korrekt detektiert und gematcht. Der Grund, warum die lokalisierte 3D-Oberfläche nicht von der „unteren" Berandung der Platine begrenzt wird, liegt also allein an dem auf Form und Größe basierenden Vergleich der 3D-Oberflächen, der jedoch zur Auflösung von Mehrdeutigkeiten bei der Lokalisation notwendig ist.
Die Szene in Abbildung 7.11 illustriert außerdem das Ergebnis des Matchingverfahrens. Aus Platzgründen sind nur die ersten beiden der drei Ansichten dargestellt. In realen Szenen werden im allgemeinen sehr viele Geraden detektiert, somit müssen entsprechend viele Korrespondenzen berechnet werden. Die Abbildungen 7.11(c) und 7.11(d) zeigen die Korrespondenzen der Bildgeraden in den ersten beiden Ansichten, deren Rekonstruktionen Schnitte mit anderen Geraden unter einem Winkel von 90° aufweisen. Die im Sinne der Vorauswahlkriterien relevanten, detektierten Bildgeraden wurden korrekt gematcht. Insbesondere sind mit den in Kapitel 5.5.1 beschriebenen Kriterien eine große Anzahl irrelevanter 3D-Geraden ausschließbar, was den Aufwand für die Suche der Kreise im Szenengraphen offensichtlich erheblich reduziert.



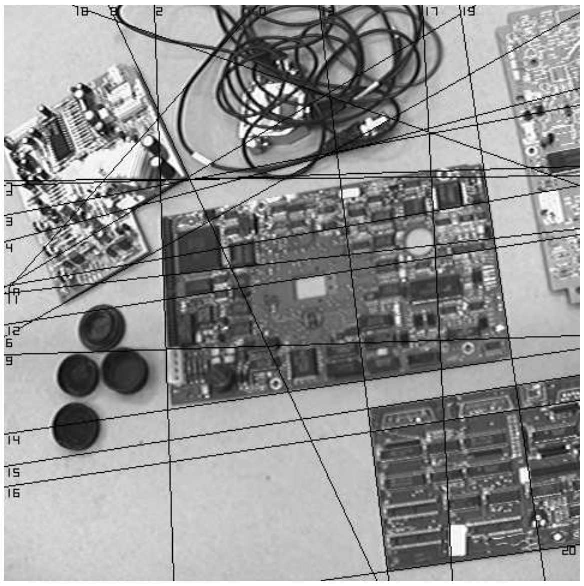
(a) detektierte Bildgeraden (linke Kamera)



(b) detektierte Bildgeraden (rechte Kamera)



(c) korrespondierende und selektierte Geraden (linke Kamera)



(d) korrespondierende und selektierte Geraden (rechte Kamera)

Abbildung 7.11: Detektierte Geraden sowie korrespondierende und selektierte Geraden in den ersten beiden Ansichten

Kapitel 8

Erscheinungsbasierte, neuronale Objekterkennung

In diesem Kapitel erfolgt die Beschreibung des Verfahrens zur Erkennung der Objekte nach dem erscheinungsbasierten, neuronalen Ansatz. Die Lokalisation und Identifikation der Objekte erfolgt durch gelernte, nichtlineare, verschiebungsinvariante Operatoren. Die Trainingsdaten sind Ansichten der 6 Objekte (Spannungsregler, Kondensator, Widerstand, IC, Transistor, Potentiometer) in jeweils verschiedenen Orientierungen. Für jedes Objekt wird ein Operator trainiert, dessen wesentliche Komponente ein RBF-Netz ist.

Die Vorverarbeitung der Eingabedaten des Netzes besteht in Dimensionsreduktionen und einer objektspezifischen Normierung. Es werden zwei Formen der Dimensionsreduktion eingesetzt. Die erste besteht in der Verringerung der Abtastrate durch den Einsatz von Oktavpyramiden: der Gaußpyramide oder der Laplacepyramide. Letztere bewirkt nicht nur die Reduktion der Auflösung, sondern auch eine Bandpaßzerlegung. Gegenstand dieser Arbeit ist auch der Vergleich der Ergebnisse für beide Pyramiden. Die zweite Form der Dimensionsreduktion besteht in der Transformation der Daten in den durch Hauptkomponentenanalyse gewonnenen, objektspezifischen Eigenraum.

In Kapitel 8.1 wird kurz die Aufnahme der Objekte beschrieben. In Kapitel 8.2 erfolgt die Beschreibung der Vorverarbeitung. Dabei werden die für die Objekterkennung wichtigen Eigenschaften herausgestellt sowie die Wahl der a priori festzulegenden Parameter diskutiert. Kapitel 8.3 führt kurz in RBF-Netze ein und beschreibt die Verfahren zur Berechnung der Parameter der Netze. Danach wird auf die Wahl der Trainingsmuster für die verschiedenen Arten des Trainings (Kap. 8.4) und auf die Schwellen zur Klassifikation der Objekte (Kap. 8.5) eingegangen. Die Konstruktion des Bildoperators zur Erkennung erfolgt dann in Kapitel 8.6.

8.1 Aufnahme von Objektansichten

Die Gewinnung der Objektansichten zum Training erfolgt mittels eines Drehtellers, der über die serielle Schnittstelle eines Rechners angesteuert wird. Abbildung 8.1 zeigt die

Anordnung der Hardware zur Aufnahme der Ansichten. Die optische Achse der Kamera stimmt – so gut wie möglich – mit der Rotationsachse des Drehtellers überein. Von den 6 Objekten (Spannungsregler, Kondensator, Widerstand, IC, Transistor und Potentiometer) wurden jeweils 360 Bilder mit einer Winkeldifferenz des Drehtellers von $\delta_\phi = 1°$ zwischen den Ansichten aufgenommen. Die Bilder haben eine Größe von 128×128 Pixel. Die Objekte befinden sich auf möglichst homogenem Hintergrund. Abbildung 8.2 zeigt eine Teilmenge der Ansichten.

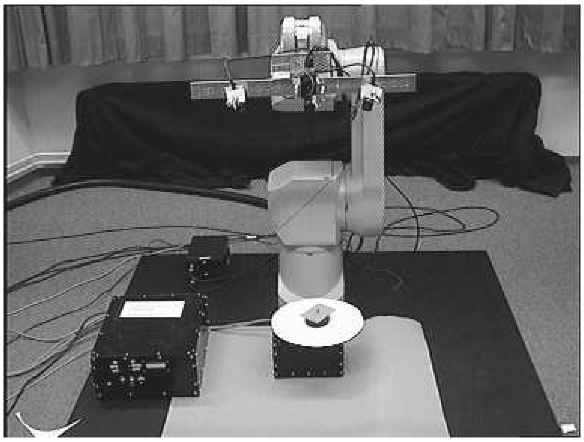


Abbildung 8.1: Aufbau zur Aufnahme der Objektansichten

8.2 Vorverarbeitung

8.2.1 Auflösungshierarchien

Da die Kamera kein Zoom besitzt, haben die Bilder der Objekte eine feste Größe, die allein durch den Abstand zwischen Bildebene und Gegenstandsebene und der Brennweite des Objektivs bestimmt wird. Die Bilder müssen auf eine für die Objekterkennung geeignete Größe skaliert werden. Dieses geschieht durch Reduktion der Abtastrate und der Auflösung unter Einhaltung des Abtasttheorems, d. h. die Abtastrate, mit der die Unterabtastung erfolgt, muß mindestens doppelt so groß sein wie die obere Grenzfrequenz. Die Unterabtastung erfolgt am effizientesten durch die Gaußpyramide [5]. Die unterste Ebene der Oktavpyramide stellt die verwendeten Bilder zur Verfügung. Dabei muß die Unterabtastung so weit erfolgen, daß sich die Bilder bzw. Objektansichten bezüglich einer Klassifikation hinreichend unterscheiden.
Mit der Gaußpyramide ist die Laplacepyramide eng verbunden. Sie bewirkt eine Bandpaßzerlegung des Bildes. Den Kanten auf einer Ebene der Gaußpyramide entsprechen Nulldurchgänge in der gleichen Ebene der Laplacepyramide. Obwohl in den Bildern Kanten nicht explizit detektiert werden, sind sie im Falle der Laplacepyramide das für die Objekterkennung signifikante Merkmal.

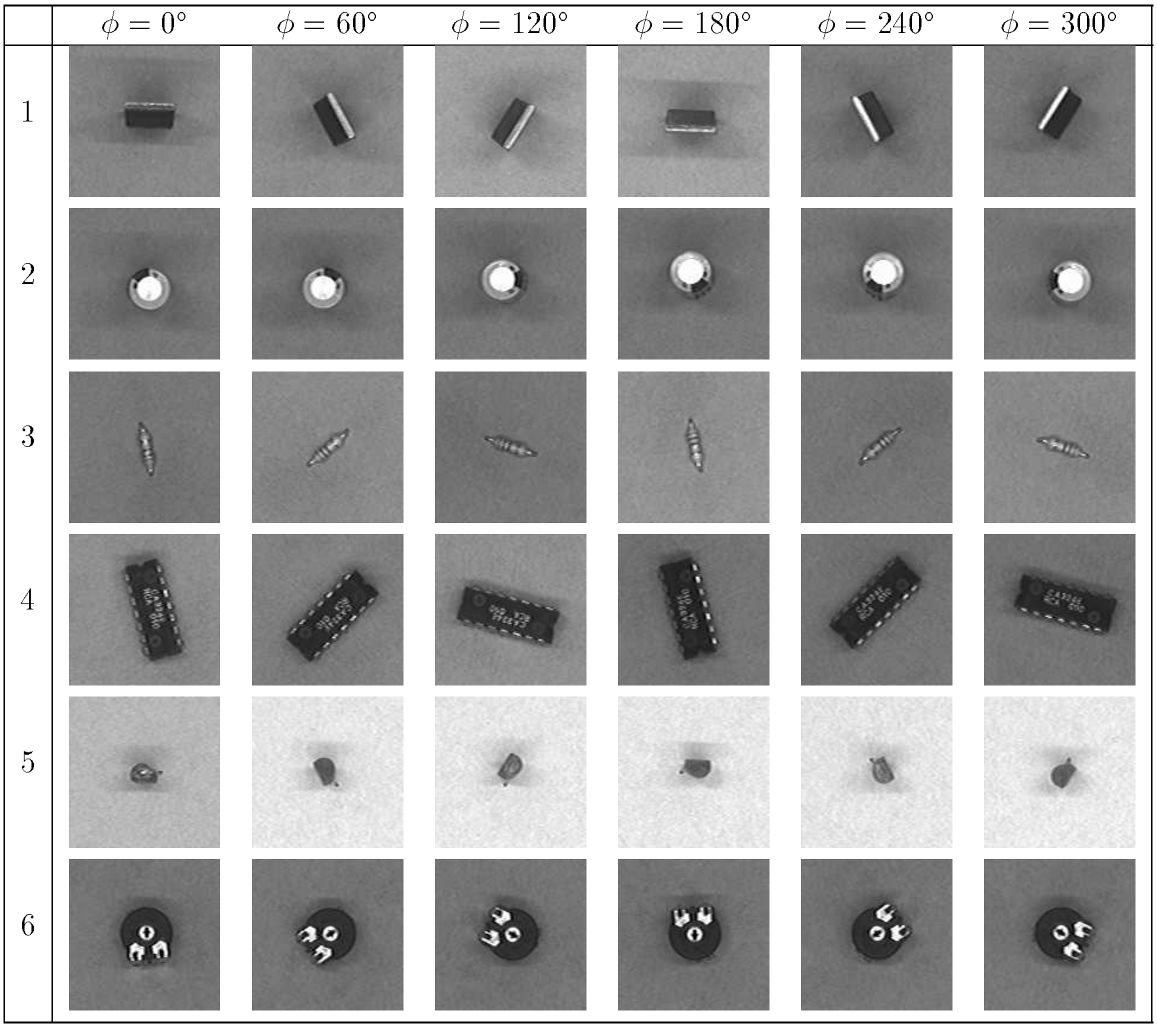


Abbildung 8.2: Eine Teilmenge der Objektansichten

8.2.1.1 Gaußpyramide und Laplacepyramide

Im folgenden werden kurz die Reduktions- und die Expansionsoperation beschrieben.

Die Reduktionsoperation $\mathcal{R}$ Das unterabgetastete Bild entsteht durch gleichzeitige Glättung und Unterabtastung. Es bezeichne $g^{[l-1]}$ das Bild mit höherer Abtastrate, $g^{[l]}$ das davon ausgehend mit halber Abtastrate unterabgetastete Bild. Nach [5] erfolgt die Unterabtastung gemäß

$$g_{i,j}^{[l]} = \sum_m \sum_n w_{m,n} g_{2i+m,2j+n}^{[l-1]} \tag{8.1}$$

Die Maske w wird auch der erzeugende Kern der Pyramide genannt. Sie bewirkt die zur Einhaltung des Abtasttheorems notwendige Glättung und wird derart gewählt, daß eine eindimensionale Maske $\breve{w}$ mit den folgenden Eigenschaften existiert:

1. Separierbarkeit: $w_{m,n} = \breve{w}_m \breve{w}_n$
2. Normiertheit: $\sum_m \breve{w}_m = 1$
3. Symmetrie: $\breve{w}_m = \breve{w}_{-m}$
4. Gleichmäßigkeit: $\sum_m \breve{w}_{2m} = \breve{w}_{2m+1}$

Die Faltungsmaske soll von der Dimension 5×5 sein, d. h. $w_{m,n} = 0$ für $|m| > 5$ und $|n| > 5$. Aus der Separierbarkeit folgt $\breve{w}_k = 0$ für $k > 5$. Aufgrund der Symmetrie gilt: $\breve{w}_2 = \breve{w}_{-2} = c$, $\breve{w}_1 = \breve{w}_{-1} = b$, $\breve{w}_0 = a$ für $a, b, c \in \mathbb{R}$. Aus der Normiertheit und der Gleichmäßigkeit erhält man [5]:

$$a \in \mathbb{R}, \quad b = \frac{1}{4}, \quad c = \frac{1}{4} - \frac{a}{2} \tag{8.2}$$

Die Form des erzeugenden Kerns w wird durch die Wahl von a bestimmt. Da die Gaußmaske die besten Glättungseigenschaften besitzt, wird a derart gewählt, daß $\breve{w}$ die Gaußsche Wahrscheinlichkeitsdichtefunktion approximiert. Nach [6] ist die beste Approximation durch $a = 0.4$ gegeben. Somit erhält man insgesamt

$$a = 0.4, \quad b = 0.25, \quad c = 0.05 \tag{8.3}$$

Aufgrund der Separierbarkeit der Maske und deren Symmetrie läßt sich die Reduktion (8.1) eines Bildes durch eine Reduktion in vertikaler Richtung und der anschließenden Reduktion in horizontaler Richtung effizienter durchführen:

1. Reduktion in vertikaler Richtung:

$$\begin{aligned} g_{i,j}^{[l]} &= \sum_{m=-2}^{2} \breve{w}_m g_{2i+m,j}^{[l-1]} \\ &= a \cdot g_{2i,j}^{[l-1]} + b[g_{2i-1,j}^{[l-1]} + g_{2i+1,j}^{[l-1]}] + c[g_{2i-2,j}^{[l-1]} + g_{2i+2,j}^{[l-1]}] \end{aligned} \tag{8.4}$$

2. Reduktion in horizontaler Richtung:

$$\begin{aligned} g_{i,j}^{[l]} &= \sum_{n=-2}^{2} \breve{w}_n g_{2i,2j+n}^{[l-1]} \\ &= a \cdot g_{i,2j}^{[l-1]} + b \cdot [g_{i,2j-1}^{[l-1]} + g_{i,2j+1}^{[l-1]}] + c \cdot [g_{i,2j-2}^{[l-1]} + g_{i,2j+2}^{[l-1]}] \end{aligned} \tag{8.5}$$

Abbildung 8.3(a) veranschaulicht die Reduktionsoperation.

Die Expansionsoperation $\mathcal{E}$ Nach [5] wird das Bild der Ebene l auf die doppelte Größe in der Ebene $l-1$ in folgender Weise expandiert:

$$g_{i,j}^{[l-1]} = 4 \sum_{\substack{m,n \\ i+m \text{ gerade} \\ \wedge j+n \text{ gerade}}} w_{m,n} g^{[l]}_{\frac{i+m}{2},\frac{j+n}{2}} \tag{8.6}$$

Die Expansionsoperation ist ebenfalls separierbar und somit effizienter durchführbar:

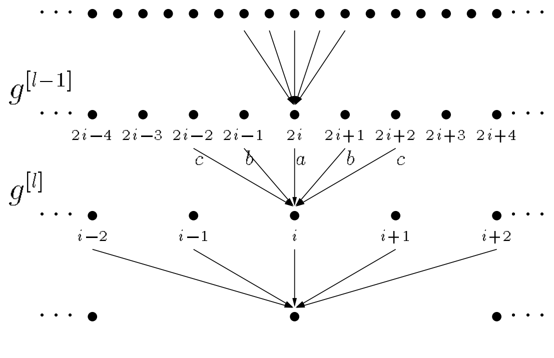
1. Expansion in horizontaler Richtung:

$$\begin{aligned} g_{i,j}^{[l-1]} &= \begin{cases} 2\sum_{n\in\{-2,0,2\}} w_n g^{[l]}_{\frac{i}{2},\frac{j+n}{2}} & \text{, falls } j \text{ gerade} \\ 2\sum_{n\in\{-1,1\}} w_n g^{[l]}_{\frac{i}{2},\frac{j+n}{2}} & \text{, falls } j \text{ ungerade} \end{cases} \\ &= \begin{cases} 2(a \cdot g^{(l)}_{\frac{i}{2},\frac{j}{2}} + c(g^{[l]}_{\frac{i}{2},\frac{j}{2}-1} + g^{[l]}_{\frac{i}{2},\frac{j}{2}+1})) & \text{, falls } j \text{ gerade} \\ 2b(g^{[l]}_{\frac{i}{2},\frac{j-1}{2}} + g^{[l]}_{\frac{i}{2},\frac{j+1}{2}}) & \text{, falls } j \text{ ungerade} \end{cases} \end{aligned} \tag{8.7}$$

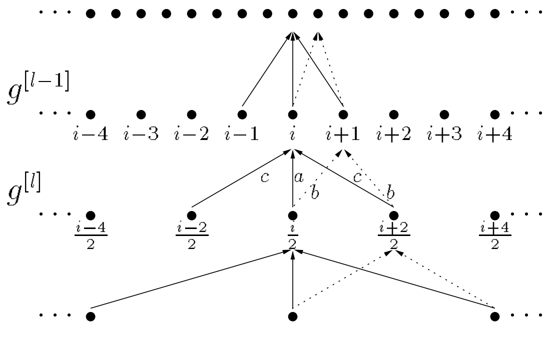
2. Expansion in vertikaler Richtung:

$$\begin{aligned} g_{i,j}^{[l-1]} &= \begin{cases} 2\sum_{m\in\{-2,0,2\}} w_m g^{[l]}_{\frac{i+m}{2},\frac{j}{2}} & \text{, falls } i \text{ gerade} \\ 2\sum_{m\in\{-1,1\}} w_m g^{[l]}_{\frac{i+m}{2},\frac{j}{2}} & \text{, falls } i \text{ ungerade} \end{cases} \\ &= \begin{cases} 2(a \cdot g^{[l]}_{\frac{i}{2},\frac{j}{2}} + c(g^{[l]}_{\frac{i}{2}-1,\frac{j}{2}} + g^{[l]}_{\frac{i}{2}+1,\frac{j}{2}})) & \text{, falls } i \text{ gerade} \\ 2b(g^{[l]}_{\frac{i-1}{2},\frac{j}{2}} + g^{[l]}_{\frac{i+1}{2},\frac{j}{2}}) & \text{, falls } i \text{ ungerade} \end{cases} \end{aligned} \tag{8.8}$$

Abbildung 8.3(b) veranschaulicht die Expansionsoperation.

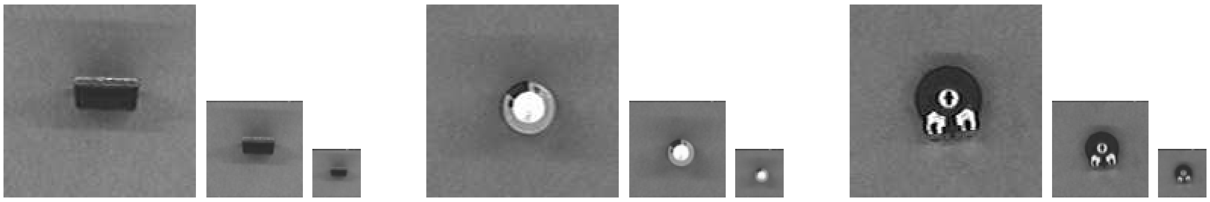


(a) Reduktion



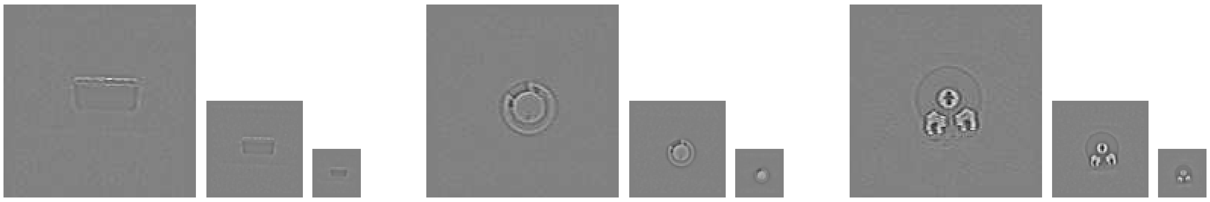
(b) Expansion

Abbildung 8.3: Graphische Darstellung der Reduktion und der Expansion



(a) (b) (c)

Abbildung 8.4: Gaußpyramiden



(a) (b) (c)

Abbildung 8.5: Laplacepyramiden

Gaußpyramide Die Gaußpyramide $g^{[0]}, \ldots g^{[n-1]}$ mit n Ebenen entsteht durch sukzessive Reduktion eines Bildes g:

$$g^{[0]} = g \tag{8.9}$$

$$g^{[l]} = \mathcal{R} g^{[l-1]}, \quad l = 1, \ldots n-1 \tag{8.10}$$

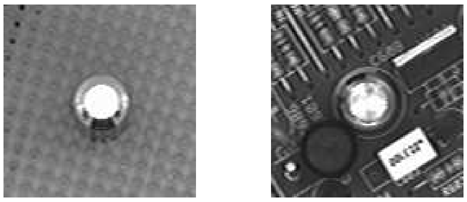
Laplacepyramide Die Laplacepyramide $h^{[0]}, \ldots h^{[n-1]}$, mit n Ebenen erhält man aus einer Gaußpyramide mit $n+1$ Ebenen gemäß

$$h^{[l-1]} = g^{[l-1]} - \mathcal{E} g^{[l]}, \quad l = 1, \ldots, n-1 \tag{8.11}$$

Abbildung 8.4 zeigt die Gaußpyramiden dreier Objekte, Abbildung 8.5 die Laplacepyramiden. Es ist deutlich das Verschwinden von Details bei zunehmender Unterabtastung sichtbar.

8.2.2 Normierung der Daten

Die Objekterkennung soll möglichst invariant gegenüber Beleuchtung sein. Unter der Annahme, daß eine lineare Beziehung zwischen der Helligkeit der Lichtquelle und den Grauwerten besteht, führt eine Normierung zweier Bilder, die mit der gleichen Anordnung von



(a) (b)

Abbildung 8.6: Beispiele, bei denen eine Normierung gemäß (8.12) ungeeignet ist

Kamera und Lichtquelle, jedoch mit unterschiedlicher Lichtintensität aufgenommen wurden, zu näherungsweise gleichen Bildsignalen. Um die Abweichung der beiden normierten Bilder möglichst gering zu halten, sollte der Rauschanteil des Signals möglichst niedrig sein. Ebenso dürfen die Objekte nicht metallisch sein, da der Grauwert in Glanzpunkten – fast unabhängig von der Helligkeit – sehr hoch ist und die beiden normierten Bilder an diesen Punkten besonders starke Abweichungen aufweisen.

In den Arbeiten von Murase und Nayar [20, 21] wird das Bildsignal durch die gesamte Energie des Bildes geteilt:

$$f = \frac{g}{\|g\|} \quad \text{mit} \quad \|g\| = \sum_{i,j}^{M} {g_{i,j}}^2 \tag{8.12}$$

wobei $M \times M$ die Größe des Bildes ist.

Diese Normierung ist geeignet, wenn sich die Bildobjekte sowohl in der Trainings- als auch in der Erkennungsphase auf homogenem Hintergrund befinden, wie es in den Arbeiten von Murase und Nayar der Fall ist.

Sollen jedoch in der Anwendungsphase Elektronikbauteile auf einer Platine erkannt werden, so befinden sich diese Objekte auf inhomogenem Hintergrund. Hohe Werte im Bild, die außerhalb des Bildbereiches des Objektes liegen, führen zur Verkleinerung der Werte in den Pixeln, die das gesuchte Objekt repräsentieren. Beispiele sind in Abbildung 8.6 dargestellt. Das gesuchte Objekt – in diesem Fall der Kondensator – befindet sich in dem quadratischen Ausschnitt auf inhomogenem Hintergrund, sei es durch Textur auf der Platine (Abb. 8.6(a)) oder durch weitere Objekte (Abb. 8.6(b)). Dabei ist es gleich, ob die Gaußpyramide oder die Laplacepyramide eingesetzt wurde. In jedem Fall wird das Objekt nicht erkannt.

Daher erfolgt nicht die Division des Bildsignals durch die Energie des gesamten quadratischen Bildausschnitts, sondern durch die Energie des „relevanten“ Bildbereiches für das jeweilige Objekt. Dieser Bildbereich wird durch einen Aperturoperator bestimmt, der sich in der Trainingsphase sehr einfach auf folgende Weise erzeugen läßt:

1. Berechne für (alle) N Ansichten eines Objektes das Bild auf der Ebene der Laplacepyramide, auf der die Objekterkennung stattfinden soll. Die resultierenden Bilder

seien mit $l^{(k)}$, $k = 1, \dots N$, bezeichnet.

2. Wende auf jedes $l^{(k)}$, $k = 1, \dots, N$ einen durch θ_a bestimmten Schwellwertoperator an:

$$a_{i,j}^{(k)} = \begin{cases} 1 & \text{, falls } |l_{i,j}^{(k)}| \geq \theta_a \\ 0 & \text{, sonst} \end{cases} \tag{8.13}$$

   Insbesondere sind dann Pixel im Binärbild, die dem homogenen Objekthintergrund in einer Ansicht entsprechen, auf 0 gesetzt.

3. Durch Disjunktion über alle Ansichten erhält man den Bereich, in dem das Objekt in einer Ansicht liegen kann:

$$a_{i,j} = \bigvee_{k=1}^{n} a_{i,j}^{(k)} \tag{8.14}$$

4. Um das Ausblenden kleinerer Werte in der Umgebung des Bildobjektes zu verhindern, wird das Bild a mit dem in Abbildung 8.7 dargestellten Strukturelement dilatiert.

Für jedes Objekt wird so ein Aperturoperator $\mathcal{A}$ berechnet. Abbildung 8.8 zeigt die Aperturoperatoren für die Objekte. Da bei der Aufnahme der Trainingsansichten die Rotationsachse und die optische Achse fast gleich orientiert sind, ergeben sich kreisförmige Aperturfenster. Die Normierung eines Bildes g erfolgt dann gemäß

$$n = \frac{g}{\|g\|_{\mathcal{A}}} \qquad \text{mit} \quad \|g\|_{\mathcal{A}} := \sum_{i,j} a_{i,j} g_{i,j} \tag{8.15}$$

Im weiteren Verlauf wird, sowohl in der Trainings- als auch in der Anwendungsphase, nur diese Norm verwendet.

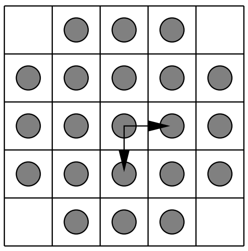
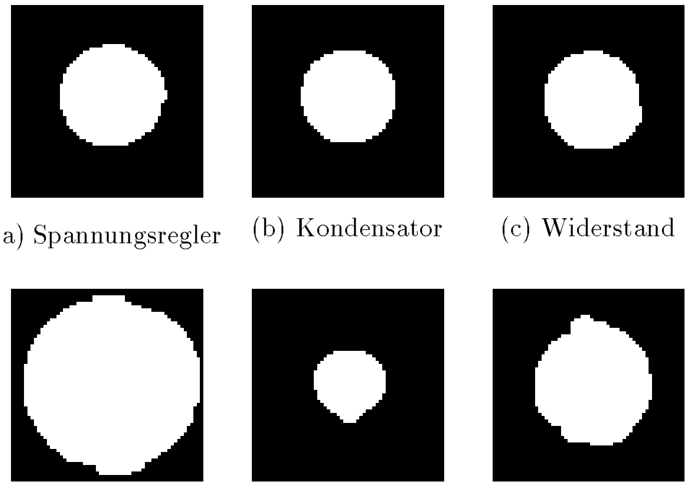


Abbildung 8.7: Strukturelement



(a) Spannungsregler (b) Kondensator (c) Widerstand

(d) IC (e) Transistor (f) Potentiometer

Abbildung 8.8: Aperturfenster der 6 Objekte

8.2.3 Hauptkomponentenanalyse

Die vorverarbeiteten Ansichten eines Objektes sind stark korreliert. Ziel der Hauptkomponentenanalyse ist die Berechnung eines orthogonalen Basissystems, dessen Basisvektoren die charakteristischen Erscheinungen der Menge der Ansichten sind. Aufgrund der hohen Korrelation der Daten ist die Dimension des so entstandenen Vektorraumes deutlich geringer als die Größe des Bildes. Die Basisvektoren dieses Vektorraumes sind die zu einer Menge größter Eigenwerte gehörenden Eigenvektoren der aus den normierten Bildvektoren gebildeten Kovarianzmatrix. Daher wird er auch als *Eigenraum* bezeichnet [36, 23, 20].
Bei der Normierung werden unterschiedliche Aperturoperatoren eingesetzt. Daher wird für jedes Objekt ein Eigenraum aus den Ansichten berechnet.

Für die Hauptkomponentenanalyse müssen die Bilder in Vektoren überführt werden. Dieses geschieht durch zeilenweise Abtastung des Bildes[1]:

$$\boldsymbol{x} = (f_{0,0}, \ldots, f_{0,M-1}, f_{1,0}, \ldots, f_{1,M-1}, \ldots, f_{M-1,0}, \ldots, f_{M-1,M-1})^T \in \mathbb{R}^n \tag{8.16}$$

mit $n = (M-1)^2$. Ist $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ die Menge von Vektoren, die der Hauptkomponentenanalyse unterzogen werden sollen, so sei zur Vereinfachung der Notation die Matrix

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1 - \boldsymbol{\mu} & \cdots & \boldsymbol{x}_N - \boldsymbol{\mu} \end{pmatrix} \in \mathbb{R}^{n \times N} \tag{8.17}$$

mit dem Mittelwert

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \tag{8.18}$$

definiert. Die Kovarianzmatrix läßt sich dann in der Form

$$\boldsymbol{A} = \frac{1}{N} \boldsymbol{X} \boldsymbol{X}^T \in \mathbb{R}^{n \times n} \tag{8.19}$$

schreiben. Die Dimension dieser Matrix ist sehr hoch, wodurch die Berechnung der Eigenwerte und Eigenvektoren sehr aufwendig ist.

Ist die Dimension n der Vektoren wesentlich größer als die Anzahl N der Daten, so lassen sich die Eigenwerte und -vektoren der Matrix $\boldsymbol{X}$ aus der kleineren Matrix

$$\overline{\boldsymbol{A}} = \frac{1}{N} \boldsymbol{X}^T \boldsymbol{X} \tag{8.20}$$

berechnen [20]:

Genau dann ist λ ein Eigenwert von $\boldsymbol{A}$, wenn λ ein Eigenwert der Matrix $\overline{\boldsymbol{A}}$ ist. Ist $\overline{\boldsymbol{q}}$ mit $\|\overline{\boldsymbol{q}}\| = 1$ der Eigenvektor zum Eigenwert λ von $\overline{\boldsymbol{A}}$, so ist

$$\boldsymbol{q} = \frac{\boldsymbol{X}\overline{\boldsymbol{q}}}{\sqrt{\lambda N}} \tag{8.21}$$

[1] Da die Topologie des Bildes bei der Hauptkomponentenanalyse keine Rolle spielt, ist eine beliebige Bijektion zwischen Bild und Vektor verwendbar.

der Eigenvektor von $\boldsymbol{A}$ zum Eigenwert λ und es gilt $\|\boldsymbol{q}\| = 1$.

Die Berechnung einer Anzahl m größter reeller Eigenwerte $\lambda_1 > \lambda_2 > \cdots > \lambda_m > 0$ mit den entsprechenden Eigenvektoren $\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_m$ erfolgt mit dem in Anhang A beschriebenen Verfahren. Die Transformation der n-dimensionalen Vektoren in den m-dimensionalen Eigenraum eines Objektes erfolgt dann gemäß

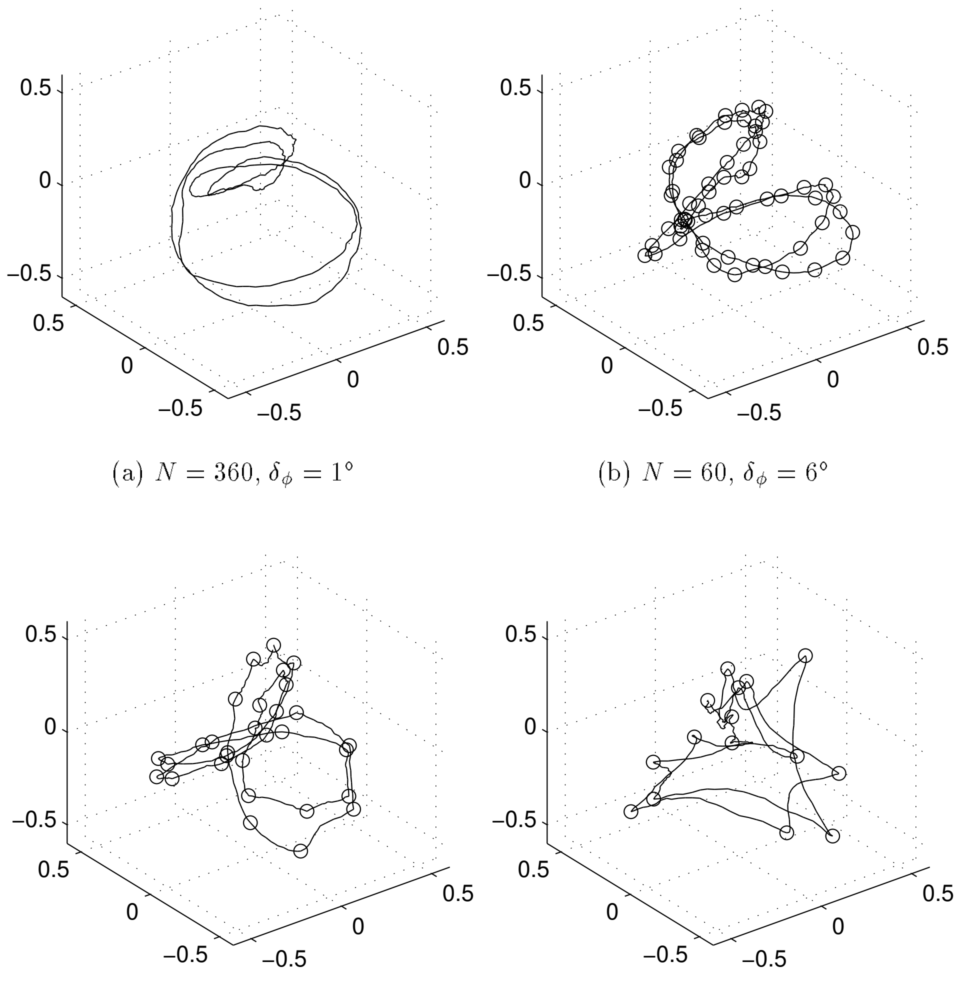
$$^{\varepsilon}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{q}_1{}^T \\ \vdots \\ \boldsymbol{q}_m{}^T \end{pmatrix} (\boldsymbol{x} - \boldsymbol{\mu}) \tag{8.22}$$

8.2.3.1 Anzahl der Ansichten

Es stellt sich die Frage, wieviele Ansichten eines Objektes aufgenommen werden müssen, damit eine Erkennung möglich ist. Eine allgemeingültige Gesetzmäßigkeit läßt sich nicht formalisieren. Es soll jedoch an einem Beispiel gezeigt werden, wie sich die Anzahl der Ansichten, aus denen die Kovarianzmatrix berechnet wird, auf die Projektion der Daten im Eigenraum auswirkt.
Abbildung 8.9 liegen die (normierten) Daten des Spannungsreglers der 1. Ebene der Laplacepyramide zugrunde. Das Koordinatensystem der dargestellten Eigenräume wird durch die jeweils ersten drei Eigenvektoren aufgespannt. Die Anzahl N der einbezogenen Ansichten ist $N = 360$ (Abb. 8.9(a)), $N = 60$ (Abb. 8.9(b)), $N = 30$ (Abb. 8.9(c)) und $N = 15$ (Abb. 8.9(d)), wobei die Winkeldifferenz δ_ϕ zwischen zwei aufeinanderfolgenden Ansichten in der jeweiligen Menge konstant ist. Abbildung 8.9(a) zeigt eine durch die Orientierung parametrisierbare Kurve. In dem Verlauf dieser Kurve spiegeln sich die Symmetrieeigenschaften des Objektes wieder. In den Abbildungen 8.9(b), 8.9(c) und 8.9(d) sind die Punkte, die den in die Berechnung einfließenden Ansichten entsprechen, durch das Symbol $\circ$ dargestellt. Die durchgezogene Kurve ergibt sich aus der Projektion aller 360 Ansichten in den Eigenraum.
Je weniger Ansichten in die Kovarianzmatrix einfließen, desto schlechter repräsentieren die berechneten Eigenvektoren die nicht in die Hauptkomponentenanalyse einfließenden Zwischenansichten. In diesem Beispiel ergibt sich, wie in den Abbildungen erkennbar ist, die stückweise Deformation der Kurven zum Ursprung hin. Die Kurve verliert an Glattheit, die Deformation der Kurven führt zu einer schlechteren Diskriminierbarkeit zwischen verschiedenen Objekten bei der sich anschließenden Objekterkennung. Außerdem gehen die im Verlauf der Kurve enthaltenen Symmetrieeigenschaften verloren.

8.2.3.2 Anzahl der Eigenvektoren

Die Dimension des Eigenraumes eines Objektes, in den die normierten Daten transformiert werden, wird durch die Anzahl m von Eigenvektoren $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m$ zu den größten Eigenwerten $\lambda_1 > \cdots > \lambda_m$ bestimmt. Je kleiner λ_i ist, desto feiner sind die Details, die



(a) $N = 360$, $\delta_\phi = 1°$ (b) $N = 60$, $\delta_\phi = 6°$

(c) $N = 30$, $\delta_\phi = 12°$ (d) $N = 15$, $\delta_\phi = 24°$

Abbildung 8.9: Wirkung der Anzahl von Ansichten bei der Hauptkomponentenanalyse auf die Transformation der Ansichten in den Eigenraum

durch $\boldsymbol{q}_i$ codiert werden (siehe Tabelle 8.1). Sowohl mit als auch ohne Bandpaßfilterung sind die Werte der Hauptkomponenten im Bereich des Bildhintergrundes sehr klein, was zu einem „Wegprojizieren“ der im allgemeinen inhomogenen Umgebung des Bildobjektes in der Anwendungsphase führt.

Die Frage ist nun, wie groß m gewählt werden soll. Einerseits wird eine möglichst kleine Fehlerrate bei der Objekterkennung angestrebt, andererseits soll die Erkennung aufgrund der charakteristischen Merkmale erfolgen. Um den Verlauf der Eigenwerte verschiedener

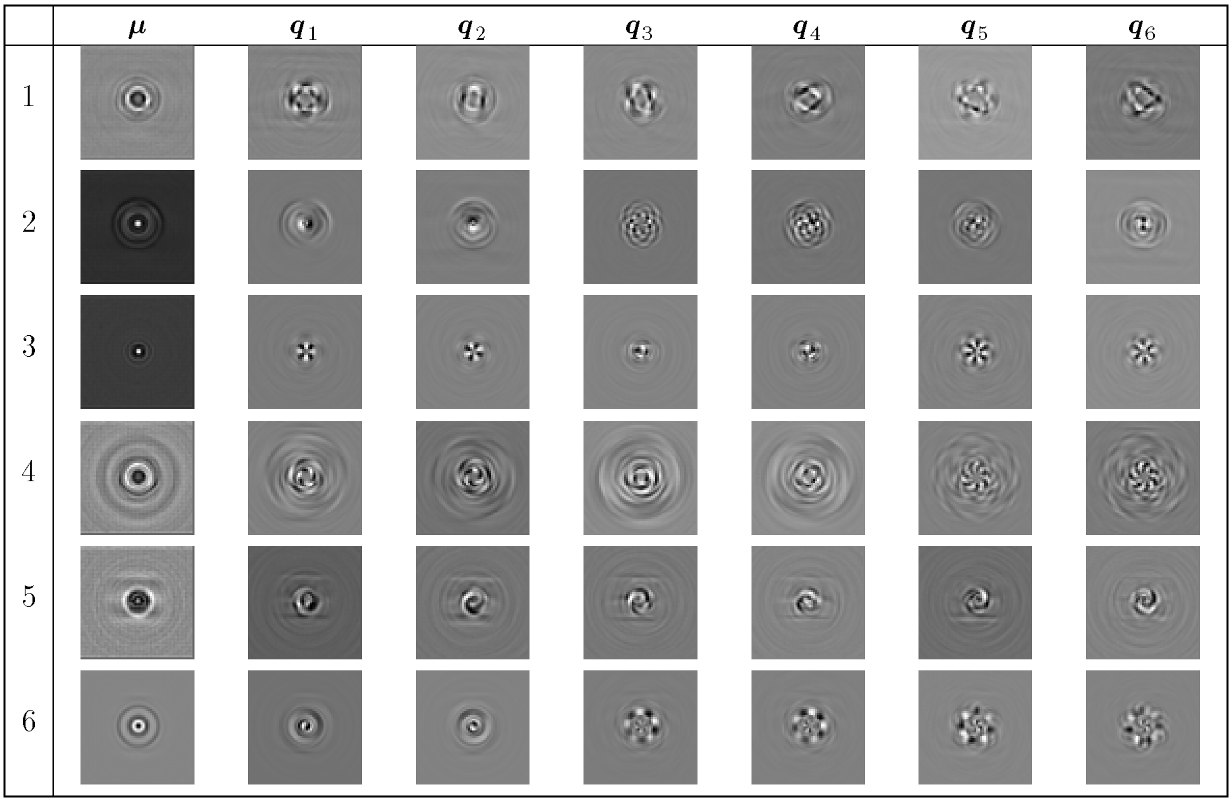


Tabelle 8.1: Die sich aus den Ansichten der Objekte jeweils ergebenden Mittelwerte und Eigenvektoren bei Bandpaßfilterung (Laplacepyramide)

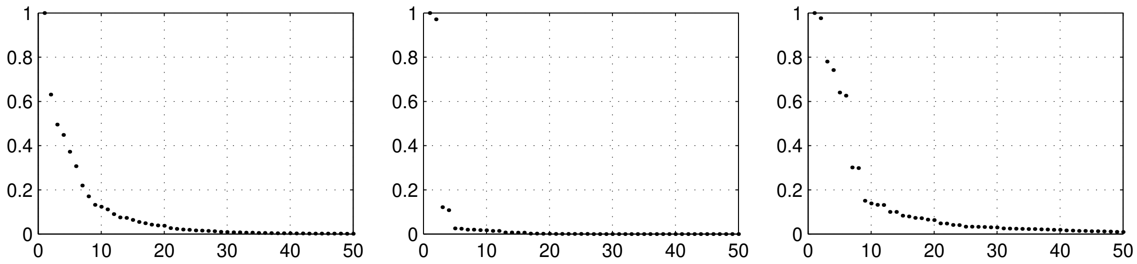
Objekte vergleichen zu können, werden die Verhältnisse

$$v_i = \frac{\lambda_i}{\lambda_1} \quad i = 1, \dots, m \tag{8.23}$$

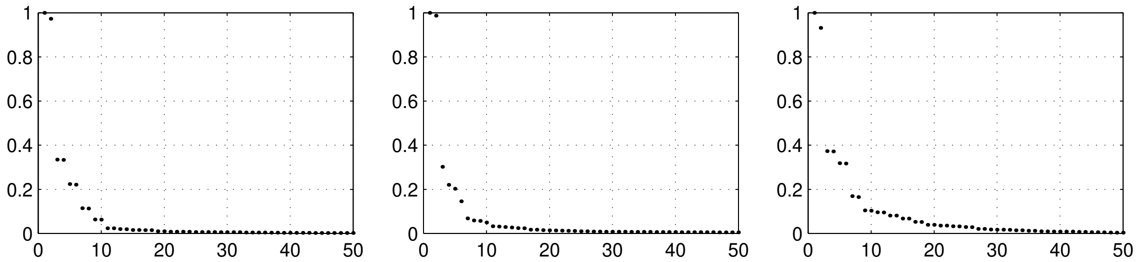
betrachtet.

Die Graphiken in Abbildung 8.10 zeigen die $v_i, i = 1, \dots, 50$ für die 6 verschiedenen Objekte, die sich jeweils durch Hauptkomponentenanalyse der normierten Bilddaten in der ersten Ebene der Gaußpyramide ergeben. Abbildung 8.11 zeigt die Ergebnisse bei Verwendung der Laplacepyramide. Es wird deutlich, daß im Falle der Laplacepyramide die Eigenwerte viel langsamer sinken, woraus sich eine höhere Anzahl notwendiger Eigenvektoren ergibt. Je korrelierter die Ansichten sind, desto schneller sinken die Eigenwerte. Beim Kondensator (Abb. 8.10(b) und Abb. 8.11(b)) ergibt sich die hohe Korrelation durch die Rotationssymmetrie des Objektes, beim Transistor (Abb. 8.10(e) und Abb. 8.11(e)) aus der Tatsache, daß das Objekt im Vergleich zum Hintergrund sehr klein ist.
Bei Verwendung der Gaußpyramide ergibt sich aufgrund des starken Abfalls der Eigenwerte eine maximale Dimension des Eigenraumes von etwa $m = 10$. Bei der Laplacepyramide ist nicht so leicht ersichtlich, wie m zu wählen ist.

Daher transformieren wir jeweils eine Ansicht in den Eigenraum und dann wieder zurück.

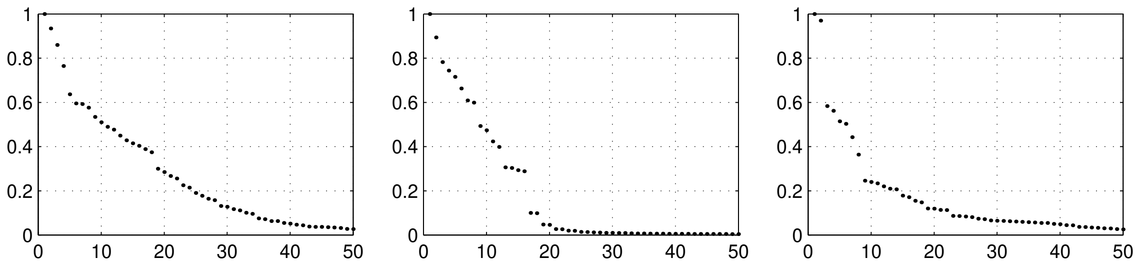


(a) Regler, $\lambda_1 = 0.0138$ (b) Kondensator, $\lambda_1 = 0.0499$ (c) Widerstand, $\lambda_1 = 0.0013$

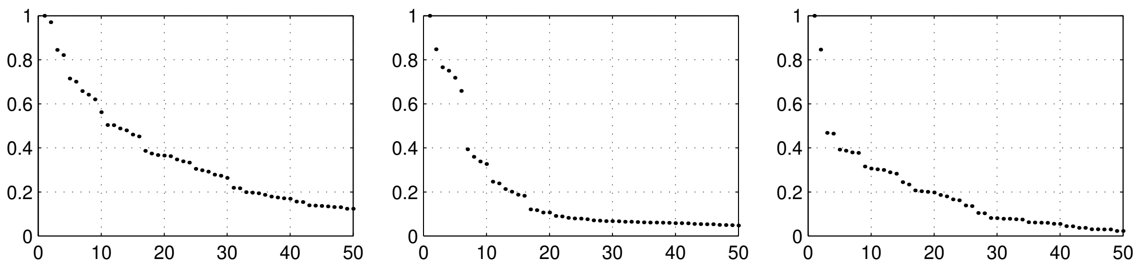


(d) IC, $\lambda_1 = 0.0156$ (e) Transistor, $\lambda_1 = 0.0049$ (f) Potentiometer, $\lambda_1 = 0.0202$

Abbildung 8.10: Eigenwerte (Gaußpyramide)



(a) Regler, $\lambda_1 = 0.0682$ (b) Kondensator, $\lambda_1 = 0.0950$ (c) Widerstand, $\lambda_1 = 0.1$



(d) IC, $\lambda_1 = 0.0445$ (e) Transistor, $\lambda_1 = 0.1099$ (f) Potentiometer, $\lambda_1 = 0.0687$

Abbildung 8.11: Eigenwerte (Laplacepyramide)

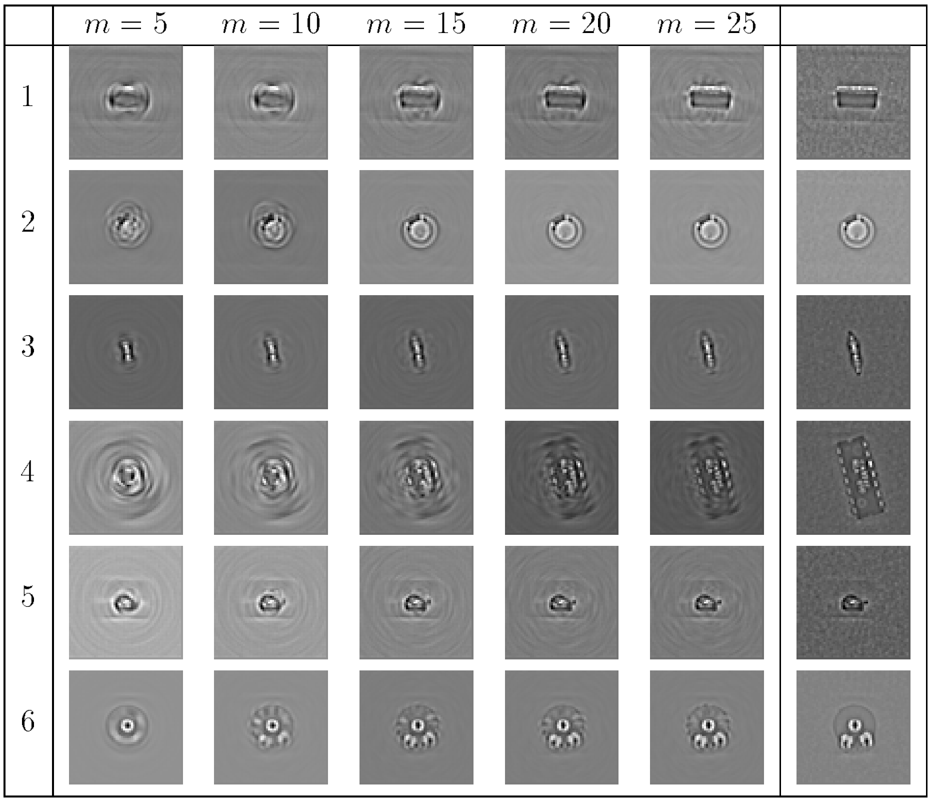


Tabelle 8.2: Rekonstruktionen von Objektansichten bei verschiedenen Dimensionen m des Eigenraumes.

Die Rücktransformation erfolgt gemäß

$$\boldsymbol{x}_r = \boldsymbol{\mu} + \sum_{i=1}^{m} x_i \cdot \boldsymbol{q}_i \tag{8.24}$$

mit $\boldsymbol{x} = (x_1, \dots, x_m) \in \mathcal{E}$. Je größer m ist, desto kleiner ist der Fehler $\|\boldsymbol{x} - \boldsymbol{x}_r\|$ zwischen $\boldsymbol{x}$ und der Rücktransformation $\boldsymbol{x}_r$.

Tabelle 8.2 zeigt die Bilder der Rücktransformationen bei verschiedenen Werten von m. Die letzte Spalte der Tabelle enthält die Bilder der Objekte in der ersten Ebene der Laplacepyramide. Für $m = 20$ ergibt sich eine hohe Ähnlichkeit der Rückprojektionen mit den Bildern in der letzten Spalte, bei $m = 25$ tritt keine wesentliche Verbesserung ein. Beim Transistor und beim Widerstand würde bereits $m = 15$, beim Transistor $m = 5$ für eine sehr genaue Codierung der Objekte genügen. Entscheidendes Kriterium für die Wahl der Dimension m muß jedoch nicht die möglichst genaue Repräsentation der vorverarbeiteten Ansichten im Eigenraum sein, die Erkennung kann bereits in Eigenräumen kleinerer Dimensionalität erfolgen. Entscheidend ist das Ergebnis des Objekterkenners insgesamt. Daher wird in den Tests $m \in \{5, 10, 15, 20\}$ variiert.

8.3 Neuronale Objekterkennung

Die Ansichten eines Objektes bilden (eindimensionale) Kurven im Eigenraum. Je dichter ein Punkt im Eigenraum, der die Transformation eines vorverarbeiteten Bildausschnitts ist, an einer Kurve liegt, desto größer ist die Ähnlichkeit zu diesem Objekt. Eine Objekterkennung basiert also insbesondere auf der Berechnung des metrischen Abstands zwischen einem Punkt und einer Kurve im Eigenraum.

Es soll kurz das von Murase und Nayar verwendete Verfahren zusammengefaßt werden. Murase und Nayar modellieren die objektspezifische Kurve durch Interpolation mit Splines [20]. Da sich die Objekte in den Ansichten sowohl in der Trainings- als auch in der Anwendungsphase auf schwarzem, homogenem Hintergrund befinden, ist die objektunabhängige Normierung gemäß (8.12) geeignet. Der Ansatz ist daher, *einen* Eigenraum aus den Ansichten von *allen* Objekten zu berechnen. In diesem globalen Eigenraum liegen dann die Kurven aller Objekte. Eine zu klassifizierende Objektansicht ist ein Punkt im Eigenraum. Die Objektklasse, der dieser Punkt zugeordnet wird, ist die am nächsten liegende Kurve, die durch die am nächsten liegende Objektansicht auf der Kurve bestimmt wird. Das Verfahren zur Suche des nahesten Punktes im Eigenraum wurde in [25] und [24] veröffentlicht, wobei letztere Quelle eine detaillierte Beschreibung des Algorithmus ist.

Die Modellierung durch Splines setzt a priori-Wissen über die Daten im Eigenraum voraus. Es muß bekannt sein, wie groß die intrinsische Dimensionalität der Kurve ist, auf der die Daten im Eigenraum liegen, also in diesem Fall eine eindimensionale Kurve. Insbesondere muß für die Interpolation durch Splines bekannt sein, ob die Kurve geschlossen oder offen ist. Inwieweit von der Modellierung durch Splines in den Objekterkennungsverfahren Gebrauch gemacht wird, wird in den Arbeiten von Murase und Nayar nicht genauer beschrieben. Die Bestimmung der am nahesten liegenden Kurve erfolgt durch das Punktsuchverfahren, das von der Modellierung durch Splines keinerlei Gebrauch macht. Insbesondere müssen alle Objektansichten gespeichert sein. Das Verfahren hat also im Gegensatz zum RBF-basierten Ansatz nicht die Fähigkeit zu generalisieren. Da nicht approximiert, sondern interpoliert wird, findet auch keine Glättung statt. Das den Daten aufgelagerte Rauschen fließt unmittelbar in die Objektmodellierung und in das Suchverfahren ein.

Die Objekterkennung erfolgt in dieser Arbeit durch RBF-Netze. Obwohl wir bereits wissen, daß die in dieser Arbeit verwendeten Trainingsdaten eindimensionale Kurven beschreiben, muß ihre intrinsische Dimensionalität nicht a priori bekannt sein, ebenso die spezielle Gestalt dieser Kurven, wie die Offenheit oder Geschlossenheit.

Die Verwendung von RBF-Netzen ergibt sich aus den nun folgenden Überlegungen. Eine Gaußsche Glockenkurve

$$h(\boldsymbol{x}, \boldsymbol{c}, \sigma) = e^{-\frac{(\boldsymbol{x}-\boldsymbol{c})^T(\boldsymbol{x}-\boldsymbol{c})}{2\sigma^2}} \tag{8.25}$$

deren Zentrum durch $\boldsymbol{c}$ und deren Breite durch σ festgelegt ist, hat ihren maximalen Funktionswert 1 im Zentrum. Je weiter $\boldsymbol{x}$ von $\boldsymbol{c}$ entfernt ist, desto kleiner ist der gegen 0

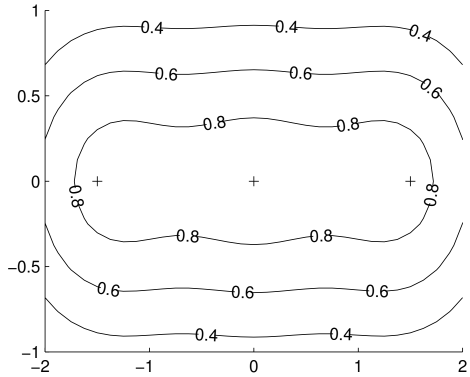


Abbildung 8.12: Konturplot der Linearkombination dreier Gaußscher Glockenkurven im zweidimensionalen Fall: $f : \mathbb{R}^2 \rightarrow \mathbb{R}; \boldsymbol{x} \mapsto \sum_1^3 w_i \cdot h(\boldsymbol{x}, \boldsymbol{c}_i, \sigma_i)$ mit $w_1 = w_3 = 0.8$, $w_2 = 0.75$, $\boldsymbol{c}_1 = (-1.5, 0)$, $\boldsymbol{c}_2 = (0, 0)$, $\boldsymbol{c}_3 = (1.5, 0)$, $\sigma_1 = \sigma_2 = \sigma_3 = \sqrt{0.5}$.

konvergierende Funktionswert. Durch eine geeignete Linearkombination von Glockenkurven unterschiedlicher Zentren läßt sich erreichen, daß der Rand der zusammenhängenden Menge von Punkten, deren Funktionswerte oberhalb einer Schwelle von etwa 0.5 liegen[2], einen annähernd gleichen Abstand zu einem durch die Zentren interpolierten oder approximierten Kurvenstück hat. Abbildung 8.12 zeigt ein Beispiel für den zweidimensionalen Fall. Durch Anwendung eines Schwellwertoperators mit einer geeignet zu wählenden Schwelle läßt sich somit ein Punkt klassifizieren.

Da die Normierung der Daten vom jeweiligen Objekt abhängt, wird auch für jedes Objekt ein Netz trainiert. Das nun folgende Kapitel beschreibt kurz RBF-Netze, sowie die verwendeten Verfahren zur Schätzung der Parameter.

8.3.1 RBF-Netze

RBF-Netze [28, 10] bestehen aus einer Eingabeschicht, einer verdeckten Schicht von Knoten mit radialen Basisfunktionen und einer Ausgabeschicht von Knoten mit linearen Aktivierungsfunktionen (siehe Abb. 8.13). Dieser Netztyp erlaubt eine nichtiterative Berechnung der Gewichte (*one shot learning*).
Im folgenden sei m die Dimension der Eingabevektoren, l die Anzahl der Basisfunktionen und s die Dimension des Ausgabevektors. Die i-te Basisfunktion sei mit h_i bezeichnet.

Der Ausgabevektor $\boldsymbol{y} = (y_1, \ldots, y_s)^T \in \mathbb{R}^s$ ergibt sich beim einfachsten Typ von RBF-

[2]Der genaue Wert dieser Schwelle wird nach dem Training eines RBF-Netzes festgelegt. Er ergibt sich aus den Funktionswerten der zu klassifizierenden Ansichten (folgt in Kap. 8.5).

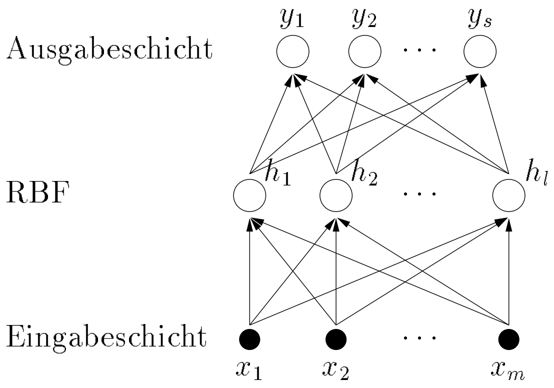


Abbildung 8.13: Struktur eines RBF-Netzes

Netzen aus der gewichteten Summe der Aktivitäten in der verdeckten Schicht:

$$y_j = \sum_{i=1}^{l} w_{i\,j} h_i(\boldsymbol{x}, \boldsymbol{c}_i) \quad , j = 1, \ldots , s \tag{8.26}$$

Als radiale Basisfunktion wird häufig die Gaußfunktion

$$h_i(\boldsymbol{x}, \boldsymbol{c}_i, \sigma_i) = e^{-\frac{(\boldsymbol{x}-\boldsymbol{c}_i)^T(\boldsymbol{x}-\boldsymbol{c}_i)}{2\sigma_i^2}} \tag{8.27}$$

verwendet. Zwecks Einsparung von Rechenoperationen wird jedoch in dieser Arbeit die äquivalente Funktion

$$h_i(\boldsymbol{x}, \boldsymbol{c}_i, \tau_i) = e^{\frac{1}{\tau_i}(\boldsymbol{x}-\boldsymbol{c}_i)^T(\boldsymbol{x}-\boldsymbol{c}_i)} \tag{8.28}$$

mit $\tau_i \in \mathbb{R}_{<0}$ verwendet.

Die Bestimmung der Parameter erfolgt durch Trainingsmuster, d. h. Paare von Eingabe- und Ausgabedaten, daher spricht man auch von einem *überwachten* Lernverfahren. Es sei P die Anzahl der Muster, $X = \{\boldsymbol{x}_i\,|, \boldsymbol{x}_i \in \mathbb{R}^m, i = 1, \ldots , P\}$ die Menge von Eingabedaten und $Y = \{\boldsymbol{y}_i\,|, \boldsymbol{y}_i \in \mathbb{R}^s, i = 1, \ldots , P\}$ die Menge der entsprechenden Ausgabedaten der Muster. Ziel des Trainings des RBF-Netzes ist die Minimierung des Fehlers zwischen den Ausgabedaten und der Ausgabe des Netzes:

$$\sum_{k=1}^{P} \|\boldsymbol{y}_k - RBF(\boldsymbol{x}_k)\|^2 \to \min \tag{8.29}$$

mit $RBF(\boldsymbol{x}_k) = (\sum_{i=1}^{l} w_{i\,1} h_i(\boldsymbol{x}_k, \boldsymbol{c}_i), \ldots , \sum_{i=1}^{l} w_{i\,s} h_i(\boldsymbol{x}_k, \boldsymbol{c}_i))^T$.

Ist die Anzahl l der Basisfunktionen gleich der Anzahl P der Trainingsmuster, so handelt es sich um eine Interpolation, im Falle $l < P$ um eine Approximation. Eine Approximation bewirkt die erwünschte Generalisierungseigenschaft und verhindert eine Überanpassung an die Daten und das den Daten aufgeprägte Rauschen (*overfitting*).

8.3.1.1 Zentren der Basisfunktionen

Für die Wahl der Zentren gibt es mehrere Möglichkeiten. Die einfachste ist die Verwendung von Eingabedaten aus der Menge der Trainingsmuster. Die andere ist das Berechnen der Zentren durch Clusterverfahren. In dieser Arbeit wird der K-Means-Algorithmus verwendet (siehe Algorithmus 4). Das Ergebnis des K-Means-Algorithmus ist eine Menge $C = \{\boldsymbol{c}_1, \dots \boldsymbol{c}_l\}$ von Clusterzentren und Mengen A_j von Indizes der Datenvektoren im j-ten Cluster. Die Clusterzentren sind die Mittelwerte der den Clustern zugeordneten Datenvektoren. Die einem Clustern zugeordneten Datenvektoren haben die Eigenschaft, daß sie zum Zentrum einen minimalen Abstand haben.

Der K-Means-Algorithmus liefert nur suboptimale Clusterzentren. Entscheidend ist die Wahl der $n \leq P$ initialen Zentren $\{\boldsymbol{c}_1, \dots \boldsymbol{c}_n\}$. Decken diese Zentren die Datenmenge schlecht ab, so kann es sogar passieren, daß sich weniger als n Zentren ergeben.

Wähle Menge $C := \{\boldsymbol{c}_1, \dots \boldsymbol{c}_n\}$, $n \leq P$ von Cluster-Zentren
$C_2 = \emptyset$
while $C \neq C_2$ **do** // solange sich die Cluster-Zentren ändern
  $C_2 = C$
  for all $j \in \{1, \dots, n\}$ **do**
    $A_j = \emptyset$ // Initialisiere Indexmengen
  end for
  for all $\boldsymbol{x}_i \in X$ **do**
    $j = \arg\min_{c_j \in C} \|\boldsymbol{x}_i - \boldsymbol{c}_j\|$ // suche Zentrum mit kleinstem Abstand zu $\boldsymbol{x}_i$
    $A_j = A_j \cup i$ // der i-te Datenvektor gehört zum j-ten Zentrum
  end for
  $C = \emptyset$
  for all $j \in \{1, \dots, n\}$ **do**
    if $A_j \neq \emptyset$ **then**
      $\boldsymbol{c} = \frac{1}{|A_j|} \sum_{i \in A_j} \boldsymbol{x}_i$ // neues Zentrum
      $C = C \cup \boldsymbol{c}$
    end if
  end for
end while // $C = \{\boldsymbol{c}_1, \dots \boldsymbol{c}_l\}$ mit $l \leq n$ ist Menge von Clusterzentren, $\{A_1, \dots, A_l\}$ die Indexmengen der in den Clustern liegenden Datenvektoren

Algorithmus 4: K-Means

8.3.1.2 Breite der Basisfunktionen

Die Festlegung der Breite σ_i bzw. τ_i einer Basisfunktion h_i erfolgt häufig durch die Varianz σ_i^2 des i-ten Clusters. Die aus den Daten der Objektansichten resultierenden Varianzen sind jedoch sehr klein. Die Basisfunktionen sind somit zu schmal, folglich konvergieren die

Funktionswerte sehr schnell gegen 0 und es gibt somit keine genügend großen Überlappungen zwischen den Basisfunktionen. Prinzipiell lassen sich die Breiten auch durch Vielfache der Varianzen festlegen oder durch Gradientenabstieg trainieren.

In dieser Arbeit wird eine andere Strategie verfolgt. Ist $M_i = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}$ die Menge der Vektoren im i-ten Cluster, so soll die Breite von h_i so klein sein, daß die Funktionswerte dieser i-ten Basisfunktion gerade oberhalb einer bestimmten Schwelle $b \in]0,1]$ liegen:

$$\tau_i = \max\{\tau \in \mathbb{R}_{<0} \,|\, \forall \boldsymbol{x} \in M_i : h(\boldsymbol{x}, \boldsymbol{c}_i, \tau) \geq b\} \tag{8.30}$$

Da die Funktion ln monoton wachsend ist, $\ln(b) < 0$ und $\tau < 0$, gelten die folgenden Äquivalenzen:

$$b \leq e^{\frac{1}{\tau}(\boldsymbol{x}-\boldsymbol{c}_i)^T(\boldsymbol{x}-\boldsymbol{c}_i)} \quad \Leftrightarrow \quad \ln(b) \leq \frac{1}{\tau}(\boldsymbol{x}-\boldsymbol{c}_i)^T(\boldsymbol{x}-\boldsymbol{c}_i) \tag{8.31}$$

$$\Leftrightarrow \quad \tau \leq \frac{(\boldsymbol{x}-\boldsymbol{c}_i)^T(\boldsymbol{x}-\boldsymbol{c}_i)}{\ln(b)} \tag{8.32}$$

Da $\ln(b) < 0$ folgt dann für τ_i aus (8.30):

$$\tau_i = \frac{(\boldsymbol{x}_{\max}-\boldsymbol{c}_i)^T(\boldsymbol{x}_{\max}-\boldsymbol{c}_i)}{\ln(b)} \tag{8.33}$$

mit

$$\boldsymbol{x}_{\max} = \arg\max_{\boldsymbol{x}\in M_i}(\boldsymbol{x}-\boldsymbol{c}_i)^T(\boldsymbol{x}-\boldsymbol{c}_i) \tag{8.34}$$

Auf diese Weise wird zu vorgegebener Schwelle $b \in]0,1]$ für jede Basisfunktion h_i die Breite $\tau_i \in \mathbb{R}_{<0}$ berechnet. Die a priori festzulegende Schwelle b wird in den Trainingsserien geeignet variiert.

8.3.1.3 Gewichte

Die Berechnung der optimalen Gewichte zwischen der versteckten Schicht und der Ausgabeschicht erfolgt durch lineare Ausgleichsrechnung. Wie in den vorhergehenden Abschnitten sei P die Anzahl der Trainingsmuster, $l \leq P$ die Anzahl der RBF-Knoten, m die Dimension der Ausgabeschicht und $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_P, \boldsymbol{y}_P)\}$ die Menge der Trainingsmuster.

Die Gewichte zwischen der versteckten Schicht und der Ausgabeschicht lassen sich in der $l \times s$-Matrix $\boldsymbol{W}$ zusammenfassen, wobei W_{ij} das Gewicht vom i-ten Knoten der RBF-Schicht zum j-ten der Ausgabeschicht bezeichnet. Mit der durch

$$H_{ij} = h_j(\boldsymbol{x}_i, \boldsymbol{c}_j) \tag{8.35}$$

definierten $P \times l$-Matrix $\boldsymbol{H}$ und der $P \times m$-Matrix

$$\boldsymbol{Y} = \begin{pmatrix} {\boldsymbol{y}_1}^T \\ \vdots \\ {\boldsymbol{y}_P}^T \end{pmatrix} \tag{8.36}$$

läßt sich das Optimierungsproblem schreiben als

$$\boldsymbol{H}\boldsymbol{W} = \boldsymbol{Y} \tag{8.37}$$

Die optimale Lösung ist durch

$$\boldsymbol{W} = \boldsymbol{H}^{+} \cdot \boldsymbol{Y} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{Y} \tag{8.38}$$

gegeben.

8.4 Wahl der Trainingsmuster

Aufgrund der objektspezifischen Normierung ergeben sich objektabhängige RBF-Netze, d. h. die RBF-Netze werden derart trainiert, daß sie ein bestimmtes Objekt von den übrigen unterscheiden. Daher muß auch die Menge $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_P, \boldsymbol{y}_P)\}$ von Trainingsmustern objektabhängig gewählt werden. Die Ausgabeschicht ist eindimensional ($s = 1$).

Für das Training der RBF-Netze gibt es zwei Möglichkeiten:

1. Training ohne Gegenbeispiele: Die Eingabedaten $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P\}$ sind die vorverarbeiteten Ansichten eines Objektes. Alle Ausgaben $\boldsymbol{y}_i \in \mathbb{R}$ werden auf einen konstanten Wert ungleich 0 gesetzt:

$$\boldsymbol{y}_i = 1 \qquad \text{für alle} \quad i = 1, \ldots, P \tag{8.39}$$

2. Training mit Gegenbeispielen: Die Menge der Eingabedaten der Trainingsmuster ist die Vereinigung der Menge X_{positiv} der vorverarbeiteten Ansichten des Objektes, und der Menge X_{negativ} der vorverarbeiteten Ansichten der übrigen Ansichten, welche die Gegenbeispiele sind. Die Ausgaben werden gemäß

$$\boldsymbol{y}_i = \begin{cases} 1 & \text{, falls } \boldsymbol{x}_i \in X_{\text{positiv}} \\ 0 & \text{, falls } \boldsymbol{x}_i \in X_{\text{negativ}} \end{cases} \qquad \text{für alle} \quad i = 1, \ldots, P \tag{8.40}$$

gesetzt.

8.5 Bestimmung der Schwelle

Die Entscheidung, ob eine Ansicht zu einem Objekt gehört, soll durch Anwendung eines Schwellwertoperators auf die eindimensionale Ausgabe des Netzes erfolgen, d. h. eine Ansicht gehört genau dann zum Objekt, wenn $\boldsymbol{y} \geq \theta_{\text{RBF}}$. Voraussetzung für die Anwendbarkeit dieses Kriteriums in der Anwendungsphase ist die Trennbarkeit der Beispiele und

Gegenbeispiele, welche die Menge der Trainingsdaten bilden. Es bezeichne $RBF(\boldsymbol{x})$ die Ausgabe des RBF-Netzes bei Eingabe des Vektors $\boldsymbol{x}$. Ist

$$a = \max\{\mathrm{RBF}(\boldsymbol{x}_i) \,|\, \boldsymbol{x}_i \in X_{\text{negativ}}\} \quad \text{und} \quad b = \min\{\mathrm{RBF}(\boldsymbol{x}_i) \,|\, \boldsymbol{x}_i \in X_{\text{positiv}}\} \tag{8.41}$$

so ist θ mit

$$a < \theta_{\mathrm{RBF}} \leq b \tag{8.42}$$

zu wählen, z. B. $\theta_{\mathrm{RBF}} = \frac{a+b}{2}$, falls $a < b$. Andernfalls ist die Menge der Trainingsdaten nicht trennbar und das trainierte Netz unbrauchbar.

8.6 Neuronaler Operator

Ziel ist ein lokaler, objektspezifischer Operator $\mathcal{O}$, der sich aus den Komponenten Normierung, Transformation in den Eigenraum und RBF-Netz zusammensetzt:

$$h = \mathcal{O}g \tag{8.43}$$

wobei g das Bild einer bestimmten Ebene der Gauß- oder Laplacepyramide ist. Der Wertebereich des Ergebnisbildes ist eine Teilmenge von $\mathbb{R}$. Der Einzugsbereich des Operators ist $M \times M$, also gleich der Dimension der dem Training zugrundeliegenden Ansichten. Der Output $h_{i,j}$ des Operator $\mathcal{O}$ an der Stelle (i, j) ergibt sich auf folgende Weise aus der $M \times M$-Umgebung des Pixels $g_{i,j}$:

1. „Ausschneiden" der $M \times M$-Umgebung:

   $$c_{k,l} = g_{i-\frac{M}{2}+k,j-\frac{M}{2}+l} \tag{8.44}$$

   für $k, l \in \{0, M - 1\}$

2. Normierung des Bildes c gemäß Gleichung (8.15) zu f mit dem objektspezifischen Aperturoperator $\mathcal{A}_O$.

3. Konvertierung von f zu $\boldsymbol{x}$ gemäß Gleichung (8.16) und Transformation von $\boldsymbol{x}$ in den objektspezifischen Eigenraum $\mathcal{E}_O$.

4. Abbildung des Ergebnisses durch das RBF-Netz ergibt $h_{i,j}$.

Dieser lokale Operator ist offensichtlich verschiebungsinvariant, jedoch nichtlinear. Aufgrund der symmetrischen Umgebung der Größe $M \times M$ um den Aufpunkt ergibt sich ein symmetrischer Bildrand. Die Berechnung der Antworten $h_{i,j}$ erfolgt unabhängig voneinander und kann somit parallel erfolgen.

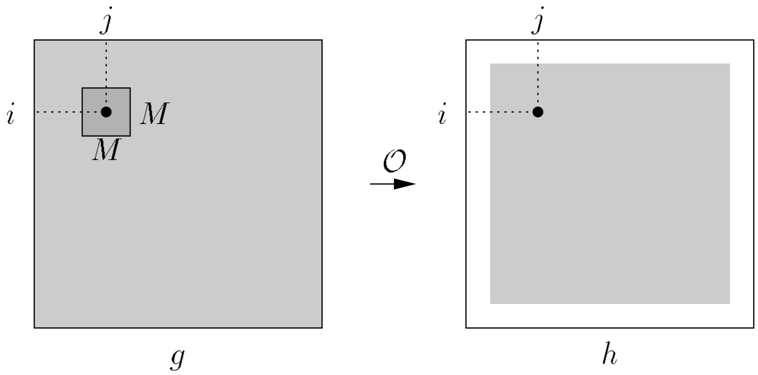


Abbildung 8.14: Darstellung des lokalen Operators

Kapitel 9

Testergebnisse des zweiten Teils

Dieses Kapitel beschreibt die Testergebnisse für die erscheinungsbasierte, neuronale Objekterkennung.

Die Anzahl der Ebenen der Gaußpyramide und der Laplacepyramide sowie die Größe der Ansichten wird durch die Geometrie der Kamera und ihrem Abstand zum Objekt bestimmt. Die Trainingsansichten sind Teilbilder der ersten Ebene der Pyramiden, die Abtastrate wird also halbiert. Die Ansichten haben eine Größe von $M \times M = 59 \times 59$ Pixel. Wie bereits erwähnt, wurden pro Objekt 360 Ansichten mit einer Winkeldifferenz von $1°$ aufgenommen (siehe Kap. 8.1). Alle Ansichten eines Objektes gehen in die Hauptkomponentenanalyse ein. Wie im vorhergehenden Kapitel sei mit m die Dimension des objektspezifischen Eigenraumes bezeichnet und diese ist gleich der Dimension der Eingabeschicht des RBF-Netzes. Die weiteren Parameter sind die Anzahl l der Clusterzentren bzw. Zentren der RBF-Netze sowie die Schranke $b \in]0,1]$, welche die Breiten der Basisfunktionen gemäß Gleichung (8.33) festlegen. Die ein RBF-Netz beschreibenden Parameter m, l, b und die Trainingsmengen werden in den nun folgenden Versuchen variiert, um für die Objekterkennung geeignete Netze zu erhalten. Dazu wird zunächst das Verhalten der Netze und der neuronalen Operatoren auf den Trainingsdaten untersucht, danach wird getestet, welche Antworten sich bei Bildern von Platinen ergeben. Dabei werden die Ergebnisse bei Vorverarbeitung durch die Gaußpyramide und bei Vorverarbeitung durch die Laplacepyramide miteinander verglichen.

9.1 Training der RBF-Netze

Für die folgenden Trainingsserien wurde zunächst für beide Pyramiden für jedes der 6 Objekte eine Hauptkomponentenanalyse seiner 360 Ansichten durchgeführt.

Serie 1 Mit den RBF-Netzen dieser Serie wird getestet, inwieweit die Modellierung der Objektkurve durch ein RBF-Netz ausreicht, um zwischen Ansichten des Objektes und der Gegenobjekte unterscheiden zu können. Jeweils für beide Pyramiden werden je Objekt RBF-Netze mit den Parametern

$$(m, b, l) \in \{5, 10, 15, 20\} \times \{0.5, 0.75, 0.9\} \times \{30, 60, 90\} \tag{9.1}$$

trainiert, also 72 Netze pro Objekt. Die l Zentren sind das Ergebnis der Clusterung der 360 Datenvektoren des Objektes. Die Trainingsdaten sind die vorverarbeiteten Ansichten des Objektes, also nicht die Daten der Gegenobjekte. Die Ausgaben der Trainingsdaten werden somit gemäß Gleichung (8.39) gesetzt. Bezüglich der trainierten Objektdaten sowie der Daten der Gegenbeispiele ergibt sich bei beiden Pyramiden ein geringer Fehler. Unterschiede bestehen jedoch in der Fähigkeit der Netze, Objektdaten und Gegenobjektdaten voneinander zu trennen. Die Tabellen B.1 und B.2 in Anhang B geben jeweils den Schwellwert $\theta_{\mathrm{RBF}} = \frac{a+b}{2}$ mit a und b nach Gl. (8.41) an, falls die Ausgabedaten linear trennbar sind. Es zeigt sich, daß im Falle der Gaußpyramide mehr Netze trennen können als bei Vorverarbeitung durch die Laplacepyramide.

Serie 2 Während in der vorhergehenden Trainingsserie für die Bestimmung der Gewichte der RBF-Netze nur die Objektdaten verwendet wurden, werden in dieser Serie auch die Daten der Gegenklassen einbezogen. Die Zentren sind die gleichen wie in Serie 1. Die Ausgabedaten der Trainingsmuster werden wie in (8.40) gewählt.
Die Tabellen B.3 und B.4 geben die Schwellen bzw. Trennungsleistungen der Netze für die verschiedenen Parameter an. Ein Vergleich der Tabellen B.1 und B.3 sowie B.2 und B.4 zeigt, daß sich gegenüber Serie 1 die Klassifikationsleistung der RBF-Netze bei Vorverarbeitung durch die Gaußpyramide überraschenderweise verschlechtert, im Falle der Laplacepyramide ergibt sich eine deutliche Verbesserung.

Je kleiner die Anzahl der Zentren pro Objekt ist, desto mehr tritt die Objektkurve im Eigenraum des Objektes in den Hintergrund. Je geringer die Anzahl der Zentren ist, desto größer sind auch die durch die Schranke b festgelegten Breiten der Basisfunktionen. Im Hinblick einer besseren Generalisierungsfähigkeit der gelernten Operatoren in der Anwendungsphase werden nun Netze mit deutlich weniger Knoten in der versteckten Schicht trainiert und auf der Menge der Ansichten getestet.

Serie 3 In dieser Serie wurden die Ansichten der Objekte in $l \in \{5, 10, 15\}$ Cluster gruppiert, ansonsten sind die Voraussetzungen die gleichen wie in Serie 1. Die trainierten Netze sind jedoch überwiegend nicht in Lage, die Ansichten der Objekte und der Gegenobjekte voneinander zu trennen.

Serie 4 Das Einbeziehen von Gegenbeispielen in die Berechnung der optimalen Gewichte bringt gegenüber der vorhergehenden Serie keine nennenswerten Verbesserungen.

Serie 5 In dieser Serie werden die Gegenbeispiele voll in das Training einbezogen, nicht nur in die Berechnung der optimalen Gewichte, sondern auch in die Clusterung. Bezüglich des RBF-Netzes eines Objektes ergeben sich die Eingabedaten des K-Means-Algorithmus durch Transformation der Objektdaten und Daten der Gegenobjekte in den Eigenraum des Objektes. Die Anzahl der initialen Zentren ist derart gewählt, daß sich durch die Clusterung pro Objekt $l_O \in \{5, 10, 15\}$ Clusterzentren ergeben und somit $l \in \{30, 60, 90\}$ RBF-Knoten in der versteckten Schicht der Netze.
Die Tabellen B.5 und B.6 zeigen die Ergebnisse. Im Falle der Gaußpyramide existiert kein gemeinsamer Satz von Parametern $(m, b, l_O) \in \{5, 10, 15, 20\} \times \{0.5, 0.75, 0.9\} \times \{5, 10, 15\}$,

für die sich Netze ergeben, mit denen jeweils alle Objekte korrekt klassifiziert werden können. Die schlechte Klassifizierbarkeit des fünften Objektes, des Transistors, durch die „transistorspezifischen“ Netze kann auf das kleine Bild dieses Objektes in den Ansichten zurückgeführt werden. Warum die Ergebnisse für das erste Objekt, den Spannungsregler, so schlecht sind, läßt sich jedoch nicht erklären.
Im Falle der Laplacepyramide (Tab. B.6) ergeben sich in Hinblick auf die weitere Anwendung viele brauchbare Netze. Die Netze können die Trainingsdaten nicht trennen, wenn die Dimension des Eigenraumes oder die Anzahl der Zentren zu klein sind.

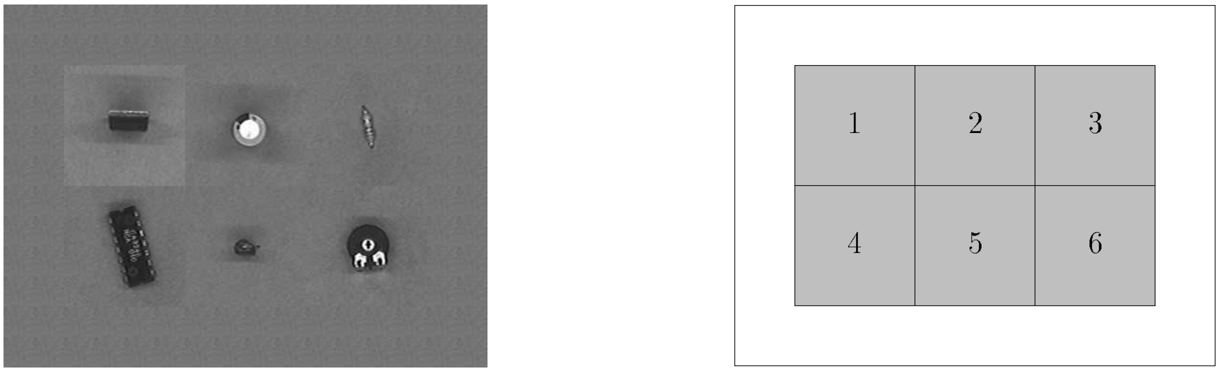
Aus den Trainingsserien ergeben sich verschiedene RBF-Netze, deren Verhalten im folgenden weiter untersucht werden soll. Überraschenderweise zeigen die RBF-Netze bei reiner Skalierung der Trainingsansichten (Gaußpyramide) im Gegensatz zur Bandpaßfilterung (Laplacepyramide) ein besseres Klassifizierungsverhalten auf den Trainingsdaten, wenn die Daten der Gegenklassen nicht in das Training einbezogen werden.

9.2 Antwort der Operatoren in der Umgebung des Mittelpunktes der Trainingsansichten

Im letzten Abschnitt wurden Netze auf unterschiedliche Art und Weise trainiert sowie das Verhalten bezüglich der Trainingsdaten untersucht. Die Antworten auf die Trainingsdaten sind nach Konstruktion des Operators in Kapitel 8.6 die Antworten in den Mittelpunkten der Trainingsansichten. In Hinblick auf die Anwendung des Operators auf beliebige Bilder ist jedoch auch das Verhalten des Operators außerhalb des Mittelpunktes der Ansichten entscheidend. Es wird erwartet, daß die Antwort eines Operators um so größer wird, je näher sein Aufpunkt an den trainierten Mittelpunkt einer Objektansicht kommt. Auf den Ansichten der Gegenobjekte soll die Antwort in etwa Null sein.

In diesem Abschnitt wird getestet, wie sich der Operator in der Umgebung des trainierten Mittelpunktes der Ansicht sowie auf den Ansichten der Gegenobjekte verhält. Als Testbild dient dabei das in Abbildung 9.1(a) dargestellte Bild, das aus Ansichten der 6 Objekte zusammengesetzt ist. Um im folgenden die Erläuterungen zu den Abbildungen zu vereinfachen, zeigt Abbildung 9.1(b) die Objektzugehörigkeit und die Lage der Ansichten im Testbild. Es werden zunächst die Ergebnisse bei Vorverarbeitung durch die Gaußpyramide vorgestellt, danach die Ergebnisse im Falle der Laplacepyramide.

Tabelle 9.1 zeigt die Antwort der objektspezifischen Operatoren, deren RBF-Netze mit $(m, l, b) = (10, 30, 50)$ wie in Serie 1 angegeben trainiert wurden, auf das durch die Gaußpyramide vorverarbeitete Testbild.
Bei allen Operatoren ergeben sich auf der Ansicht des Objekt, für welches das Netz trainiert wurde, Peaks, deren Maximum im Mittelpunkt der Ansicht liegt. Die Nebenpeaks, die einige Antworten aufweisen, entstehen durch die Ähnlichkeit der Bildausschnitte in den Aufpunkten mit den Ansichten des Objektes anderer Orientierungen. Besonders deutlich



(a) Testbild (b) Zugehörigkeit der Objektansichten

Abbildung 9.1: Testbild aus Ansichten der Objekte

wird dies beim symmetrischen IC. Da der Mittelpunkt des ICs bei der Bildaufnahme nicht exakt auf der Rotationsachse des Drehtellers lag, ergibt sich aufgrund der Symmetrie dieses Bildobjektes eine hohe Ähnlichkeit mit der um 180° rotierten, „verschobenen“ Ansicht. Somit ergeben sich zwei fast gleichhohe Peaks.
Bezüglich der Ansichten der Gegenobjekte weisen der Spannungsregler-Operator und der Widerstand-Operator relativ hohe Antworten bei den Ansichten einiger Gegenobjekte auf. Beim Spannungsregler-Operator ergeben sich die höheren Antworten offensichtlich aus der teilweisen Ähnlichkeit des Spannungsregler-Grauwertprofils mit den Profilen des IC-Randes und des Potentiometers. Beim Widerstand ist die hohe Antwort in den Ansichten des Transistors auf die geringe Größe bei der Bildobjekte und ihrer so zustandekommenden Ähnlichkeit zurückzuführen. Der Widerstand-Operator ist somit für eine zuverlässige Objekterkennung ungeeignet. Ebenso problematisch ist der Einsatz des Transistor-Operators. Er liefert zwar einen scharfen Peak, was einerseits eine genau Lokalisation, andererseits jedoch eine schlechte Generalisierung bedeutet. Der Grund liegt ebenfalls in der geringen Größe des Transistors im Bild. Eine positive Eigenschaft der Operatoren ist ihre geringe Rauschempfindlichkeit. Auf dem Hintergrund, der ein leichtes Rauschen aufweist, ergeben sich Antworten, die fast Null sind.
Bei Operatoren, bei denen die versteckte Schicht der RBF-Netze mehr RBF-Knoten besitzt, ergeben sich schärfere Peaks.

Die Antwort der Operatoren bei Vorverarbeitung durch die Laplacepyramide wird am Beispiel der Netze aus Trainingsserie 5 mit $(m, l_O, b) = (15, 5, 0.5)$ diskutiert. Die versteckte Schicht der RBF-Netze besteht aus 30 Knoten. Tabelle 9.2 zeigt die Antworten der Operatoren der Objekte auf das Testbild. Auffällig ist die deutlich höhere Sensitivität der Operatoren gegenüber Rauschen. Insbesondere beim Transistor-Operator sind die Antworten außerhalb des Mittelpunktes der Transistor-Ansicht fast genauso hoch wie der schwache Peak. Dieser Operator ist somit aufgrund der geringen Größe des zu erkennenden Objektes ungeeignet. Trotz der geringeren Anzahl von Knoten (5 statt 30) ergeben sich im Vergleich

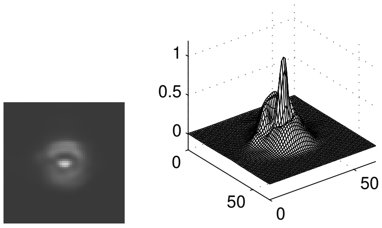
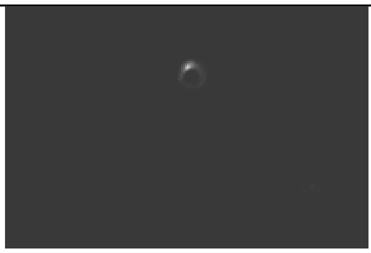
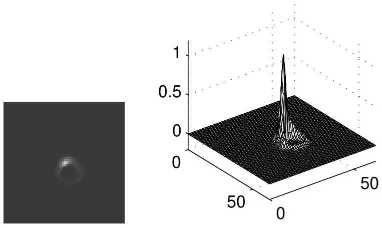
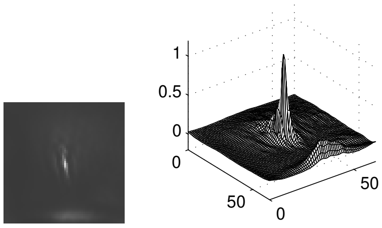
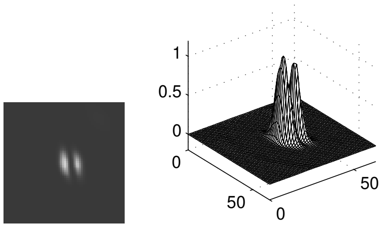
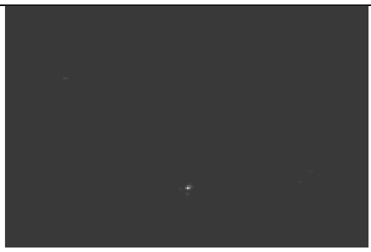
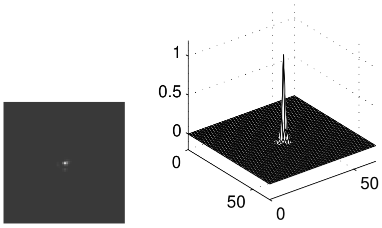
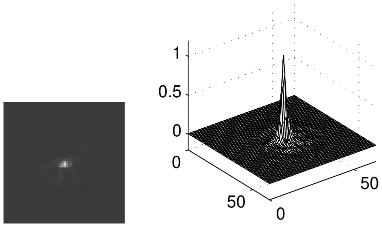
1 (Spannungsregler)		
2 (Kondensator)		
3 (Widerstand)		
4 (IC)		
5 (Transistor)		
6 (Potentiometer)		

Tabelle 9.1: Antworten der Operatoren auf das Testbild (Gaußpyramide, $(m, l, b) = (10, 30, 0.5)$)

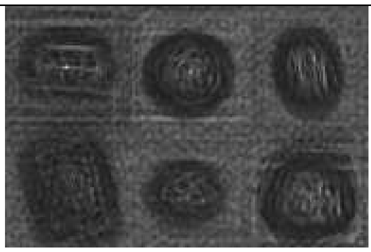
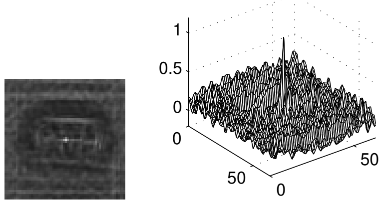
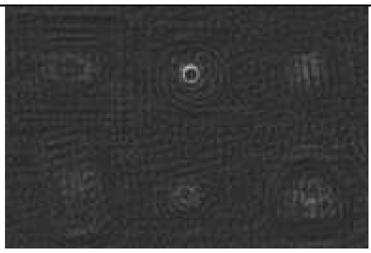
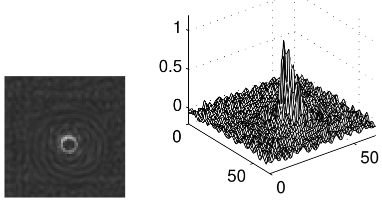
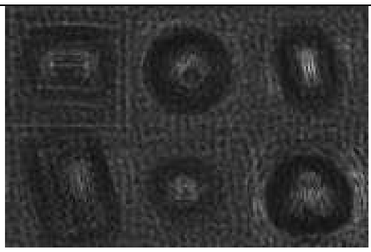
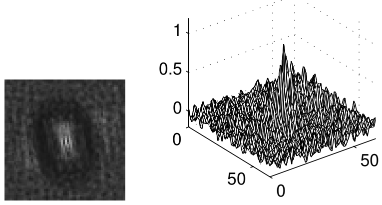
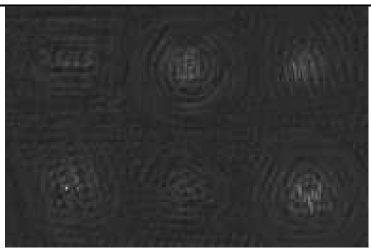
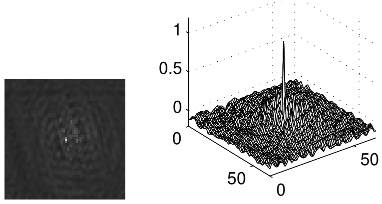
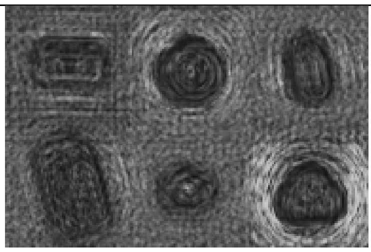
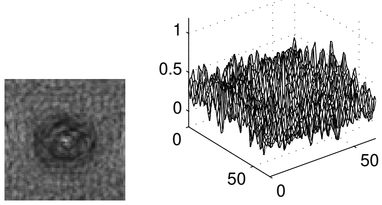
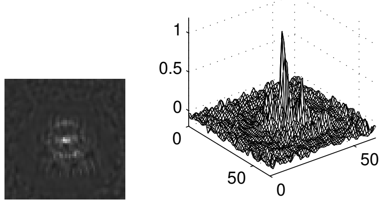
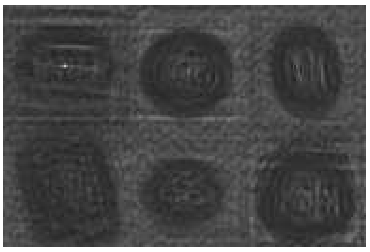
<table>
<tr><td>1
(Spannungsregler)</td><td></td><td></td></tr>
<tr><td>2
(Kondensator)</td><td></td><td></td></tr>
<tr><td>3
(Widerstand)</td><td></td><td></td></tr>
<tr><td>4
(IC)</td><td></td><td></td></tr>
<tr><td>5
(Transistor)</td><td></td><td></td></tr>
<tr><td>6
(Potentiometer)</td><td></td><td></td></tr>
</table>

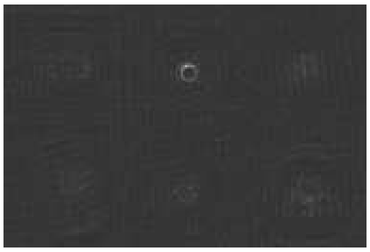
Tabelle 9.2: Antworten eines Operators auf das Testbild (Laplacepyramide, $(m, l_O, b) = (15, 5, 0.5)$)

zur Gaußpyramide schärfere Peaks und somit eine genauere Lokalisation des erkannten Objektes. Deutlich erkennbar ist auch die Sensitivität gegenüber Kanten, die zu den Kanten des zu erkennenden Objektes gleichförmig sind. Der Operator des Spannungsreglers zeigt hohe Antworten bei geraden Kantensegmenten, der Operator des Kondensators bei kreisförmigen Kanten.

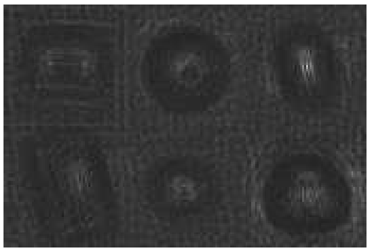
Die Abbildungen 9.2 und 9.3 zeigen die Antworten der Operatoren bei höherer Schwelle b bzw. einer geringeren Dimension der Eigenräume. Eine höhere Schwelle b führt zu breiteren Basisfunktionen. Im Vergleich zu Tabelle 9.2 ergeben sich deutlich glattere Antworten und breitere Peaks. Bei einer geringeren Dimension der Eigenräume (Abb. 9.3) ist ein ähnliches Verhalten der Operatoren erkennbar, jedoch ist die Klassifikationsleistung der Netze gesunken, da die Operatoren auch auf den Ansichten der Gegenbeispiele im Vergleich zu Tabelle 9.2 höhere Antworten liefern.



(a) Spannungsregler-Operator

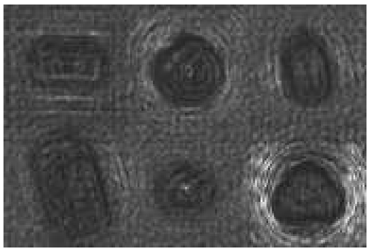


(b) Kondensator-Operator



(c) Widerstand-Operator

(d) IC-Operator



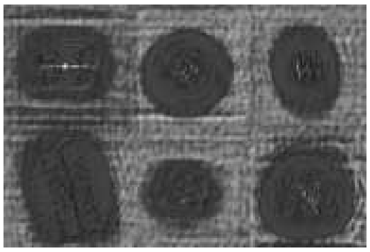
(e) Transistor-Operator

(f) Potentiometer-Operator

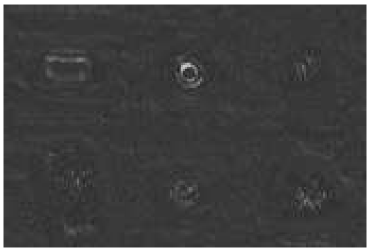
Abbildung 9.2: Antworten der Operatoren bei höherer Schwelle (Laplacepyramide, $(m, l_O, b) = (15, 5, 0.9)$)

9.3 Verhalten der Operatoren auf untrainierten Szenen

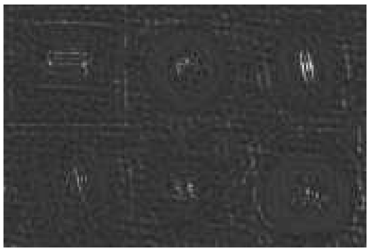
In diesem Abschnitt wird das Verhalten der trainierten Operatoren auf Bildern realer Szenen getestet. Im Gegensatz zum letzten Abschnitt sind die Trainingsansichten nicht in den Bildern enthalten. Abbildung 9.4 zeigt die Bilder zweier Testszenen. Bei der Bildaufnahme



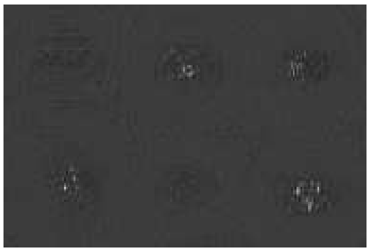
(a) Spannungsregler-Operator



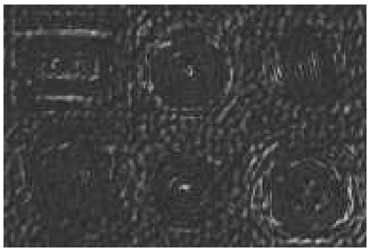
(b) Kondensator-Operator



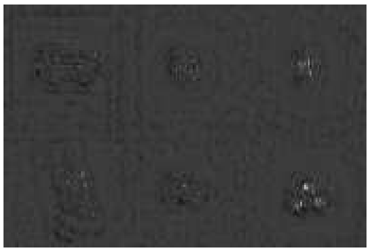
(c) Widerstand-Operator



(d) IC-Operator



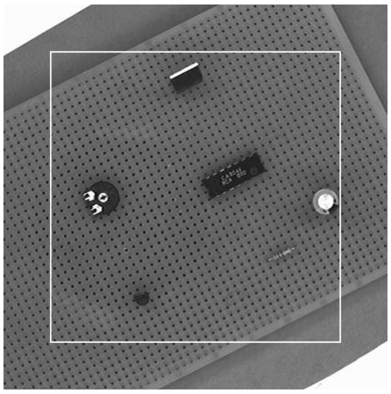
(e) Transistor-Operator



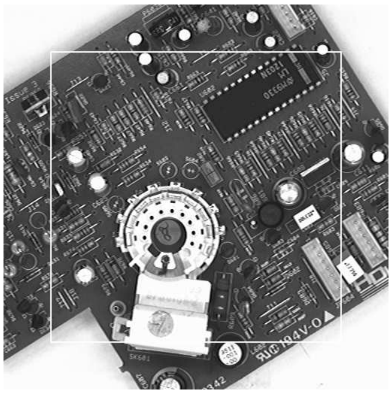
(f) Potentiometer-Operator

Abbildung 9.3: Antworten der Operatoren bei kleiner Dimension des Eigenraumes (Laplacepyramide, $(m, l_O, b) = (5, 5, 0.75)$)

war der Abstand der Kamera zur Platine annähernd der gleiche wie bei der Aufnahme der Trainingsansichten. Die Platine in Abbildung 9.4(a) ist mit den Objekten bestückt, die auch zur Aufnahme der Trainingsansichten verwendet wurden. Die Szene zeichnet sich durch den inhomogenen Hintergrund aus, der eine andere Helligkeit besitzt als der Hintergrund der Trainingsansichten der Objekte. Auf der zweiten Platine in Abbildung 9.4(b) befinden sich ein Spannungsregler, Kondensatoren, Widerstände und Transistoren, die un-



(a) Platine mit Trainingsobjekten



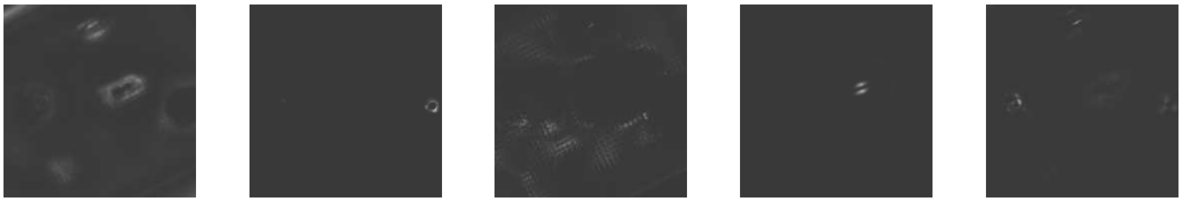
(b) Platine mit ähnlichen Objekten

Abbildung 9.4: Bilder zweier Testszenen mit Platinen gleich der Trainingsobjekte sind und in Form und Größe eine unterschiedliche Ähnlichkeit zu den Trainingsobjekten besitzen. Erschwerend für die Objekterkennung ist auch hier der inhomogene Hintergrund um die zu erkennenden Objekte herum.

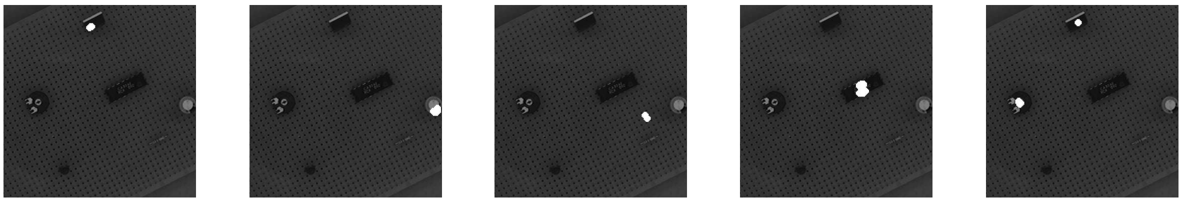
Die nun folgenden Ergebnisbilder beziehen sich auf die in Abb. 9.4 umrandeten Bereiche. Es werden sowohl die Bilder der Antworten der Operatoren dargestellt, als auch die Teilbilder der Szenen mit den markierten Bereichen, an denen die Antworten oberhalb der Schwellen liegt. Das durch die Anwendung des Schwellwertoperators entstandene Binärbild wurde zwecks deutlicherer Darstellung mit dem Strukturelement in Abb. 8.7 dilatiert, bevor es mit dem Bild der Szene überlagert wurde. Die Schwellen zu den Operatoren sind in den entsprechenden Tabellen in Anhang B abzulesen.

Abbildung 9.5 zeigt die Ergebnisse für die Operatoren aus Serie 1 mit $(m, l, b) = (10, 30, 0.5)$ und Anwendung der Gaußpyramide. Der Spannungsregler, der Kondensator und der IC werden richtig erkannt. Der Widerstand-Operator liefert sowohl in Bereichen der Gegenobjekte als auch auf dem Platinenhintergrund hohe Antworten (Abb. 9.5(c)), wodurch es zur Fehldetektion kommt (Abb. 9.5(h)). Das Potentiometer (Abb. 9.5(e) und 9.5(j)) wird erkannt, allerdings liegt auch die niedrigere Antwort im Bereich des Spannungsreglers oberhalb der Schwelle. Der Operator des Transistors liefert einen zu schwachen Wert (< 0.1) im Bereich dieses Bildobjektes, auf dem Hintergrund ist die Antwort in etwa Null.

Abbildung 9.6 zeigt die Ergebnisse für die gleichen Operatoren nach Anwendung auf die zweite Testszene. Die Antwort des Spannungsregler-Operators ist zwar im Bildbereich des Spannungsreglers relativ hoch, jedoch unterhalb der Schwelle. In den übrigen Bereichen auf der Platine, in denen kein Spannungsregler existiert, ergeben sich Antworten oberhalb

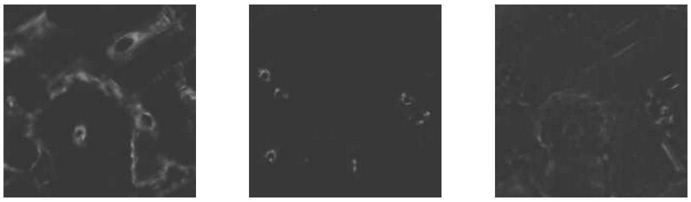


(a) Spannungsregler (b) Kondensator (c) Widerstand (d) IC (e) Potentiometer

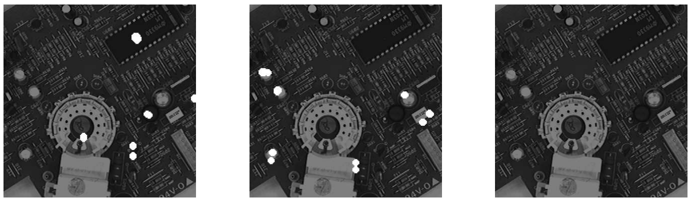


(f) Spannungsregler (g) Kondensator (h) Widerstand (i) IC (j) Potentiometer

Abbildung 9.5: Antworten der Operatoren aus Serie 1 mit $(m, l, b) = (10, 30, 0.5)$ bei Vorverarbeitung durch Gaußpyramide (Abb. (a)-(e)), darunter die markierten Bildpositionen, an denen die Antworten oberhalb der jeweiligen Schwellwerte liegen (Abb. (f)-(j)).



(a) Spannungsregler (b) Kondensator (c) Potentiometer



(d) Spannungsregler (e) Kondensator (f) Potentiometer

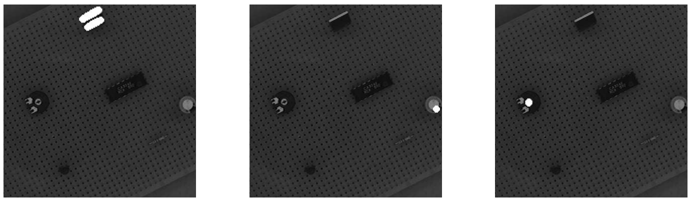
Abbildung 9.6: Antworten der Operatoren aus Serie 1 mit $(m, l, b) = (10, 30, 0.5)$, Gaußpyramide, darunter die markierten Positionen mit oberhalb der Schwelle liegenden Antworten.

der Schwelle. Kondensatoren, deren Größe mit denen der gelernten Kondensatoren in etwa übereinstimmen, werden erkannt (Abb. 9.6(e)). Es befindet sich kein Potentiometer auf der Platine. Die Antworten des Potentiometers sind deutlich niedrig (Abb. 9.6(c)) und liegen unterhalb der Schwelle (siehe Abb. 9.6(f)).

Es folgen nun die Ergebnisse für die Operatoren mit Vorverarbeitung durch die Laplacepyramide. Die Netze wurden wie in Serie 2 angegeben trainiert. Die Parameter der RBF-Netze sind $(m, l, b) = (10, 90, 0.9)$. Die Schwellen zur Trennung zwischen Objekt und Gegenobjekt sind Tabelle B.4 entnommen. Die Ergebnisse für das erste Testbild sind in Abbildung 9.7 dargestellt. Im Vergleich zu Abbildung 9.5 sind die Antworten in den Bereichen der Gegenobjekte kleiner. Die Bandpaßfilterung führt also zu einer besseren Trennbarkeit zwischen den Objektklassen. Der Potentiometer-Operator liefert ein eindeutiges Ergebnis (vergl. Abb. 9.7(f) mit Abb. 9.5(j)). Jedoch kann der IC aufgrund der innerhalb des Durchlaßbereiches des Aperturoperators liegenden, hohen Antworten des Bandpaßfilters auf dem inhomogenen Hintergrund nicht erkannt werden. Die in dieser Arbeit verwendete Normierung führt also bei Anwendung der Laplacepyramide zu schlechteren Ergebnissen, wenn die Größe und die Form des Aperturfensters von denen der Bildobjekte zu stark abweichen. Deutlich erkennbar ist auch wieder der höhere Rauschpegel in den Antworten der Operatoren. Diese ist beim Widerstand-Operator und beim Operator des Transistors besonders hoch (nicht durch Abbildung dargestellt).

Abbildung 9.8 zeigt die Ergebnisse für das zweite Testbild im Falle der Laplacepyramide für die Operatoren mit $(m, l, b) = (10, 30, 0.9)$. Mit den Schwellen aus Tabelle B.4 ist nur der Spannungsregler erkennbar. Da die Antworten der Operatoren für die zu erkennenden Objekte in den realen Szenen im allgemeinen niedriger sind, lassen sich durch Verringerung der ermittelten Schwellen häufig die richtigen Objekte erkennen. Abbildung 9.8 zeigt das

(a) Spannungsregler (b) Kondensator (c) Potentiometer



(d) Spannungsregler (e) Kondensator (f) Potentiometer

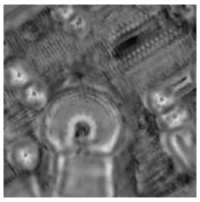
Abbildung 9.7: Antworten der Operatoren aus Serie 2 mit $(m, l, b) = (10, 90, 0.9)$ bei Vorverarbeitung durch Laplacepyramide (Abb. (a)-(c)), darunter die markierten Bildpositionen, an denen die Antworten oberhalb der Schwellwerte liegen (Abb. (d)-(f)).

Ergebnis der Klassifikation bei einer Schwelle von 0.09. In Abbildung 9.8(c) wurde – im Gegensatz zu den übrigen Abbildungen dieses Kapitels – die Grauwerte normiert, so daß den hellen Bildpunkten relativ kleinen Antworten entsprechen (≈ 0.1). Es zeigt sich, daß die Bildregionen außerhalb des gesuchten Objektes kleinere Antworten liefern. Somit läßt sich sich auch ein Maximumoperator zur Lokalisation einsetzen. Die Voraussetzung ist das a priori-Wissen um die Existenz des gesuchten Objektes im Bild.

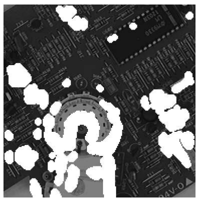
Die nach Serie 5 trainierten RBF-Netze, also die Netze, bei denen auch RBF-Knoten von Zentren der Gegenobjekte existieren, liefern bei beiden Pyramiden schlechtere Ergebnisse. Die Abbildungen 9.9 und 9.10 zeigen die Ergebnisse der Operatoren mit $(m, l_O, b) = (10, 5, 0.5)$ (Gaußpyramide, Schwelle 0.58) bzw. $(m, l_O, b) = (15, 5, 0.75)$ (Laplacepyramide). Im Gegensatz zu den Netzen ohne RBF-Zentren der Gegenklassen gibt es erheblich mehr Fehldetektionen.

(a) Spannungsregler (b) Spannungsregler (c) Kondensator (d) Kondensator

Abbildung 9.8: Ergebnis der Operatoren aus Trainingsserie 2 mit $(m, l, b) = (10, 90, 0.9)$, Laplacepyramide

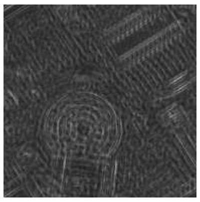


(a) Kondensator

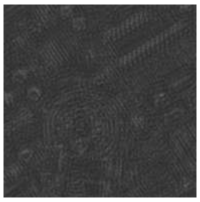


(b) Kondensator

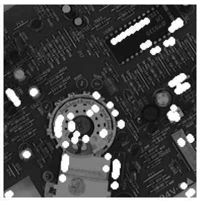
Abbildung 9.9: Ergebnisse der Operatoren aus Serie 5 mit $(m, l_O, b) = (10, 30, 0.5)$, Gaußpyramide



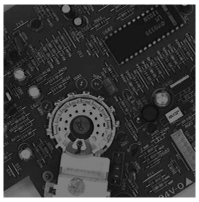
(a) Spannungsregler



(b) Kondensator



(c) Spannungsregler



(d) Kondensator

Abbildung 9.10: Ergebnisse der Operatoren aus Serie 5 mit $(m, l_O, b) = (15, 30, 0.75)$, Laplacepyramide

Abbildung 9.11 zeigt ein weiteres Testbild. Die Ergebnisse der auf den bandpaßgefilterten Bilddaten basierenden Operatoren ist in Abbildung 9.12 dargestellt. Der Operator für den IC liefert zu kleine Antworten, es wird somit kein IC erkannt. Der Spannungsregler-Operator zeigt wieder hohe Antworten an geraden Kanten oder rechtwinkligen Ecken. An

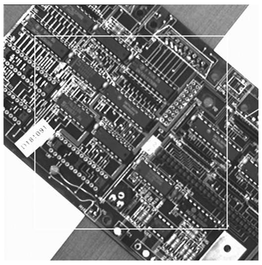
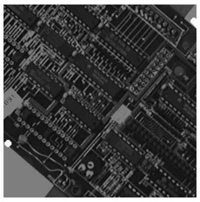


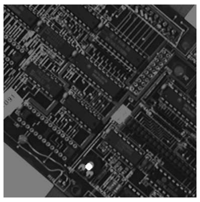
Abbildung 9.11: Testbild

(a) Spannungsregler

(b) Potentiometer

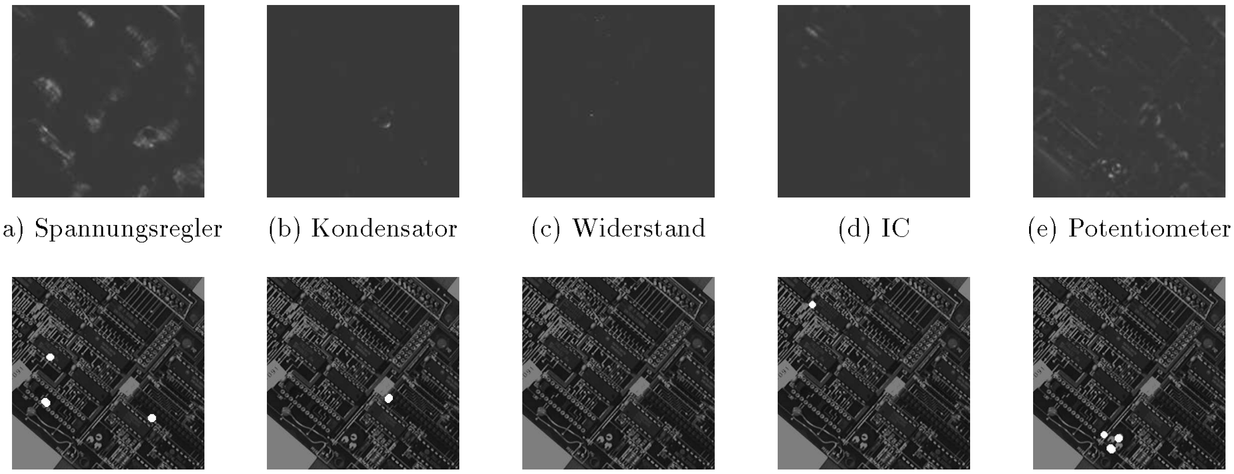


(c) Spannungsregler



(d) Potentiometer

Abbildung 9.12: Ergebnisse zu Testbild in Abb. 9.11 für die Operatoren aus Serie 2 mit $(m, l, b) = (10, 90, 0.9)$, Laplacepyramide



(a) Spannungsregler (b) Kondensator (c) Widerstand (d) IC (e) Potentiometer

(f) Spannungsregler (g) Kondensator (h) Widerstand (i) IC (j) Potentiometer

Abbildung 9.13: Ergebnisse zu Testbild in Abb. 9.11 für die Operatoren aus Serie 1 mit $(m, l, b) = (10, 30, 0.5)$, Gaußpyramide

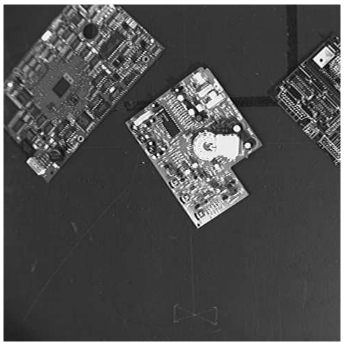
der Position, an der sich das Potentiometer befindet, ergeben sich Werte oberhalb von 0.3. Abbildung 9.13 zeigt die Ergebnisse der grauwertbasierten Operatoren. Die Erkennung des ICs in Abb. 9.13(i) ist auf die relativ helle Umgebung um das Objekt herum zurückzuführen.

Das nun folgende, abschließende Beispiel stammt aus einem Gesamtdurchlauf. Es unterscheidet sich von den vorhergehenden Beispielen durch eine größere Abweichung von dem Abstand und der Orientierung, mit denen die Trainingsansichten aufgenommen wurden. Die Abbildungen 9.14(a)-9.14(c) zeigen die Szene aus den 3 Sichten des Kamerasystems. In die erste Ansicht ist die Rückprojektion des erkannten Objektes eingezeichnet. Nach der Bewegung des Kamerasystems relativ zum Zielobjekt blickt die Kamera, mit der die Erkennung der Elektronikbauteile erfolgen soll, auf den Mittelpunkt der Platine (siehe Abb. 9.14(c)).

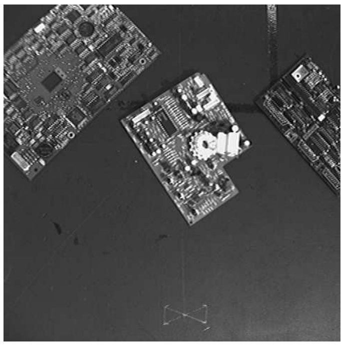
Im Falle der Laplacepyramide zeigt nur der Spannungsregler hohe Antworten, obwohl sich dieses Objekt nicht im relevanten Bildausschnitt befindet. Die Antworten des Kondensator-Operators ist zu klein. Der Grund liegt in der geringeren Toleranz dieser Operatoren, die bereits in den sehr scharfen Peaks (Tabelle 9.2) zum Ausdruck kam.
Bei den grauwertbasierten Operatoren ergeben sich bessere Resultate. Diese sind in Abbildung 9.16 dargestellt. Die Antworten des Kondensator-Operators sind im Bereich der Kondensatoren relativ gering, so daß die Anwendung des ermittelten Schwellwertes zur Klassifizierung nur ein erkanntes Objekt ergibt (Abb. 9.16(g)). Die Verwendung einer kleineren Schwelle zeigt jedoch, daß sich im Bereich der Kondensatoren deutlich höhere Werte als im übrigen Bild ergeben (siehe Abb. 9.16(k)).

Welche der beiden Vorverarbeitungen, d. h. Gaußpyramide oder Laplacepyramide, für die

(a) 1. Ansicht (linke Kamera)

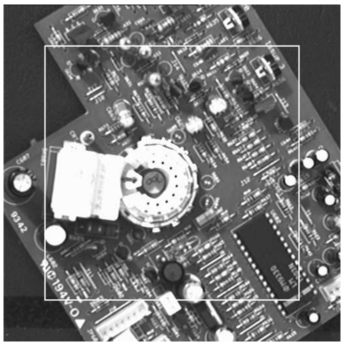


(b) 2. Ansicht (rechte Kamera)



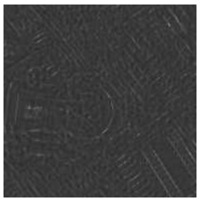
(c) 3. Ansicht

(d) 1. Ansicht (linke Kamera) nach der Bewegung

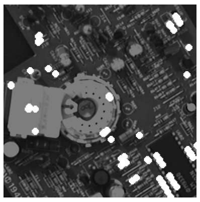


(e) Bild nach Bewegung des Kamerasystems (mittlere Kamera)

Abbildung 9.14: Ansichten der Kameras vor und nach der Manipulation des Kamerasystems

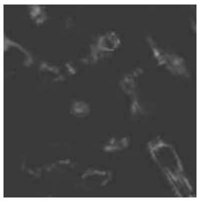


(a) Spannungsregler



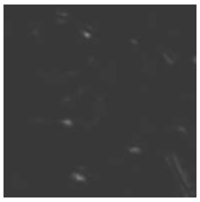
(b) Spannungsregler

Abbildung 9.15: Antwort des Spannungsregler-Operators aus Serie 2 mit $(m, l, b) = (10, 90, 0.9)$, Laplacepyramide

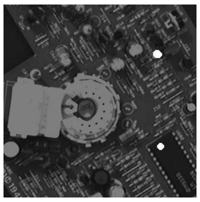

(a) Spannungsregler

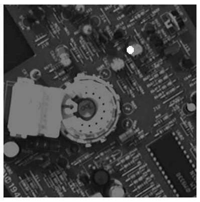

(b) Kondensator

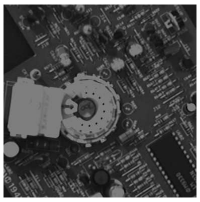

(c) Widerstand

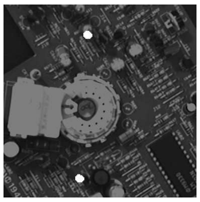

(d) IC

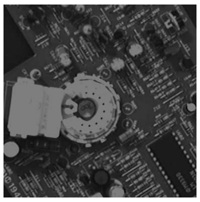

(e) Transistor

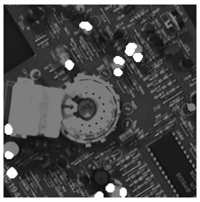

(f) Spannungsregler

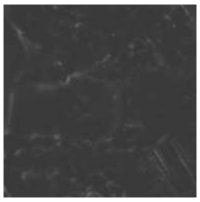

(g) Kondensator


(h) Widerstand


(i) IC


(j) Transistor


(k) Kondensator


(l) Potentiometer

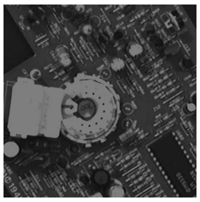

(m) Potentiometer

Abbildung 9.16: Ergebnisse der Operatoren aus Serie 1 mit $(m, l, b) = (10, 30, 0.5)$, Gaußpyramide

Erkennung dieser gelernten Objekte im allgemeinen geeigneter ist, läßt sich nicht genau festlegen. Für das grauwertbasierte Verfahren spricht die geringere Dimension der Eigenräume und die geringere Anzahl an Eingabeneuronen und versteckten RBF-Neuronen. Andererseits treten aufgrund der größeren Toleranz dieser Operatoren mehr Fehldetektionen auf, d. h. Bildregionen, die eigentlich dem Hintergrund oder Gegenobjekten zuzuordnen sind, werden fälschlicherweise als Objekt erkannt. Bei den Operatoren, denen die bandpaßgefilterten Daten (Laplacepyramide) zugrunde liegen, ist die geringere Anzahl von Fehldetektionen von Vorteil. Nachteilig ist jedoch erstens die stärkere Übertragung des Rauschens und zweitens, daß in Bildern, in denen die Ansichten der zu erkennenden Objekte zu den trainierten Ansichten eine geringere Ähnlichkeit aufweisen, aufgrund der niedrigeren Antworten weniger Objekte detektiert werden. Diese Eigenschaft ist insbesondere bei ungünstigen Beleuchtungsverhältnissen oder größeren Abweichungen von der Größe und Orientierung der Objekte, die in den Ansichten implizit enthalten ist, von Bedeutung und somit auch für die praktische Anwendung in diesem Gesamtsystem. Durch die unterschiedliche Lage des Roboterarmes sind signifikante Beleuchtungsänderungen kaum auszuschließen. Dann jedoch liefern die toleranteren grauwertbasierten Operatoren mehr Detektionen der gewünschten Objekte im Bild.

Kapitel 10

Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Hand-Auge-System zur Erkennung von Objekten entwickelt und implementiert. Der Einsatz eines mobilen Kamerasystems ergab sich aus der Notwendigkeit, den Suchraum einzuschränken und die Blickposition und -orientierung in geeigneter Weise auf die letztendlich zu erkennenden Objekte anzupassen. Aus den durch die Komplexität der Szene, den Umweltbedingungen und der Zielstellung gegebenen Voraussetzungen dieser Aufgabe ergeben sich zwei unterschiedliche Ausprägungen der Objekterkennung. Zur Lösung der Erkennungsaufgaben wurden zwei verschiedene Paradigmen der Objekterkennung angewendet. Diese sind die geometrische, modellbasierte Vorgehensweise unter Anwendung des Prinzips der perzeptuellen Organisation und die erscheinungsbasierte, neuronale Form der Objekterkennung.

Dem geometrischen, modellbasierten Erkennungsverfahren zur Lokalisation und Identifikation der polygonalen, planaren Flächen (wie es bei Platinen der Fall ist) wurden Geraden als Bildprimitiva zugrunde gelegt. Die Detektion dieser Geraden erfolgte durch die orientierungsselektive Hough-Transformation [13, 11, 14], da diese effizient und robust gegen Rauschen sowie teilweisen Verdeckungen ist. Dabei wurde der Einsatz weiterer Bildoperatoren und Strategien notwendig, um die Detektion in Bildern realer Szenen zu verbessern. Zur besseren Lokalisation von Geraden an flachen Grauwertkanten wurde ein Verfahren zur Unterdrückung von Nicht-Maxima eingesetzt [15], außerdem wurden Strategien diskutiert, mit denen sich die durch die Diskretisierung des Parameterraumes ergebenden Probleme reduzieren lassen.
Die Ermittlung der Eckpunkte der Objektoberflächen erfolgte insbesondere über die rekonstruierten Geraden. Die Rekonstruktion der Geraden setzt die Berechnung der Korrespondenzen von Bildgeraden in mehreren Ansichten voraus. Weil a priori kein Wissen über die Lage der 3D-Geraden existiert, ist die Bestimmung der Korrespondenzen in drei Ansichten notwendig. Daher wurde in dieser Arbeit ein Verfahren zum punktweisen Matching von Bildgeraden in Stereoansichten entwickelt, welches nicht nur bereits existierende geometrische und photometrische Ansätze (epipolare Bedingung und Template-Matching), sondern auch ein Votierungsverfahren einbezieht. Letzteres ist erforderlich, um gegen Rauschen und aus den durch die verschiedenen Blickwinkel der Stereokameras resultierenden Überdeckungen robust zu sein. Dieses Verfahren wurde auf drei Ansichten erweitert. Die Rekonstruktion der Geraden erfolgte durch das Verfahren von Faugeras [9]. Es wurde ein

Verfahren entwickelt, welches über mehrere Abstraktionsstufen hinweg in der Szene vorhandene Objektoberflächen lokalisiert, rekonstruiert und identifiziert. Dabei wurden Invarianten sowie Repräsentationen und Vergleichsverfahren einbezogen, die positions-, rotations- und skalierungsinvariant sind. Hier ist die Repräsentation der planaren Kontur als formbeschreibendes Merkmal von besonderer Bedeutung [2].

Die Eigenschaften der Objekte auf der Platine, insbesondere ihre allgemeine Form, sowie die Vorteile konnektionistischer Ansätze motivierten die erscheinungsbasierte, neuronale Objekterkennung. Dazu wurde das erscheinungsbasierte Verfahren von Murase, Nayar et al. modifiziert, um trotz des inhomogenem Hintergrunds der Objektansichten in der Anwendungsphase annähernd beleuchtungsunabhängig gelernte Objekte erkennen zu können. Anstelle einer globalen Normierung werden die Daten auf den in den Ansichten eines Objektes relevanten Bildbereich normiert. Dabei bleibt jedoch die Erkennung bei solchen Objekten unzureichend, bei denen die äußere Form von diesem Bildbereich stark abweicht. Die Erkennung erfolgte nicht über die Modellierung der Objektkurven im Eigenraum durch Splines, sondern neuronal durch RBF-Netze. Es wurde ein neuronaler, nichtlinearer Bildoperator entwickelt. Außerdem wurde das Verhalten der Operatoren bei zwei verschiedenen low level Vorverarbeitungen getestet, der Skalierung des Grauwertbildes durch eine Gaußpyramide und der (zusätzlichen) Bandpaßfilterung mittels der Laplacepyramide [6]. Dabei ergaben sich auf bandpaßgefilterten Bildern eine höhere Sensitivität gegenüber Rauschen sowie eine höhere Dimension der Eigenräume und der RBF-Netze. Welche der beiden Alternativen die besseren Operatoren liefert, konnte jedoch nicht festgelegt werden.

Die Algorithmen zum Stereomatching und zur Rekonstruktion von Punkten und Geraden setzen eine implizite Kalibrierung der Kameras voraus. Zur Bewegung des Kamerasystems durch den Roboterarm ist außerdem eine Hand-Auge-Kalibrierung notwendig. Beide Kalibrierungen wurden ebenfalls implementiert und getestet.

Für das vorliegende System bieten sich viele Test- und Erweiterungsmöglichkeiten an. Die geometrische Objekterkennung kann für Objekte erweitert werden, die nicht zur Klasse der polygonalen, planaren Objekte gehören. Sinnvoll ist auch der Austausch einzelner Komponenten, wie z. B. der orientierungsselektiven Kantendetektion durch möglicherweise robustere Operatoren. Hier bietet sich beispielsweise ein auf Quadraturfiltern basierender Kantendetektor an [14, Kap. 11.5.3].
Für die erscheinungsbasierte Objekterkennung sind Tests mit anderen Objekten sowie mehr als nur einem Freiheitsgrad bei der Aufnahme der Ansichten sinnvoll. In dieser Arbeit war der inhomogene Hintergrund erschwerend für die Objekterkennung, interessant ist auch die Frage, wie sich die Operatoren bei Verdeckungen der trainierten Objektansichten verhalten. Um den Rechenaufwand der erscheinungsbasierten Objekterkennung zu reduzieren, ist die Entwicklung geeigneter Attentionsmechanismen denkbar.

Neben den getesteten Umgebungen bietet sich eine Anwendung des Systems auf anderen Roboterarchitekturen an, z. B. einem mobilen Roboter. Die Rekonstruktion der Objektoberflächen erfolgte bezüglich des mobilen Kamerasystems und nicht etwa in Bezug auf ein äußeres, festes Koordinatensystem, wie beispielsweise das Basissystem des Roboters.

Dieses ermöglicht eine leichte Anpassung. Es muß lediglich die Hand-Auge-Schnittstelle modifiziert werden. Eine rein technische Verbesserung ist die Übertragung des Systems auf höher parallele Rechnersysteme als es momentan vorliegt, um Echtzeitfähigkeit anzustreben.
Da die Position und die Orientierung im Raum Teil des polygonalen Oberflächenmodells ist, ergibt sich bei Einsatz eines geeigneten Endeffektors die Möglichkeit des Greifens oder Manipulierens der erkannten Objekte.

Anhang A

Iterative Berechnung der Eigenwerte und Eigenvektoren

In den Kapiteln 5.7.1 und 8.2.3 ist die Berechnung von Eigenwerten und korrespondierenden Eigenvektoren von symmetrischen, positiv semidefiniten Matrizen notwendig[1]. Dabei ist eine Anzahl größter Eigenwerte (und entsprechenden Eigenvektoren) von Interesse, die relativ zur Dimension der Matrix sehr klein ist. Daher ist ein Verfahren zweckmäßig, das nicht alle Eigenwerte einer Matrix simultan berechnet, wie es z. B. bei der Singulärwertzerlegung der Fall ist, sondern nur eine bestimmte Anzahl größter oder kleinster Eigenwerte. Aus diesem Grund wurde das von Yang et al. [40] veröffentlichte Verfahren verwendet, welches kurz vorgestellt werden soll.

Sei $\boldsymbol{M} \in \mathbb{R}^n$ eine reelle, symmetrische, positiv semidefinite Matrix. Diese hat n Eigenwerte $\lambda_1, \cdots, \lambda_n \in \mathbb{R}_{\geq 0}$ mit $\lambda_1 \leq \cdots \leq \lambda_n$ und entsprechende Eigenvektoren $\boldsymbol{q}_1, \cdots, \boldsymbol{q}_n \in \mathbb{R}^m \setminus \{\boldsymbol{0}\}$. Der kleinste Eigenwert λ_1 von $\boldsymbol{M}$ ist das globale Minimum der als *Rayleigh-Quotient* bezeichneten Abbildung

$$R(\boldsymbol{x}) = \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \tag{A.1}$$

$$= \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\|\boldsymbol{x}\|^2} \tag{A.2}$$

Die Minimallösung von R ist der zu λ_1 gehörende Eigenwert $\boldsymbol{q}_1$. Das Iterationsverfahren zur Berechnung von Minimalwert und -lösung basiert auf konjugiertem Gradientenabstieg (siehe Algorithmus 5). Er setzt voraus, daß die Matrix $\boldsymbol{M}$ reel, symmetrisch und außerdem positiv definit oder negativ definit ist.

Die Berechnung des größten Eigenwertes geschieht durch Ersetzen von $\boldsymbol{M}$ durch $-\boldsymbol{M}$. Die Matrix $-\boldsymbol{M}$ ist negativ semidefinit[2], hat also n Eigenwerte $\lambda_1 \leq \cdots \leq \lambda_n \leq 0$. Das Verfahren liefert dann den kleinsten Eigenwert λ_1 von $-\boldsymbol{M}$. Somit ist $-\lambda_1$ der größte Eigenwert von $\boldsymbol{M}$.

[1] Eine Matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ ist nach Definition positiv semidefinit genau dann, wenn $\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} \geq 0$ für alle $\boldsymbol{x} \in \mathbb{R}^n \setminus \boldsymbol{0}$

[2] Eine Matrix $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ ist negativ semidefinit genau dann, wenn $\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} \leq 0$ für alle $\boldsymbol{x} \in \mathbb{R}^n \setminus \boldsymbol{0}$

Eine Menge $\lambda_1, \cdots, \lambda_m$, $m \leq n$ von Eigenwerten erhält man iterativ durch

$$\boldsymbol{M}_1 = \boldsymbol{M} \tag{A.3a}$$

$$\boldsymbol{M}_s = \boldsymbol{M}_{s-1} - \lambda_{s-1}\boldsymbol{q}_{s-1}\boldsymbol{q}_{s-1}{}^T \tag{A.3b}$$

initialisiere $\boldsymbol{x} \in \mathbb{R}^n$
$b := 0$
$\lambda := \frac{\boldsymbol{x}^T\cdot\boldsymbol{M}\cdot\boldsymbol{x}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
$\boldsymbol{p} := \frac{\lambda\cdot\boldsymbol{x}-\boldsymbol{M}\cdot\boldsymbol{x}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
$not\_ready :=$ `true`
$R_{\min} = \infty$
while not_ready **do**
    $P_a := \frac{\boldsymbol{p}^T\cdot\boldsymbol{M}\cdot\boldsymbol{x}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
    $P_b := \frac{\boldsymbol{p}^T\cdot\boldsymbol{M}\cdot\boldsymbol{p}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
    $P_c := \frac{\boldsymbol{p}^T\cdot\boldsymbol{x}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
    $P_d := \frac{\boldsymbol{p}^T\cdot\boldsymbol{p}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
    $D := P_b \cdot P_c - P_a \cdot P_d$
    if $|D| > 0$ **then**
        $B := P_b - \lambda \cdot P_d$
        $C := P_a - \lambda \cdot P_c$
        $t := \frac{-B+\sqrt{B^2-4\cdot C\cdot D}}{2\cdot D}$
        $\boldsymbol{x} := \boldsymbol{x} + t \cdot \boldsymbol{p}$
        $\boldsymbol{x} := \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$
    end if
    $R = \frac{\boldsymbol{x}^T\cdot\boldsymbol{M}\cdot\boldsymbol{x}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
    if $R < R_{\min} \wedge |R - R_{\min}| > \delta$ **then**
        $R_{\min} := R$
        $\lambda := \frac{\boldsymbol{x}^T\cdot\boldsymbol{M}\cdot\boldsymbol{x}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
        $\boldsymbol{r} := \frac{\lambda\cdot\boldsymbol{x}-\boldsymbol{M}\cdot\boldsymbol{x}}{\boldsymbol{x}^T\cdot\boldsymbol{x}}$
        $b := \frac{-(\boldsymbol{r}^T\cdot\boldsymbol{M}\cdot\boldsymbol{p}+r^2\cdot(\boldsymbol{x}^T\cdot\boldsymbol{p}))}{\boldsymbol{p}^T\boldsymbol{M}\boldsymbol{p}-\lambda\cdot\boldsymbol{p}^T\boldsymbol{p})}$
        $p := \boldsymbol{r} + b \cdot \boldsymbol{p}$
    else
        $not\_ready :=$ `false`
    end if
end while   // $R_{\min}$ ist kleinster Eigenwert, $\boldsymbol{q} := \boldsymbol{x}$ der Eigenvektor

Algorithmus 5: Iterative Berechnung des kleinsten Eigenwertes und des entsprechenden Eigenvektors durch konjugierten Gradientenabstieg

Anhang B

Tabellen

Dieser Anhang enthält die Tabellen zu den Trainingsserien in Kapitel 9.1. Die Tabellen enthalten im Falle der Trennbarkeit der Trainingsdaten die Schwellen $\frac{a+b}{2}$ mit a und b gemäß Gleichung 8.41. Sind die Daten nicht trennbar, so existiert keine Schwelle, was durch das Symbol - gekennzeichnet wird.

Bedeutung der Parameter:

$m \in \{5, 10, 15, 20\}$: Dimension der Eingabeschicht des RBF-Netzes.

$b \in \{0.5, 0.75, 0.9\}$: Untere Schranke, welche die Breite der Basisfunktionen bestimmt.

$l \in \{30, 60, 90\}$: Anzahl der RBF-Knoten in der versteckten Schicht, d. h. Anzahl der radialen Basisfunktionen.

$l_O \in \{5, 10, 15\}$: Anzahl der RBF-Knoten pro Objekt. Da sechs Objekte trainiert werden, ergibt sich somit eine sechsfache Anzahl von radialen Basisfunktionen im RBF-Netz.

$O \in \{1, 2, \ldots, 6\}$: Objektnummer. 1: Spannungsregler, 2: Kondensator, 3: Widerstand, 4: IC, 5: Transistor, 6: Potentiometer

m=5									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	–	0.3813	–	–	0.6728	–	–	0.8885
2	0.6495	0.3788	0.3667	0.8476	0.5592	0.4827	0.9585	0.7751	0.6017
3	0.5772	0.3793	0.3132	0.8008	0.5576	0.4222	–	0.7462	0.5314
4	0.4431	0.3988	0.3759	0.5120	0.4834	0.4829	0.8267	0.4968	0.4905
5	0.3968	0.2722	0.2806	0.5579	0.3676	0.3913	0.8997	0.4949	0.4771
6	0.5668	0.3934	0.3489	0.9543	0.5232	0.4785	–	0.7768	0.6406

m=10									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	0.6015	0.3108	0.3170	–	0.5514	0.4145	–	0.9064	0.6056
2	0.4476	0.3707	0.3681	0.6069	0.4732	0.4781	0.8769	0.5326	0.4989
3	0.4742	0.3878	0.3403	0.8275	0.4875	0.4541	–	0.6631	0.5012
4	0.4463	0.4026	0.3802	0.5530	0.4828	0.4845	–	0.5049	0.4940
5	0.4238	0.2906	0.2930	0.5649	0.3778	0.3899	0.7410	0.5081	0.4622
6	0.5203	0.3924	0.3573	0.9230	0.4947	0.4624	–	0.7179	0.5142

m=15									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	0.6569	0.2954	0.3150	–	0.4957	0.4102	–	–	0.6087
2	0.4367	0.3742	0.3707	0.6130	0.4777	0.4821	0.8886	0.5324	0.4984
3	0.6043	0.3971	0.3227	–	0.5241	0.4453	–	0.9234	0.5628
4	0.4417	0.3930	0.3830	0.6523	0.4790	0.4837	–	0.5300	0.4983
5	0.4468	0.3135	0.2951	0.6401	0.4067	0.3903	0.7802	0.5557	0.4682
6	0.7238	0.3829	0.3628	–	0.5356	0.4715	–	–	0.5846

m=20									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	0.7154	0.3029	0.3206	–	0.5104	0.4082	–	–	0.6361
2	0.4536	0.3818	0.3718	0.6306	0.4756	0.4820	0.9106	0.5403	0.4992
3	0.6770	0.3946	0.3470	–	0.5609	0.4636	–	–	0.5200
4	0.4545	0.3944	0.3834	0.7826	0.4789	0.4839	–	0.5879	0.4981
5	0.4798	0.3434	0.2967	0.6775	0.4346	0.3948	0.8550	0.6056	0.4764
6	–	0.4140	0.3621	–	0.8091	0.4884	–	–	0.9210

Tabelle B.1: Schwellen zu Serie 1, Gaußpyramide

m=5									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.6165	0.3247	–	0.8341	0.5371	–	–	0.8026
2	0.6721	0.3792	0.3176	–	0.5300	0.4320	–	0.9507	0.5687
3	0.7636	0.5073	0.3303	–	0.7247	0.5386	–	0.9650	0.7483
4	–	0.5593	0.3401	–	0.8002	0.5743	–	–	0.8428
5	0.3979	0.3452	0.3078	0.6186	0.4571	0.4221	–	0.5691	0.4641
6	0.6990	0.3669	0.3206	–	0.6624	0.4918	–	0.9311	0.7205

m=10									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.4055	0.3529	–	0.7054	0.4944	–	–	0.8418
2	0.6182	0.3862	0.3620	–	0.4966	0.4745	–	0.9839	0.5232
3	–	0.3992	0.3131	–	0.6598	0.4393	–	–	0.6800
4	0.6696	0.3864	0.3507	–	0.5463	0.4617	–	–	0.5592
5	0.4862	0.3684	0.3214	0.9040	0.4735	0.4263	–	0.6861	0.4893
6	–	–	0.4543	–	–	–	–	–	–

m=15									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.3948	0.3224	–	0.7333	0.4744	–	–	0.8699
2	0.7562	0.3876	0.3497	–	0.5085	0.4710	–	–	0.5532
3	–	0.3819	0.3261	–	0.7236	0.4473	–	–	0.6668
4	–	0.3874	0.3404	–	0.6830	0.4590	–	–	0.8159
5	0.6018	0.3785	0.3181	–	0.4923	0.3948	–	–	0.5409
6	–	–	0.3759	–	–	0.9209	–	–	–

m=20									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.4145	0.3473	–	0.9349	0.4802	–	–	–
2	–	0.3913	0.3635	–	0.5413	0.4785	–	–	0.5986
3	–	0.4092	0.3401	–	0.9275	0.4624	–	–	0.6938
4	–	0.3885	0.3337	–	–	0.5024	–	–	–
5	0.7755	0.3804	0.3082	–	0.5422	0.4020	–	–	0.6386
6	–	–	0.3844	–	–	–	–	–	–

Tabelle B.2: Schwellen zu Serie 1, Laplacepyramide

m=5									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.2314	0.3760	–	–	0.2363	–	–	–
2	0.4284	0.5067	0.4465	0.4841	–	0.5287	0.5445	–	0.4067
3	–	–	0.4878	–	–	–	–	–	–
4	0.4898	0.4986	0.4726	0.4941	0.4822	0.6546	0.5011	0.5035	0.5070
5	–	–	–	–	–	–	–	–	–
6	0.3135	0.5169	0.4999	0.4064	0.5122	0.4757	0.3906	0.4567	0.4734
m=10									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.3108	0.3170	–	0.4746	0.4145	–	–	0.3349
2	0.4455	0.5200	0.5147	0.4576	0.5162	0.4236	0.4753	0.5225	0.3035
3	–	–	–	–	–	0.5203	–	–	0.5341
4	–	0.4759	0.4998	0.4923	0.4832	0.4978	0.5048	0.5061	0.5073
5	–	0.8449	0.6574	–	–	0.8342	–	–	0.7989
6	–	0.5427	0.4937	0.5125	0.5219	0.5002	0.4597	0.5221	0.5216
m=15									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.2954	0.3150	–	0.4931	0.4102	–	–	0.3057
2	0.4734	0.5057	0.5067	0.4817	0.5259	0.5213	0.4997	0.6062	0.5406
3	–	–	–	–	–	–	–	–	–
4	–	0.5117	0.4852	0.2501	0.4994	0.4894	0.5005	0.5050	0.5051
5	–	0.8703	0.7894	–	–	0.8825	–	–	0.8303
6	–	0.4041	0.4873	–	0.5011	0.5206	0.4873	0.5205	0.5063
m=20									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.3029	0.3206	–	0.4594	0.4082	–	–	0.3350
2	0.4498	0.5221	0.5231	0.4807	0.5245	0.5294	0.4912	0.5206	0.5093
3	–	–	–	–	–	–	–	–	0.3979
4	–	0.4715	0.5002	–	0.4907	0.4961	0.5038	0.5046	0.5033
5	–	–	0.7755	–	–	0.8377	–	–	–
6	–	0.4096	0.4422	–	0.4394	0.4764	0.5067	0.5129	0.5077

Tabelle B.3: Schwellen zu Serie 2, Gaußpyramide

m=5									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.4893	0.3242	–	–	0.5117	–	–	0.3397
2	0.3726	0.5090	0.4947	0.5030	0.4932	0.4878	0.5026	0.4892	0.5886
3	0.4049	0.4394	0.4371	0.3896	0.4051	0.4488	0.3844	0.3880	0.4931
4	0.4125	0.4025	0.4634	0.4737	0.4510	0.4689	0.4605	0.4310	0.2848
5	0.5231	0.4432	0.4671	0.5301	0.4555	0.4443	0.4874	0.4926	0.7601
6	–	0.5102	0.5421	0.4959	0.4819	0.4699	0.5042	0.4918	0.4985
m=10									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.4053	0.3529	–	–	0.4899	–	–	0.3088
2	–	0.5218	0.5077	0.4825	0.5140	0.4997	0.5225	0.5089	0.4951
3	–	0.4934	0.4755	0.4667	0.5329	0.4991	0.4857	0.5202	0.4829
4	–	0.4512	0.5029	0.5281	0.4656	0.4820	0.5274	0.4664	0.4671
5	0.4106	0.4872	0.4706	0.5038	0.4865	0.4800	0.5255	0.4952	0.4752
6	–	–	0.4316	–	0.6002	0.5005	0.5234	0.5478	0.4835
m=15									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.3937	0.3224	–	–	0.4730	–	–	0.3415
2	–	0.5052	0.4874	0.2550	0.5160	0.4912	0.4812	0.5123	0.5062
3	–	0.4352	0.5173	–	0.5192	0.5169	0.3873	0.5056	0.5051
4	–	0.4514	0.5130	0.2686	0.4505	0.5334	0.4059	0.4754	0.5148
5	–	0.4784	0.5006	0.4720	0.4907	0.4947	0.4872	0.4943	0.5081
6	–	–	–	–	0.5138	0.3577	–	0.5425	0.4361
m=20									
	b=0.5			b=0.75			b=0.90		
O	l=30	l=60	l=90	l=30	l=60	l=90	l=30	l=60	l=90
1	–	0.4131	0.3473	–	–	0.4731	–	–	–
2	–	0.4857	0.5146	–	0.5229	0.5132	0.3684	0.5214	0.5010
3	–	0.3575	0.4873	–	0.4793	0.5092	0.2992	0.4909	0.5138
4	–	0.5072	0.4861	–	0.5161	0.4982	0.2939	0.5078	0.4999
5	–	0.3989	0.5058	–	0.4754	0.5091	0.4241	0.4936	0.5006
6	–	–	–	–	0.5442	0.5429	0.4638	0.5682	0.5621

Tabelle B.4: Schwellen zu Serie 2, Laplacepyramide

m=5									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	–	–	–	–	–	–	–	–	–
2	–	–	0.3493	0.4634	0.4016	0.4736	0.4987	0.4750	0.4888
3	–	–	–	–	–	0.4401	–	–	0.4884
4	0.5761	0.6226	0.4710	0.5087	0.5381	0.5547	0.5011	0.5035	0.5070
5	–	–	–	–	–	–	–	–	–
6	–	–	0.4980	0.4057	0.1646	0.5235	0.3906	0.4567	0.4734
m=10									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	–	–	–	–	–	–	–	–	–
2	0.5827	0.5499	0.4729	0.5414	0.5579	0.5131	0.5141	0.5047	0.5037
3	–	–	0.5923	–	–	0.5444	–	–	0.5341
4	0.5821	0.6434	0.4807	0.5419	0.5613	0.5121	0.5048	0.5061	0.5073
5	–	–	–	–	–	0.7506	–	–	0.7989
6	–	–	0.5611	0.4333	0.5315	0.5908	0.4597	0.5221	0.5216
m=15									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	–	–	–	–	–	–	–	–	–
2	0.2835	0.5379	0.4909	0.4107	0.5495	0.5521	0.4798	0.5036	0.5078
3	–	–	0.5840	–	–	0.5634	–	–	–
4	0.5964	0.6706	0.5086	0.5338	0.5543	0.5123	0.5005	0.5050	0.5051
5	–	–	–	–	–	0.7796	–	–	0.8303
6	–	–	0.5715	0.4685	0.4733	0.5528	0.4873	0.5205	0.5063
m=20									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	–	–	–	–	–	0.3296	–	–	–
2	0.3216	0.5573	0.5008	0.4107	0.5588	0.5597	0.4905	0.5031	0.5134
3	–	–	0.4888	–	–	0.4117	–	–	0.3979
4	0.5688	0.5760	0.4924	0.5312	0.5262	0.5211	0.5038	0.5046	0.5033
5	–	–	–	–	–	–	–	–	–
6	–	–	0.6129	0.4569	0.3771	0.5855	0.5067	0.5129	0.5077

Tabelle B.5: Schwellen zu Serie 5, Gaußpyramide

m=5									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	–	–	–	–	–	–	–	0.2741	0.3431
2	0.5651	0.6176	0.6032	0.5378	0.5803	0.5573	0.4878	0.5011	0.5074
3	0.4737	0.4172	0.4351	0.3537	0.4365	0.4975	0.4034	0.4853	0.4978
4	–	–	–	0.4220	0.4935	0.4945	0.4008	0.4957	0.5061
5	–	0.6465	0.5707	0.4822	0.5359	0.5366	0.4928	0.4799	0.4652
6	–	0.4414	0.3934	0.3700	0.5194	0.5532	0.4952	0.5251	0.5287
m=10									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	–	0.2906	0.2359	0.2953	0.4174	0.4594	0.3547	0.3769	0.4306
2	0.5722	0.5886	0.7449	0.5837	0.5461	0.5516	0.5190	0.4978	0.5032
3	–	0.5757	0.6137	0.5500	0.5298	0.5245	0.5089	0.4799	0.4978
4	0.5050	0.3392	0.6865	0.4798	0.5079	0.5243	0.4597	0.4635	0.4784
5	0.5827	0.5942	0.5721	0.5273	0.5199	0.5235	0.4747	0.4685	0.4774
6	–	0.4180	–	0.4561	0.4805	0.4958	0.3865	0.3589	0.4724
m=15									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	0.5469	0.5652	0.2870	0.4028	0.4113	0.3691	0.2946	0.3284	0.3613
2	0.5832	0.6480	0.6603	0.5544	0.5451	0.5262	0.4912	0.4906	0.4935
3	0.5305	–	0.6425	0.5718	0.5291	0.5099	0.5189	0.4805	0.4790
4	0.4148	–	–	0.4705	0.4837	0.4912	0.4247	0.4124	0.4279
5	–	0.5786	0.5899	0.5531	0.5093	0.5104	0.5362	0.4581	0.4775
6	–	0.4174	–	0.4216	0.4684	0.4928	0.4478	0.4286	0.4395
m=20									
	b=0.5			b=0.75			b=0.90		
O	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15	l_O=5	l_O=10	l_O=15
1	0.4505	0.5844	0.5509	0.4990	0.5449	0.5363	0.4539	0.4486	0.4502
2	0.4828	0.6218	0.5181	0.5356	0.5417	0.5259	0.5265	0.5074	0.5010
3	–	0.5623	0.3700	0.5351	0.5255	0.5157	0.5027	0.4962	0.4877
4	0.5206	0.5649	0.6312	0.5292	0.5166	0.5368	0.5177	0.4513	0.4462
5	–	0.5370	0.6001	0.5152	0.5236	0.5128	0.5206	0.4889	0.4867
6	0.4414	0.4560	0.4840	0.4622	0.5141	0.5122	0.4705	0.4912	0.4736

Tabelle B.6: Schwellen zu Serie 5, Laplacepyramide

Literaturverzeichnis

[1] ALOIMONOS, YIANNIS: *Introduction: Active Perception Revisited.* In: ALOIMONOS, YIANNIS (Herausgeber): *Active Perception*, Computer Vision. Lawrence Erlbaum Associates, Inc., Publishers, 1993.

[2] ARKIN, E. M., L. P. CHEW, D. P. HUTTENLOCHER, K. KEDEM und J. S. B. MITCHEL: *An efficiently computable metric for comparing polygonal shapes.* PAMI, 13(3):209–215, 1991.

[3] BALLARD und BROWN: *Computer Vision.* Prentice Hall, 1982.

[4] BROOKS, RODNEY A.: *Intelligence without Reason.* Technischer Bericht A. I. Memo No. 1293, MIT AI Laboratory, 1991.

[5] BURT, PETER J.: *Fast Filter Transforms for Image Processing.* Computer Graphics and Image Processing, 16:20–51, 1981.

[6] BURT, P. J.: *The Pyramid as a Structure for Efficent Computation.* In: ROSENFELD, AZRIEL (Herausgeber): *Multiresolution Image Processing and Analysis*, Band 12 der Reihe *Springer Series in Information Sciences.* Springer-Verlag, 1984.

[7] CRAIG, JOHN J.: *Introduction to robotics: mechanics and control.* Addison-Wesley Series in Electrical and Computer Engeneering: Control Engeneering. Addison-Wesley Publishing Company, Inc., zweite Auflage, 1989.

[8] DANIILIDIS, KONSTANTINOS: *Hand-Eye Calibration with Dual Quaternions.* International Journal of Robotics Research, März 1999.

[9] FAUGERAS, OLIVIER: *Three-Dimensional Computer Vision: A Geometric Viewpoint.* The MIT Press, Cambridge, MA, 1993.

[10] GIROSI, F., M. JONES und T. POGGIO: *Regularization Theory and Neural Networks Architectures.* Neural Computation, 7:218–269, 1995.

[11] GONZALEZ, RAFAEL C. und RICHARD E. WOODS: *Digital Image Processing.* World Student Series. Addison-Wesley Publishing Company, Inc., 1992.

[12] Harnad, Stevan: *The Symbol Grounding Problem.* Physica D 42, Seiten 335–346, 1990.

[13] Hough, P. V. C.: *Methods and Means for Recognizing Complex Patterns.* U. S. Patent 3,069,654. U.S. Patent Office, 1962.

[14] Jähne, B.: *Digitale Bildverarbeitung.* Springer-Verlag, vierte Auflage, 1997.

[15] Jain, Ramesh: *Machine Vision.* International Edition, 1995.

[16] Leavers, V. F.: *Which Hough Transform?* CVGIP: Image Understanding, 58(2):250–264, September 1993.

[17] Leonardis, Alěs und Horst Bischof: *Computational Complexity Reduction in Eigenspace Approaches.* In: Sommer, Gerald, Kostas Daniilidis und Josef Pauli (Herausgeber): *Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science, Seiten 1–8. Springer Verlag, 1997.

[18] Minsky, Marvin: *Logical Versus Analogical or Symbolic versus Connectionist or Neat Versus Scruffy.* AI Magazine, Seiten 34–51, 1991.

[19] Mundy, J., A. Liu, N. Pillow, A. Zisserman, S. Abdallah, S. Utcke, S. Nayar und C. Rothwell: *An Experimental Comparison of Appearance and Geometric Model Based Recognition.* In: *International Workshop on Object Representations in Computer Vision.* in accociation with ECCV, 1996.

[20] Murase, Hiroshi und Shree K. Nayar: *Visual Learning and Recognition of 3-D Objects from Appearance.* International Journal of Computer Vision, 14:5–24, 1995.

[21] Murase, Hiroshi und Shree K. Nayar: *Detection of 3D objects in cluttered scenes using hierarchical eigenspace.* Pattern Recognition Letters, 18:375–384, 1997.

[22] Nayar, Shree K., Simon Baker und Hiroshi Murase: *Parametric Feature Detection.* IEEE Conference on Computer Vision and Pattern Recognition, Seiten 471–477, 1996.

[23] Nayar, Shree K., Hiroshi Murase und Sameer A. Nene: *Learning, Positioning and Tracking Visual Appearance.* IEEE Conference on Computer Vision and Pattern Recognition, Seiten 3237–3244, 1994.

[24] Nene, Sameer A. und Shree K. Nayar: *A Simple Algorithm for Nearest Neighbor Search in High Dimensions.* Technischer Bericht CUCS-030-95, Department of Computer Science, Columbia University, New York, N. Y. 10027, 1995.

[25] Nene, Sameer A. und Shree K. Nayar: *Closest Point Search in High Dimensions.* IEEE Conference on Computer Vision and Pattern Recognition, Seiten 859–865, 1996.

[26] O'Gorman, Lawrence und Arthur C. Sanderson: *The Converging Squares Algorithm: An Efficient Method for Locating Peaks in Multidimensions.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(3):280–288, März 1984.

[27] Pauli, J.: *Projective Invariance and Orientation Consensus for Extracting Boundary Configurations.* In: Paulus, E. und F. M. Wahl (Herausgeber): *Mustererkennung, Tagungsband, 19. DAGM-Symposium*, Seiten 375–383. Springer-Verlag, 1997.

[28] Poggio, Tomaso und Frederico Girosi: *Networks for Approximation and Learning.* Proceedings of the IEEE, 78(9):1481–1496, September 1990.

[29] Press, W. H., S. A. Teukolsky, W. T. Vetterling und B. P. Flannery: *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, zweite Auflage, 1992.

[30] Rothwell, Charlie, Joe Mundy und Bill Hoffman: *Representing Objects using Topology.* In: *International Workshop on Object Representations in Computer Vision.* in accociation with ECCV, 1996.

[31] Sarkar, Sudeep und Kim L. Boyer: *Perceptual Organization in Computer Vision: A Review and a proposal for a Classificatory Structure.* IEEE Transactions on Systems, Man and Cybernetics, 23(2):382–399, März/April 1993.

[32] Sarkar, Sudeep und Kim L. Boyer: *A Computational Structure for Preattentive Perceptual Organization: Graphical Enumeration and Voting Methods.* IEEE Transactions on Systems, Man and Cybernetics, 24(2):246–267, Februar 1994.

[33] Shiu, Yiu Cheung und Shaheen Ahmad: *Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form $AX = XB$.* IEEE Transactions on Robotics and Automation, 5(1):16–29, Februar 1989.

[34] Tan, Y. und H. Freeman: *The Surface-Attribute Probe: A "Smart" Tool for 3-D Object Recognition.* In: *Prooceedings of the Workshop on Visual Behaviors, Seatlle, Washington.* IEEE Computer Society Press, Juni 1994.

[35] Tsai, Roger Y. und Reimar K. Lenz: *A New Technique for Fully Autonomous and Efficient 3D Robotis Hand/Eye Calibration.* IEEE Transactions on Robotics and Automation, 5(3):345–358, Juni 1989.

[36] Turk, Matthew und Alex Pentland: *Eigenfaces for Recognition.* Journal of Cognitive Neuroscience, 3(1):71–86, 1991.

[37] Vinod, V. V., S. Chaudhury, S. Ghose und J. Mukherjee: *A Connectionist Approach for Peak Detection in Hough Space.* Pattern Recognition, 25(10):1253–1264, 1992.

[38] Wang, Ching-Cheng: *Extrinsic Calibration of a Vision Sensor Mounted on a Robot.* IEEE Transactions on Robotics and Automation, 8(2):161–175, April 1992.

[39] Wei, Guo-Qing und Song De Ma: *Implicit and Explicit Camera Calibration: Theory and Experiments.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(5):469–480, Mai 1994.

[40] Yang, Xiaopu, Tapan K. Sarkar und Ercument Arvas: *A Survey of Conjugate Gradient Algorithms for Solution of Extreme Eigen-Problems of a Symmetric Matrix.* IEEE Transactions on Acoustics, Speech and Signal Processing, 37(10):1550–1556, Oktober 1989.

[41] Ylä-Jääski, Antii und Frank Ade: *Grouping Symmetrical Structures for Object Segmentation and Description.* Computer Vision and Image Understanding, 63(3):399–417, Mai 1996.

[42] Zhang, Z. und O. Faugeras: *3D Dynamic Scene Analysis.* Springer Series in Information Science. Springer-Verlag Berlin Heidelberg, 1992.

[43] Zisserman, Andrew, David Forsyth, Joe Mundy, Charlie Rothwell, Jane Liu und Nic Pillow: *3D Object Recognition using Invariance.* Artificial Intelligence, 78:239–288, 1995.

Danksagung

Für Diskussionen und Anregungen bedanke ich mich bei J. Pauli, J. Bruske, K. Daniilidis und A. Schmidt, für die technische Unterstützung bei G. Diesner.

Mein besonderer Dank gilt meinen Eltern.

Erklärung

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

Kiel, den 6. Januar 1999