Lehrstuhl für Steuerungs- und Regelungstechnik

Ansichtsbasierte Objekterkennung mit Hilfe optimierter Musterbäume

Eugen M. Ettelt

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. techn. J. Swoboda
Prüfer der Dissertation:	
	1. Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. G. Schmidt
	2. apl. Prof. Dr.-Ing., Dr.-Ing. habil. G. Ruske

Die Dissertation wurde am 24. 10. 2001 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.02.2002 angenommen.

Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Steuerungs- und Regelungstechnik der Technischen Universität München.

An dieser Stelle möchte ich dem Lehrstuhlinhaber, Prof. Dr.-Ing. Dr.-Ing. E.h. G. Schmidt, danken, der diese Arbeit nicht nur angeregt und ermöglicht hat, sondern durch Integration der Arbeit in das ROMAN-Projekt und die damit gegebenen, unkonventionellen Anforderungen auch die Voraussetzungen für die entsprechenden Ideen und Lösungen geschaffen hat.

Mein Dank gilt insbesondere auch Herrn Prof. Dr.-Ing. Ruske für die Übernahme des Korreferats und das Interesse an der Arbeit.

Die Arbeit, insbesondere ihr experimenteller Teil, wäre ohne die Unterstützung meiner Kollegen am Lehrstuhl nicht möglich gewesen, insbesondere möchte ich Herrn Dr. Freyberger für die kontinuierliche Hilfe danken, die wesentlich zum Gelingen beigetragen hat.

Den Kollegen im ROMAN-Team, W. Daxwanger, R. Furtwängler, C. Fischer und U. Hanebeck gilt mein Dank für die meist konstruktive Zusammenarbeit, ebenso den Herrn Kubick, Ritt, Gradl, Stöber und Jaschik von der mechanischen und elektrischen Werkstatt.

München, im Oktober 2001 Eugen Ettelt

Inhaltsverzeichnis

Ansichtsbasierte Objekterkennung mittels Musterbäumen

Kurzfassung Gegenstand dieser Arbeit ist die schnelle, ansichtsbasierte Objekterkennung in Videosequenzen. Das entwickelte Verfahren verwendet als implizites Modell eines Objekts nur Aufnahmen des Objekts und Aufnahmen der typischen Objektumgebung als Zurückweisungsklasse. Mittels eines binären Entscheidungsbaumes werden diese Ansichten für die Erkennung des Objekts mit Kamerabildern verglichen. Zu diesem Zweck wird in kleinen Schritten ein Suchfenster über das Kamerabild geschoben und nach jeder Verschiebung festgestellt, ob der Inhalt des Suchfensters einer Ansicht des Objekts oder einer Ansicht der Zurückweisungsklasse ähnlicher ist. Das Ergebnis einer solchen Klassifikation ist die Information, zu welcher Klasse der Bildausschnitt gehört. Diese Information läßt sich mathematisch exakt erfassen und dient als Gütekriterium für die optimale Dimensionierung des Klassifikators. Diese Auslegung führt zu einer Verkleinerung der Entscheidungsbäume, und damit zu einer schnellen und robusten Erkennung. Für die Geschwindigkeitssteigerung des gesamten Erkennerverfahrens wurden zusätzlich noch zwei unterschiedliche Verfahren zur effizienten Vor-Auswahl von Information entwickelt, die eine Objekterkennung in Echtzeit erlauben. Dadurch kann das Verfahren auch für die Objekterkennung und -verfolgung in einer Videosequenz eingesetzt werden. Eine Fortschreibung und Prädiktion der Objekttrajektorie in einer Videosequenz wird durch eine Hypothesenfortschreibung ermöglicht; die hohe Dynamik des Erkenners ermöglicht damit auch den Einsatz des Erkenners in geschlossenen Regelkreisen. Die Praxistauglichkeit, insbesondere die Robustheit der entwickelten Verfahren wird beispielhaft mit einer prototypischen Implementierung für den mobilen Serviceroboter ROMAN und einer schnellen Gesichtsdetektion nachgewiesen.

Abstract This thesis describes a method for fast, appearance-based object recognition within a videostream. The model for object recognition consists only of images of the object and images of the typical object environment as rejection class. For purposes of object recognition a small search window is shifted over the camera image and after each shift the content of the window is compared with the model of the object and the rejection class. This comparision is done with a binary decision tree. The classification result being the information on the class can be calculated exactly and serves as objective function for the optimization of the classifier. This approach leads to small and therefore robust and fast decision trees. In addition two undersampling methods have been integrated into the algorithm for efficent pre-selecting of appropriate information. They make object recognition in real-time possible. This enables fast object tracking and object trajectory following as well as a prediction of the object's movement for high dynamic closed-loop applications. The object recognition system has been applied on the mobile service-robot ROMAN as part of an intelligent module for object grasping. The experiments demonstrate the robustness and the real-time capability of the proposed algorithm. Furthermore fast face recognition has been performed and shows the wide variety of objects being able to be recognised.

1 Einleitung

1.1 Videosensorik

Bildverstehen ist sicherlich eine der großen Herausforderung der modernen Informatik und umfaßt neben der konventionellen Bildverarbeitung auch die Interpretation des Bildinhaltes. Es erlaubt automatisierungstechnischen Systemen eine preiswerte Sensierung numerischer und symbolischer Information, beispielsweise die Erkennung und Lokalisierung von Objekten, die Hinderniserkennung und die Bestimmung von Form und Farbe eines Objekts.

Alle derartigen Anwendungen bedürfen der Kenntnis des meist impliziten Zusammenhangs zwischen dem von der Kamera aufgenommenen Bild und der gewünschten Information. Der Zusammenhang zwischen Bild und gewünschter Information muß als Vorwissen in dem Bildverarbeitungssystem enthalten sein. Je mehr Effekte der optischen Abbildung berücksichtigt werden, desto größer muß dieses Vorwissen sein. Außerdem wird die Abbildung durch Störgrößen beeinflußt wie variable Beleuchtung und Verdeckung durch andere Objekte.

Sobald ein automatisiertes System nicht nur die Umgebung beobachtet, sondern in diese Umgebung auch aktiv eingreift, kann es die Wirkung der eigenen Aktorik wahrnehmen. Dieser Vorgang findet aber nicht nur wie in einem klassischen Regelkreis auf rein numerischer Ebene statt. So kann ein Roboter videogestützt feststellen, wieviele Objekte des betreffenden Typs im Greifbereich sind und wie jedes Objekt gegriffen werden kann. Beim Greifvorgang selbst können die einzelnen Finger einer Roboterhand videogestützt positioniert werden und eventuell korrigiert werden. Bei diesem als 'visual servoing' bekannten Verfahren kann auf eine präzise, schwere und teure Roboterkonstruktion verzichtet werden; die erforderliche Genauigkeit wird durch den Regelkreis sichergestellt. Mißlingt ein Greifvorgang, wird dies durch die Videosensorik erkannt und entsprechend korrigiert. Ist kein Objekt des spezifizierten Typs im Greifbereich des Roboters, wird keine Aktion durchgeführt. Für den Einsatz in der Rückführung der Steuerung muß Videosensorik natürlich in Echtzeit arbeiten, um nicht den gesamten Vorgang zu verlangsamen.

Natürlich bedarf es für derartige Steuerungen auch geeigneter numerisch und symbolisch arbeitender Algorithmen, die die rückgekoppelte Information in zielorientierte Aktionen umwandeln. Diesen Algorithmen fällt ferner auch die Aufgabe zu, die für das Problem relevante Information von der Videosensorik anzufordern. Aus Zeitgründen ist eine Erfassung der gesamten, möglichen Information selten möglich. Beispielsweise muß der Sensorik der Typ des zu erkennenden Objekts oder die Blickrichtung

mitgeteilt werden. Diese Fokusierung der Sensorik auf die gewünschte Information wird um so wichtiger, je mehr unterschiedliche Informationen angefordert werden können.

Eine größere Vielfalt unterschiedlicher Information erweitert die Flexibilität technischer Systeme und somit das Spektrum ihrer möglichen Anwendungen. Kann dadurch auf menschliche Steuereingriffe ganz oder teilweise verzichtet werden, spricht man von Autonomie oder Teilautonomie des betreffenden technischen Systems.

1.2 Problemstellung

Ein solches teilautonomes System stellt der Serviceroboter ROMAN nach Abb. 6.1 dar. Er dient zur Entlastung von Servicepersonal im medizinischen Bereich. Typische Aufgaben sind die sogenannten Hol-Bring Aufgaben, in denen der Serviceroboter ein bestimmtes Objekt von einem spezifizierten Ort, beispielsweise einem Tisch, zu einem anderen Ort bringen soll. Eine Ablaufsteuerung nach [16] nimmt Befehle vom Anwender entgegen und plant entsprechende Aktionen des Roboters, sowohl für die Fahrbewegung der Plattform - eventuell in andere Räume - als auch für den Greifvorgang des Roboterarms. Bei diesem Greifvorgang wird ein Objekt zunächst visuell erkannt, lokalisiert und anschließend gegriffen.

ABBILDUNG 1.1: Greifen einer Flasche durch den semiautonomen Serviceroboter ROMAN und Ausschnitt eines typischen Kamerabildes.

Aufgabenstellung in diesem Kontext ist die *Erkennung eines Objekts*, ein Beispiel für die Extraktion symbolischer Information. Bei dem hier betrachteten Serviceroboter können die Objekte beispielsweise Flaschen oder Gläser sein, die entsprechend Abb. 1.1 auf einem Tisch stehen. Aus dem Szenario ergeben sich folgende Anforderungen an die Objekterkennung:

- Die Erkennung eines Objekts soll so *schnell* erfolgen, daß das Annäherungs- und Greifmanöver nicht durch die Objekterkennung verzögert wird. Die Zeit für die Objekterkennung muß deutlich unterhalb einer Sekunde liegen, um die Roboterbewegung in Hinblick auf eine günstige Greifposition möglichst verzögerungsfrei beeinflussen zu können.
- Die *Genauigkeit* der Objektlokalisierung muß für den Greifvorgang im Bereich weniger Millimeter liegen. Bereits während der Annäherung des Fahrzeugs an das Objekt soll frühzeitig eine ungefähre Lage des Objekts bekannt sein, um die Plattformbewegung zu beeinflussen. Diese zunächst ungenaue Lokalisierung wird während der Annäherung verfeinert bis zum eigentlichen Greifvorgang.
- Dieser Sachverhalt führt zu der nächsten Anforderung, daß das Objekt nicht nur in einem Bild erkannt wird, sondern während dem ganzen Vorgang *kontinuierlich verfolgt* wird. Dies erhöht neben der Genauigkeit der Objektlokalisierung auch die Robustheit. Zum einen führen viele Erkennungen zu einer geringeren Wahrscheinlichkeit, ein anderes Objekt als das gesuchte zu erkennen. Zum anderen können Störungen wie das Entfernen oder das Verschieben des Objekts während der Annäherung erkannt werden.
- Die Objekterkennung soll *flexibel auf neue Objekte* angepaßt werden können. Dementsprechend sollen weitere Objekte von einem Anwender ohne Spezialisten-Kenntnisse hinzugefügt werden können. Die Arbeit soll insbesondere ohne Verändern der Programme zur Objekterkennung möglich sein. Es sollen lediglich objektspezifische Datensätze erzeugt werden, die von den selben Programmen verarbeitet werden. Diese Datensätze sollen auch modifizierbar sein, wenn während des Einsatzes des Erkenners Fehl-Erkennungen auftreten und gezielt ausgeschlossen werden sollen.
- Die Objekterkennung muß Objekte erkennen können, die in ihrer *Struktur sehr verschieden* sind. Abb. 1.2 zeigt Objekte aus dem Serviceszenario mit unterschiedlicher Ausprägung. Enthalten sind Objekte mit geraden und elliptischen Kanten und Objekte, die durch die Beleuchtung maßgeblich verändert werden wie das Glas. Ein Algorithmus soll nicht durch eine Vorauswahl von Merkmalen wie geraden Kanten in seiner Flexibilität eingeschränkt werden und grundsätzlich die Erkennung von Objekten unmöglich machen, die derartige Merkmale nicht besitzen.

ABBILDUNG 1.2: Typische Objekte: Thermostat, Dose, Flasche, Glas, Schachtel und Spitzer.

Eine weitere Aufgabenstellung ist die Detektion von menschlichen Gesichtern im Kamerabildfeld. Gegeben ist also ein Bild oder eine Bildfolge, die ein beliebiges Gesicht enthält oder nicht. Die Objekterkennung soll herausfinden, in welchen Bildern und wo auf diesen Bildern ein Gesicht zu finden ist. Resultat der Objekterkennung ist für jedes Bild die Angabe, wieviele Gesichter enthalten sind und jeweils die *Bildkoordinaten*.

1.3 Stand der Technik

1.3.1 Merkmalsbasierte Ansätze

Merkmalsbasierte Objekterkennung geht von *Modellprimitiven* wie Kanten oder Flächen aus. Mittels dieser Primitive wird ein 3D-Objektmodell erstellt, ähnlich der Modellierung in CAD-Programmen. Dieses offline generierte 3D-Objektmodell wird zum Zeitpunkt der Erkennung mit dem Bild verglichen.

Der Vergleich des 3D-Objektmodells mit dem Bild zerfällt in 2 Arbeitsschritte. Im 1. Schritt werden im Bild Strukturen gesucht, die den Modellprimitiven entsprechen. Besteht beispielsweise das 3D-Modell aus Kanten, werden im Bild ebenfalls Kanten gesucht. Diese Elemente sind also eine Art ´Sprachelemente´ oder Zwischenrepräsentation, in der sowohl das 3D-Modell des Objekts als auch seine 2D-Abbildung beschrieben werden können. In einem 2. Schritt werden die aus dem Bild extrahierten 2D-Bildprimitive den 3D-Modellprimitiven des Objekts zugeordnet. Ergebnis dieser Zuordnung ist die Aussage, ob das Objekt abgebildet ist, und, falls ja, die geometrische Lage relativ zur Kamera. Einen umfassenden Überblick zu dieser Thematik gibt [49].

Die Verfahren zur Primitivzuordnung sind zumindest theoretisch unabhängig von der Position des Objekts, d.h. sie können das Objekt in 6 Freiheitsgraden erkennen. Trotzdem unterliegen die merkmalsbasierten Verfahren erheblichen Einschränkungen.

- Zunächst müssen die verwendeten Primitive in der Lage sein, das gesuchte Objekt zu modellieren. Besonders kontinuierliche Grauwertverläufe, Texturen und komplexe Kantenverläufe lassen sich kaum mit den bekannten Primitiven modellieren. Je nach Art der verwendeten Primitive geht zuviel Information verloren.
- Des weiteren ist nicht bekannt, welche Teile des Objekts für die Erkennung maßgeblich sind und zur Diskriminierung gegenüber anderen Objekten beitragen. Dies gilt insbesondere dann, wenn andere, verwechslungsrelevante Objekte nicht bei der Modellierung betrachtet werden.
- Die Bestimmung der Bildprimitive ist abhängig von der Beleuchtung und der Lage des Objekts. Die Erkennung der Objekte in allen 6 Freiheitsgraden wird dadurch erheblich eingeschränkt.

Da die in dieser Arbeit betrachteten Objekte sehr unterschiedlich strukturiert sind, treffen die Einschränkungen für den 1. Arbeitsschritt besonders zu. Die in der Problemstellung geforderte, halbautomatische Modellierung erscheint mit merkmalsbasierten Ansätzen kaum möglich. Eine geeignete Anpassung des Modells und der verwendeten Modellprimitive kann empirisch durch Expertenwissen erfolgen. Allerdings muß in einem aufwendigen, iterativen Prozeß die Auswirkung unterschiedlicher Modellierungen auf die Erkennung eines bestimmten Objekts untersucht werden. Beispiele für diese Probleme finden sich in [36] und [72].

Auch der 2. Arbeitsschritt der merkmalsbasierten Erkennung beinhaltet ein grundlegendes Problem. Je nach Verfahren muß entweder im Verlauf oder am Schluß des Zuordnungsprozesses entschieden werden, ob die Abbildung zu dem gesuchten Objekt paßt. Dazu wird ein Paß- oder Gütemaß definiert. Die Definition dieses Gütemaßes orientiert sich an diskreten Größen, beispielsweise an der Anzahl der für die Zuordnung verwendeten Merkmale und an kontinuierlichen Größen, beispielsweise wie genau die Modellmerkmale zu den Bildmerkmalen passen. Eine Gewichtung solcher unterschiedlichen Größen erfolgt meist empirisch. Außerdem wird oft ebenfalls nur empirisch eine Schwelle für das Gütemaß festgelegt, oberhalb der die Zuordnung als korrekt angenommen wird.

Die Festlegung der Schwelle dient dazu, das gesuchte Objekt von anderen Objekten zu unterscheiden. Eine systematische Vorgehensweise erfordert deshalb auch eine Betrachtung, wie groß das Gütemaß bei anderen Objekten wird. Dieser Schritt wird in vielen Arbeiten vernachlässigt und führt zu einer heuristischen, experimentellen Vorgehensweise.

1.3.2 Ansichtsbasierte Ansätze

Während die Objekterkennung mittels 3D-Modellen seit mehreren Jahrzehnten verfolgt wird[1] und sehr intensiv diskutiert wurde, ist die ansichtsbasierte Objekterkennung ein vergleichsweise junges Thema, das zwar ebenfalls seit langem bekannt ist, aber nur wenig bearbeitet wurde. Die Gründe dafür liegen sicher in der notwendigen aber erst seit wenigen Jahren verfügbaren Rechenleistung, und dem Mangel an leistungsfähigen Vergleichsalgorithmen.

Trotzdem erreichen die ansichtsbasierten Verfahren bereits jetzt beachtliche Ergebnisse. Zu erwähnen ist hier vor allem die Arbeit von Pentland und Turk [70] zur ansichtsbasierten *Erkennung von Personen*. Das Verfahren benötigt ein Portraitbild der Person; das Gesicht muß dazu aufrecht und mit der Blickrichtung zur Kamera ausgerichtet sein. Außerdem soll es möglichst genau formatfüllend sein, was gegebenenfalls durch eine Normierung der Größe erreicht wird. Verwendet werden Bildgrößen im

1 Eine ausführliche Übersicht bietet [49].

Bereich 50*x*50 bis 256*x*256 Bildpunkte. Durch diese Voraussetzungen muß das Objekt nicht im Bild gesucht werden, die Position ist bereits bekannt und es muß nur klassifiziert werden. Ausgehend von einer kleinen Anzahl verschiedener Trainingsbilder wird zunächst eine Karhunen-Loève-Entwicklung[1] (kurz: KL-Entwicklung) durchgeführt. Die ersten Komponenten dieser Zerlegung werden als repräsentativ für beliebige Gesichter angesehen und in Anlehnung an den englischen Begriff ´eigenvector´ auch als *´eigenface´* bezeichnet. Um ein beliebiges Gesicht mit wenigen Zahlen zu beschreiben, wird es durch eine Linearkombination weniger KL-Komponenten approximiert. Die Gewichtung dieser Komponenten stellt eine niedrigdimensionale Kennung des Gesichts dar, die einmal bestimmt und abgespeichert wird. Zu Identifikationszwecken kann dann die Kennung eines unbekannten Gesichts mit den gespeicherten Kennungen verglichen werden, beispielsweise unter Verwendung der Nächsten-Nachbar-Regel. Das Verfahren nach Pentland und Turk hat den Vorteil, daß mit einer einmal generierten KL-Entwicklung eines kleinen Trainingssets beliebige Gesichter ausgedrückt und identifiziert werden können. Trotzdem sind für eine robuste Identifizierung eines bestimmten Gesichts mehrere Ansichten nötig, um die jeweilige Verteilung der KL-Komponenten zu bestimmen; jedes Gesicht wird also durch eine statistische Verteilung beschrieben.

In der Folge wurde deshalb das bekannte Klassifikationsverfahren nach Fisher [20] auf das Problem angewandt, das für lineare Klassifikatoren und normalverteilte Wahrscheinlichkeitsdichtefunktionen ein pragmatisches Verfahren zur Klassentrennung darstellt. Zur Beschleunigung wird dieses bereits mit Entscheidungsbäumen kombiniert, eine Implementierung dieses Verfahrens beschreiben Swets und Weng in [69]. Dort wird zunächst eine KL-Zerlegung durchgeführt und der hochdimensionale Vektorraum mittels einer Linearen Diskriminanz Analyse (LDA) in einen Vektorraum niedrigerer Dimension transformiert; die Klassifikation in dem dann niedrigdimensionalen Raum wird nach Fisher durchgeführt. Das Verfahren stellt also eine Kombination dreier bekannter Verfahren dar, nämlich des Entscheidungsbaums, der KL-Zerlegung mit anschließender LDA, und der Klassifikation nach Fisher. Die im gleichen Aufsatz beschriebenen Ergebnisse beschränken sich auf ein kleines Trainingsset mit insgesamt 1316 Mustern von 526 verschiedenen Objekttypen. Die meisten Objekte werden also nur durch 2 Trainingsbilder repräsentiert und können deshalb nur in einer stark eingeschränkten Situation erkannt werden. Die Klassifikation eines einzelnen Musters benötigt je nach Parametrierung der einzelnen Verarbeitungsschritte zwischen 2 s und 400 s auf einer Sparc-20. In [68] beschreiben Swets und Weng den Unterschied zwischen der KL-Zerlegung und der *Generalisierten KL-Zerlegung*[2]. Sie vergleichen beide

1 Auch als ´Principal Component Analysis´ oder ´PCA´ bekannt.

2 Die KL-Zerlegung beschreibt eine einzige Datenquelle optimal in Hinblick auf ein quadratisches Gütemaß. Die Generalisierte KL-Zerlegung gibt Vektoren an, die für die Trennung mehrerer Datenquellen optimal sind, ebenfalls in Hinblick auf ein quadratisches Gütemaß.

Verfahren experimentell und zeigen, daß die Generalisierte KL-Zerlegung zu niedrigeren Fehlerraten führt. Daneben werden auch weitere Kombinationsmöglichkeiten der Verfahren angesprochen; beispielsweise eine Kombination bestehend aus der normalen KL-Zerlegung, einer LDA zur Verringerung der Dimension des Musterraumes und anschließend einer Generalisierten KL-Zerlegung.

Ähnlich den Verfahren von Swets und Weng arbeitet der Algorithmus von Nayar [43]. Er ist ebenfalls konzipiert zur ansichtsbasierten Erkennung beliebiger, kleiner Objekte. Durch geeignete Algorithmen wird ein kleines Analysefenster ausgewählt und das enthaltene Objekt als eines von 100 bekannten Objekten klassifiziert. Diese Algorithmen zur Vorauswahl des Analysefensters eignen sich nur für eine Erkennung vor schwarzem Hintergrund. Durch diese Voraussetzung kann das Analysefenster mit einer einfachen Flächenextraktion bestimmt und auf eine vordefinierte Größe skaliert werden. In einem 2. Schritt wird dieses in der Größe normalisierte Fenster klassifiziert. Dazu wird für jedes der 3 Farbbänder Rot, Grün und Blau eine KL-Zerlegung durchgeführt und die jeweils 30 wichtigsten Hauptkomponenten bestimmt. In diesen Komponenten wird das zunächst unbekannte Muster mit jeweils 100 Ansichten von 100 Objekten, also insgesamt 10000 Ansichten verglichen und über einen speziellen Auswahlmechanismus dem nächstliegenden Muster zugeordnet. Nayar gibt zusätzlich eine einfache Systematik für die Generierung von Objektmustern an. Dazu wird die Objektansicht in einem einzigen, rotatorischen Freiheitsgrad verändert; die 360°-Drehung wird in 100 Intervalle von jeweils 3,6° unterteilt, was zu den 100 Aufnahmen pro Objekt führt. Die Beleuchtung oder andere geometrische Freiheitsgrade, wie die Entfernung, werden nicht variiert, aber bei der Erkennung teilweise kompensiert, da der Bildausschnitt in Größe und Helligkeit normalisiert wird. Als Zeit für die Klassifikation eines einzelnen Musters bei Verwendung einer leistungsfähigen Workstation gibt Nayar 700 ms an.

Eine andere, interessante Arbeit Nayars [42] beschäftigt sich mit der Veränderung von Objektansichten bei *Veränderung der Beleuchtung und Position*. Nayar setzt eine lambertsche[1] Oberfläche des Objekts voraus. Außerdem dürfen sich die Objekte nicht verdekken; auch Selbstverdeckung muß ausgeschlossen sein. Lediglich 3 Ansichten bei unterschiedlicher Beleuchtung sind nötig für ein analytisch formuliertes Modell, das alle möglichen Abbildungen eines solchen Objekts beschreibt. Nayar gibt eine Methode an, wie mittels dieses Modells der Suchbereich eines Erkenners eingeschränkt und dadurch die Suche vereinfacht werden kann. Für praktische Anwendungen erscheint das Verfahren nur in Sonderfällen geeignet zu sein, in denen die Voraussetzungen gegeben sind.

1 Eine Oberfläche mit ideal diffuser Reflexion.

Ein Beispiel für die Gesichtsdetektion[1] mittels neuronaler Netze ist die Arbeit von Rowley [56]. Dort werden bereits Trainingssets für die Objekt- und die Zurückweisungsklasse verwendet. Ein Suchfenster wird über das Bild geschoben, nach jeder Verschiebung wird der Fensterinhalt einer der beiden Trainingsmengen zugeordnet. Diese Zuordnung erfolgt mit einem neuronalen Multi-Layer Netz. Die verwendeten Netze sind nicht vollvernetzt, sondern weisen heuristisch festgelegte Teilvernetzungen auf. Die letzte Schicht dieser Netze besitzt nur ein einziges Neuron, das für Gesichter einen positiven, für Zurückweisungsmuster einen negativen Wert annehmen soll, 0 ist der Schwellwert für die Klassenzugehörigkeit. Trotz vergleichsweise kleiner Bilder von 320 x 240 Pixel und Mustern von nur 20 x 20 Pixel dauert eine Detektion 7,2 s.

Rowley gibt einige interessante Heuristiken an, beispielsweise wie die Anzahl der Zurückweisungsmuster verringert werden kann, und wie räumlich eng beieinander liegende Mehrfachdetektionen eines Gesichts bewertet werden können.

Ein weiteres Verfahren zur Gesichtsdetektion wurde von Sung [67] veröffentlicht. Dieses ebenfalls ansichtsbasierte Verfahren verwendet ein Suchfenster mit 19 x 19 Pixel, das über das Bild geschoben wird. Der Inhalt dieses Fensters wird als Vektor aufgefaßt. Für das Training werden die Vektoren der Trainingssets einer Vektorquantisierung unterzogen, deren Ergebnis sind 6 bis 12 Hyperellipsoide pro Klasse. Sung beschreibt eine Methode mit Objekt- und Zurückweisungsklasse, und ein Verfahren mit nur einer Objektklasse. Dazu werden unterschiedliche Distanzmaße untersucht und wie die Distanzmaße zu mehreren Hyperellipsoiden mittels neuronaler Netze ausgewertet werden können, um das Muster einer der beiden Klassen zuzuordnen. Verarbeitungszeiten gibt Sung allerdings nicht an.

1.4 Überblick über diese Arbeit

1.4.1 Objektsuche in Bildern[2]

Für die ansichtsbasierte Objektsuche in einem Bild wird ein Suchfenster in kleinen Schritten über das Bild geschoben. Nach jeder Verschiebung wird der Inhalt des Suchfensters als Objekt oder Nicht-Objekt klassifiziert. Entsprechend Abb. 1.3 wird das Suchfenster beginnend in der linken, oberen Ecke des Bildes um jeweils wenige Pixel nach rechts verschoben. Am Ende jeder Zeile wird es um wenige Pixel nach unten und dort an den Zeilenanfang verschoben. Das Verfahren wird bis zur rechten, unteren Bildecke fortgesetzt. Der Versatz zwischen zwei Suchfenstern wird ausführlich in

1 Gesichtsdetektion bedeutet das Auffinden eines beliebigen Gesichts in einem Bild. Gesichtserkennung bezeichnet die Zuordnung eines Gesichts zu einer bestimmten Person.

2 Der Abschnitt 1.4 soll einen Überblick über die in dieser Arbeit vorgeschlagenen Algorithmen zur Objekterkennung geben und das Verständnis der nachfolgenden Kapitel erleichtern. Die Algorithmen sind teilweise sehr stark vereinfacht beschrieben.

Abschnitt 4.2.1 behandelt. Er ist generell geringer als die Größe des Suchfensters; die Suchfenster überlappen sich also. In jedem Suchfenster wird der Inhalt klassifiziert und festgestellt, ob es sich um das gesuchte Objekt, hier ein Gesicht, handelt oder nicht.

Die Größe des Suchfensters wird so groß gewählt, daß es den wesentlichen Teil des Objekts abdeckt. Allerdings variiert die Größe der Objektabbildung je nach Entfernung zwischen Kamera und Objekt; das Fenster wird bei großer Objektentfernung einen Teil des Objekthintergrundes beinhalten. Das Klassifikationsverfahren ist allerdings so beschaffen, daß ein im Suchfenster sichtbarer Hintergrund die Klassifikation nicht beeinträchtigt; lediglich der Rechenaufwand steigt bei einem unnötig groß gewählten Suchfenster.

ABBILDUNG 1.3: Verschieben des Suchfensters über das gesamte Bild. Das Suchfenster wird beginnend in der linken, oberen Ecke pixel- und zeilenweise bis zur rechten, unteren Ecke über das Bild geschoben. Sobald das Suchfenster das gesuchte Objekt - wie eingezeichnet hier ein Gesicht - überdeckt, wird dieses erkannt. An allen anderen Positionen im Bild ist das Ergebnis der Erkennung ´Nicht-Objekt´.

1.4.2 Mustervergleich mittels Musterbäumen

Da zur Suche eines Objekts in einem Bild sehr viele Suchfenster klassifiziert werden müssen und die Echtzeitbedingung einer Robotikanwendung eingehalten werden muß, ist eine im Vergleich zu den in Abschnitt 1.3 zitierten Verfahren sehr schnelle Klassifikation nötig. Die hohe Geschwindigkeit ist mit Entscheidungsbäumen realisierbar. Eine Vorverarbeitung, beispielsweise eine Cosinustransformation oder KL-Zerlegung einzelner Suchfenster entfällt; die in den Entscheidungsbäumen verarbeiteten Daten stammen direkt von den Kamerabildern.

Der folgende Abschnitt zeigt vereinfacht die Wirkungsweise dieser Entscheidungsbäume. Der im vorliegenden Fall binäre Entscheidungsbaum enthält ausgehend von einer Wurzel zwei Knoten mit einem geeigneten Muster der Objektklasse (in Knoten 1) und der Zurückweisungsklasse (in Knoten 2). Die Erzeugung der Muster wird in Kapitel 4 erläutert. Das zu klassifizierende Muster M wird nun zunächst mit beiden Mustern verglichen, dazu wird im einfachsten Fall der euklidische Abstand zu den in Knoten 1 und 2 gespeicherten Mustern M_1 und M_2 berechnet; die Variablen i und j laufen über die Ortskoordinaten der Muster M und M_1 bzw. M_2:

$$d_1 = \sum_i \sum_j (M(i,j) - M_1(i,j))^2$$

$$d_2 = \sum_i \sum_j (M(i,j) - M_2(i,j))^2$$

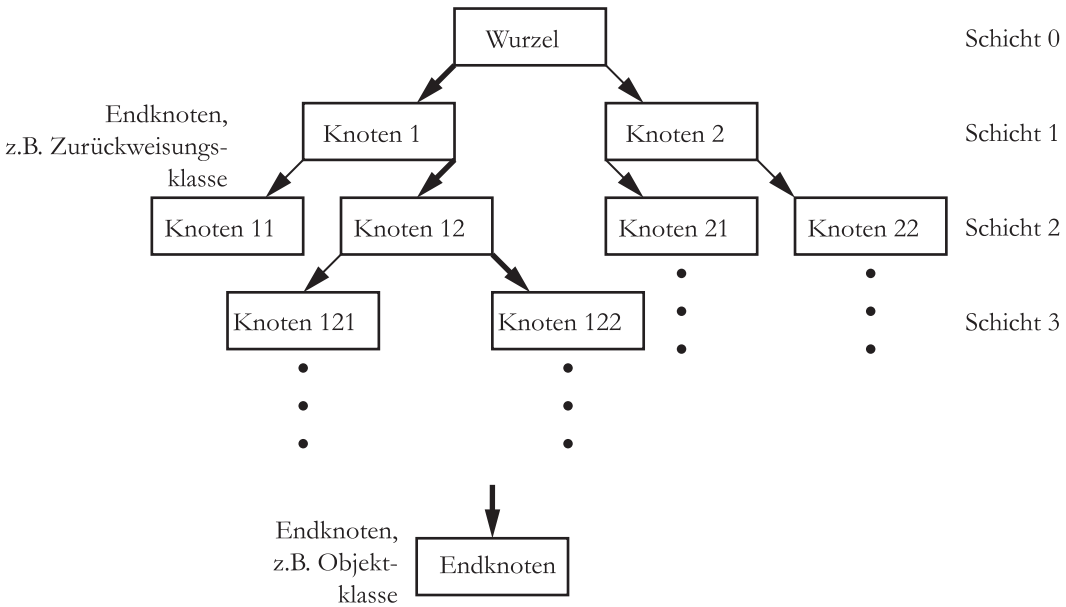

ABBILDUNG 1.4: Binärer Musterbaum zur Klassifikation.

Im Beispiel nach Abb. 1.4 ist das Muster M dem Muster M_1 in Knoten 1 ähnlicher, d.h. $d_1 < d_2$. Diese Teilklassifikation unterteilt den von einem Muster aufgespannten hochdimensionalen Raum[1] in zwei durch eine Hyperebene geteilte Halbräume. In der Folge werden für die Klassifikation dieses zunächst unbekannten Musters M nur noch die Nachfolger von Knoten 1 betrachtet. Die Knoten 2 nachfolgenden Knoten werden für die Klassifikation dieses Musters nicht weiter verwendet. Eine optimale Methode zur Parametrierung dieser Ebene wird in Kapitel 2 dargelegt, außerdem werden neben der Ebene auch quadratische Trennflächen betrachtet.

Da Knoten 1 kein Endknoten des Baums ist, wird das unbekannte Muster nun in einem zweiten Schritt mit den in den Knoten 11 und 12 abgespeicherten Mustern verglichen. Daraus wird wieder eine binäre Entscheidung gefällt, die im vorliegenden Beispiel zu Knoten 12 führt. Eine Verzweigung zu dem Endknoten 11 würde bereits zu einer endgültigen Entscheidung führen, im Beispiel zu einer Klassifikation als Zurückweisungsklasse. Das Klassifikationsresultat *´Objekt´* oder *´Nicht-Objekt´* ist jeweils in den Endknoten symbolisch abgespeichert.

Der wesentliche Vorteil dieser Suche in binären Entscheidungsbäumen besteht in der *sukzessiven Einschränkung* des Suchbereichs mit jeder Teilklassifikation. Es wird also nicht eine sehr große Informationsmenge extrahiert und anschließend ausgewertet. Statt dessen wird stets nur wenig Information extrahiert, diese aber unmittelbar anschließend verwendet, um die Komplexität des Suchproblems einzuschränken.

Ein weiterer Vorteil dieses Verfahrens besteht in der einfachen Möglichkeit, die Anzahl der Pixel für den Mustervergleich und damit den Rechenaufwand zu reduzieren. Dazu wird in den einzelnen Schichten des Baums eine unterschiedliche Auflösung zum Vergleich der Muster verwendet, beginnend in der ersten Schicht bei kleiner Auflösung bis zu voller Auflösung bei tieferen Schichten. Diese Methode der Unterabtastung wird detailliert in Abschnitt 4.2 diskutiert.

1.4.3 Generierung von Musterbäumen

Für die Objekterkennung wurde vorausgesetzt, daß bereits ein geeigneter Musterbaum vorhanden ist. Zur Generierung dieses Musterbaums wird eine Menge von Objektmustern benötigt und eine zweite Beispielmenge von Mustern, die zeigen, wie das Objekt nicht aussieht. Damit läßt sich aus dem Algorithmus der musterbaumgestützten Klassifikation direkt ein Verfahren zur Erzeugung des Baums angeben. Die N_1 Muster der Objektklasse $M_{1i}, i \in 1, 2 \ldots N_1$ werden gemittelt und das Ergebnis der Mittelung

1 Bei einer typischen Mustergröße von 31x31 Pixel ein 961-dimensionaler Raum.

$$\overline{M_1} = \frac{1}{N_1}\sum_{i=1}^{N_1} M_{1i}$$

werde in Knoten 1 als Muster abgespeichert. Entsprechend werden die N_2 Muster $M_{2i}, i \in 1, 2 \ldots N_2$ der Zurückweisungsklasse gemittelt und das Ergebnis $\overline{M_2}$ in Knoten 2 gespeichert. Bei einer testweisen Klassifikation der gesamten Trainingsmenge wird festgestellt, daß in Knoten 1 neben Mustern der Objektklasse auch Muster der Zurückweisungsklasse fallen, entsprechend fallen in Knoten 2 ebenfalls Muster beider Klassen. In einem zweiten Schritt werden nun die Muster in den Knoten 11 und 12 gebildet. Die Mittelwerte berechnen sich allerdings nicht aus den Mittelwerten der gesamten Objekt- bzw. Zurückweisungsklasse, sondern nur aus denjenigen Mustern, die bei der testweisen Klassifikation in Knoten 1 gefallen sind. In Knoten 11 wird also der Mittelwert aller Objektmuster gespeichert, die in Knoten 1 gefallen sind und in Knoten 12 der Mittelwert aller Zurückweisungsmuster, die in Knoten 1 gefallen sind. Das Verfahren wird in allen Knoten rekursiv fortgesetzt, bis bei testweiser Klassifikation in jeden Endknoten nur noch Muster einer Klasse fallen. Dies ist im Beispiel nach Abb. 1.4 bei Knoten 11 der Fall.

1.4.4 Erzeugung der Trainingsmenge

Während die Methode der Musterbäume allgemein zur Klassifikation beliebiger Muster verwendbar ist, müssen die Muster selbst aufgabenspezifisch für jedes Objekt festgelegt werden. Die Muster sollen die bei der Erkennung möglichen Situationen, mit genügend feiner Rasterung aller Variablen, genügend fein abdecken, also ohne ´Löcher´ und Häufungspunkte. Variablen sind vor allem die Beleuchtung, die relative Lage des Objekts zur Kamera und der Hintergrund. Nach der Aufnahme der Trainingsbilder des Objekts muß in jedem dieser Bilder ein Muster als Objektmuster angegeben werden. Dazu wird mit einem Programm das Szenenbild vergrößert dargestellt und der Mittelpunkt des Musters manuell markiert, vergleiche Abb. 6.5. Als Mittelpunkt wird ein beliebiger, aber bestimmter Objektbezugspunkt ausgewählt. Im Fall des Glases nach Abb. 1.2 ist dies beispielsweise der Übergang des oberen, weiten Zylinders zum unteren, engeren Fuß des Glases. Dieser Punkt wird bei der Erkennung als ´Glas´ erkannt und ist später für die Objektlokalisierung maßgebend. Er sollte so gewählt werden, daß sich die Muster möglichst ähneln, wodurch die Klassifikation einfacher wird. Beispielsweise sollte bei einem flachen und im Grundriß quadratischen Objekt der Mittelpunkt des Quadrats angegeben werden und nicht eine der vier Ecken, die je nach Wahl am linken oder am rechten Rand des Objekts liegen kann. Der Aufwand für die manuelle Festlegung des Mittelpunkts des Musters ist relativ gering. Er beträgt bei Unterstützung durch ein grafisches Hilfsprogramm pro Bild ca. 10 Sekunden. Dem

Anwender werden alle Bilder in vergrößerter Darstellung gezeigt, und er markiert nur mittels Fadenkreuz den gewünschten Objektpunkt. Auf diese Weise können 100 Trainingsbeispiele eines Objekts in wenigen Minuten erzeugt werden.

Neben den Objektmustern müssen auch die Zurückweisungsmuster vom Anwender festgelegt werden. Dazu werden Bilder von typischen Szenen aufgenommen, allerdings ohne das Objekt bzw. mit abgedecktem Objekt. Jeder mögliche Ausschnitt der festgelegten Größe in diesen Bildern wird als Zurückweisungsmuster verwendet, der Anwender muß also nicht bestimmte Ausschnitte dieser Bilder als Zurückweisungsmuster manuell angeben. Auf diese Weise entstehen mit nur geringem Aufwand viele Zurückweisungsmuster. Ihre Anzahl entspricht in etwa der Anzahl der Pixel eines Bildes, also ca. 10^5 pro Bild. Damit wird die Generierung von bis zu 10^8 Zurückweisungsmustern möglich. Der Speicheraufwand ist relativ gering, da nicht jedes Muster einzeln abgespeichert wird, sondern jedes Bild und die Zurückweisungsmuster sich innerhalb dieser Bilder überdecken[1].

Selbstverständlich sollte darauf geachtet werden, daß andere, typische Objekte der Szene in den Zurückweisungsbildern enthalten sind. Sollten bestimmte Objekte oder Einzelheiten in der Anwendung mit dem gesuchten Objekt verwechselt werden, empfiehlt es sich, mehrere Bilder dieses Objekts in die Zurückweisungsklasse aufzunehmen. Mit dieser so erweiterten Zurückweisungsklasse wird die Baumerzeugung wiederholt.

1.4.5 3D-Positionsbestimmung und Hypothesenfortschreibung

Da als Ergebnis der Objekterkennung zunächst nur die 2D-Position der Objekte im Bild anfällt, für die Greifaufgabe aber die 3D-Position in Fahrzeugkoordinaten notwendig ist, wird die Entfernung durch zusätzliches Wissen rekonstruiert. Im Serviceszenario ist beispielsweise die Standhöhe durch die vorgegebene Tischhöhe bekannt. Mittels Triangulation kann aus dieser Vorinformation und den Bildkoordinaten des Objekts direkt die Objektposition in Fahrzeugkoordinaten bestimmt werden. Neben der Triangulationsmethode kommt auch das Bewegungs-Stereoverfahren (´Stereo by Motion´) zum Einsatz [73], basierend auf der Fähigkeit des Roboterfahrzeugs zur genauen Selbstlokalisierung [28].

Um Fehl-Erkennungen festzustellen, wird die Objektposition während der Annäherung des Fahrzeugs an das Objekt fortgeschrieben und erst nach mehrmaliger Erkennung des Objekts akzeptiert; Fehl-Erkennungen lassen sich damit fast gänzlich ausschließen. Selbst wenn ein anderes Objekt oder eine zufällige Schattierung dem

1 Bei 100 Bildern und jeweils 100000 Pixel ergibt sich ein Speicherbedarf von ca. 10 MByte.

gesuchten Objekt ähneln, wird dies nur bei einer bestimmten Ansicht und damit auf einem kurzen Wegstück des Roboters der Fall sein, während das gesuchte Objekt kontinuierlich erkannt wird.

Neben dem Verifizieren einer Objekterkennung dient die Fortschreibung auch der Filterung der Objektposition [12], die hier durch ein Kalman-Filter unter Verwendung des bekannten, vom Fahrzeug während der Annäherung zurückgelegten Differenzweges realisiert wird. Sie ermöglicht eine absolute Lokalisierungsgenauigkeit von ca. $\pm$ 10 mm bei einem Objektabstand von ca. 1 m (Triangulationsmethode, Fahrzeuggeschwindigkeit ca. 10 cm/s). Die Genauigkeit der Objektlokalisierung muß gewährleisten, daß die beiden Finger des Greifers das Objekt berührungslos umfassen, bevor sich der Greifer schließt. Andernfalls würde das Objekt von den Fingern verschoben, ein erfolgreiches Greifen wäre nicht mehr gewährleistet.

2 Einzelklassifikation

2.1 Vektorklassen

Die in diesem Kapitel betrachtete Einzelklassifikation führt zu der *binären* Entscheidung, ob in einem Knoten des Musterbaums in den rechten oder linken Ast verzweigt werden soll. Zur Klassifikation eines Suchfensters werden die Grauwerte seiner Pixel in Vektoren $\underline{x} \in I\!R^n$ sortiert. Die Dimension n dieser Vektoren ist identisch mit der Anzahl der Pixel im Suchfenster. Anschließend werden diese Vektoren einer Vektorklassifikation unterzogen. Die hier vorgestellten Methoden sind weitgehend unabhängig von der Objekterkennung in Videobildern, sie lassen sich auch in anderem Kontext verwenden.

Ausgangspunkt unserer Überlegungen ist eine *Trainingsmenge* von Vektoren, die eine Teilmenge von Objektvektoren und eine Teilmenge von Zurückweisungsvektoren enthält. Die Objektvektoren stammen von Bildausschnitten, die das Objekt beinhalten; die Zurückweisungsvektoren stammen von Bildausschnitten, die andere Objekte als das gesuchte Objekt zeigen. Die Anzahl N_1 der Objektvektoren ist in unserer Anwendung vergleichsweise gering, typisch sind 100 Vektoren, die Anzahl N_2 der Zurückweisungsvektoren liegt im Bereich $10 \cdot 10^6$ bis $100 \cdot 10^6$.

Diese Trainingsmuster sind *diskrete Stichproben* einer unendlich großen, *kontinuierlichen Mustermenge*. Bei Betrachtung als unendliche Menge treten an Stelle der diskreten Trainingsmuster 2 kontinuierliche Wahrscheinlichkeitsdichteverteilungen; sie sind Funktionen des Vektorraumes. Die Wahrscheinlichkeitsdichteverteilung der Objektmuster wird als $g_1(\underline{x})$ bezeichnet, die Wahrscheinlichkeitsdichteverteilung der Zurückweisungsmuster als $g_2(\underline{x})$. Sie sind definitionsgemäß normiert, d.h.

$$\int_{I\!R^n} g_{1/2}(\underline{x}) \cdot d\underline{x} = 1$$

Die unterschiedliche Häufigkeit wird nun nicht mehr durch die Vektoranzahl, sondern durch die Auftretenswahrscheinlichkeiten p_1 und p_2 beschrieben; es gilt $p_1 + p_2 = 1$ mit $1 \geq p_{1/2} \geq 0$. Die Produkte aus Wahrscheinlichkeit und Wahrscheinlichkeitsdichtefunktion $p_1 \cdot g_1(\underline{x})$ bzw. $p_2 \cdot g_2(\underline{x})$ werden als Häufigkeitsverteilungen bezeichnet. Abb. 2.1zeigt ein Beispiel im 1-dimensionalen Raum.

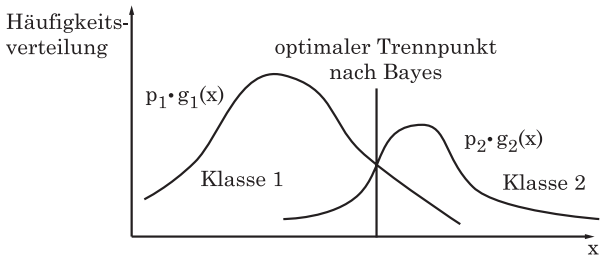

ABBILDUNG 2.1: Häufigkeitsverteilungen zweier Klassen in einem 1-dimensionalen Raum.

Die kontinuierlichen Wahrscheinlichkeitsdichteverteilungen stehen allgemein nicht als analytische Funktionen zur Verfügung. Statt dessen werden sie ausgehend von den Vektormengen durch analytische Funktionen approximiert. Im Kontext dieser Anwendungen haben sich Normalverteilungen durchgesetzt. Der Formalismus zur Berechnung der Normalverteilung ausgehend von der Vektormenge ist in Anhang B dargestellt. Die n-dimensionalen Normalverteilungen werden durch eine Funktion in Abhängigkeit von $\underline{x}$ angegeben:

$$g(\underline{x}) = \frac{1}{\sqrt{(2\pi)^n |det(C)|}} e^{-\frac{1}{2}(\underline{x}-\underline{m})^T C^{-1}(\underline{x}-\underline{m})}$$

Die Matrix C wird als Kovarianzmatrix bezeichnet; sie ist positiv definit. Sie beschreibt, in welchen Richtungen die Verteilung flacher bzw. steiler abfällt. Der Vektor $\underline{m}$ bestimmt den Mittelpunkt der Verteilung.

Bei der Beschreibung einer Vektormenge durch eine Normalverteilung geht Information verloren, da die Normalverteilung in der Regel nur eine Approximation der Vektormenge ist. Des weiteren ist die prinzipielle Form einer Normalverteilung á priori festgelegt und nur ihre Parameter lassen sich an die tatsächliche Verteilung adaptieren. Wird ein Klassifikator anhand der originalen Vektormenge parametriert, lassen sich entsprechend bessere Ergebnisse erwarten als bei einer Parametrierung anhand der approximierten Normalverteilung.

Der Vorteil der Verwendung der Normalverteilung im Vergleich zu einer großen Vektormenge besteht in der geringeren Anzahl von Parametern und dem dadurch bedingt geringeren Aufwand bei der Parametrierung des Klassifikators. Bei unserem Verfahren wird deshalb in einem ersten Schritt aufgrund der Normalverteilung eine Näherungslösung für den Klassifikator berechnet. Diese Lösung wird in einem zweiten Schritt anhand der Vektormenge verbessert.

2.2 Klassifikationsziel

2.2.1 Fehlerminimierung

Zur Parametrierung eines Klassifikators muß zunächst mathematisch das Klassifikationsziel formuliert werden. Gebräuchlich ist der sogenannte *Bayes-Klassifikator*, der anhand der Häufigkeitsverteilung entworfen wird. Sein Ziel ist ein kleinstmöglicher Fehler bei der Klassifikation, so daß an jedem Punkt $\underline{x}_0$ im Vektorraum diejenige Klasse als Ergebnis ausgegeben wird, deren Häufigkeit größer ist, wie in Abb. 2.1 eingezeichnet. Gilt also $p_1 \cdot g_1(\underline{x}_0) > p_2 \cdot g_2(\underline{x}_0)$, wird als Ergebnis ´Klasse 1´ ausgegeben und für $p_1 \cdot g_1(\underline{x}_0) < p_2 \cdot g_2(\underline{x}_0)$ ´Klasse 2´. Die Trennfläche läßt sich offensichtlich aus der Gleichung

$$p_1 \cdot g_1(\underline{x}) = p_2 \cdot g_2(\underline{x})$$

berechnen, sofern die beiden Funktionen analytisch vorliegen. Das Ziel eines kleinstmöglichen Fehlers führt zu folgenden Problemen. Durch mangelnde Information[1] kann die Funktion $p_1 \cdot g_1(\underline{x})$ in jedem Punkt kleiner als die Funktion $p_2 \cdot g_2(\underline{x})$ sein, besonders da die Wahrscheinlichkeit p_2 der Zurückweisungsvektoren sehr viel größer ist als die Wahrscheinlichkeit der Objektvektoren p_1. Abb. 2.2 veranschaulicht diese Möglichkeit im 1-dimensionalen Fall.

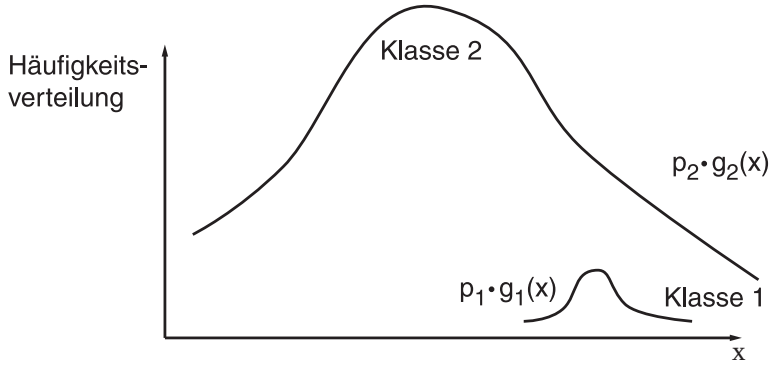

ABBILDUNG 2.2: Schematische Darstellung zweier Klassen mit stark unterschiedlichen Wahrscheinlichkeiten p_1 und p_2

1 z.B. durch Unterabtastung in den ersten Schichten des Musterbaumes, entsprechend Abschnitt 4.2.1.

Die fehlerminimale Klassifikation besteht für den Fall nach Abb. 2.2 offensichtlich darin, als Resultat *unabhängig vom Vektor* $\underline{x}$ das Ergebnis ´Klasse 2´ auszugeben. Durch diese vom Vektor $\underline{x}$ unabhängige Aussage wird die *Unsicherheit* über die Klassenzugehörigkeit nicht vermindert. In vielen Anwendungen mit ähnlich asymmetrischen Vektormengen wird dieses Problem umgangen, indem nicht mit den Häufigkeits- sondern mit den Wahrscheinlichkeitsdichteverteilungen gearbeitet wird, z.B. in [57]. Dieser Klassifikator wird als *Maximum-Likelihood-Klassifikator* bezeichnet. Da die Wahrscheinlichkeitsdichteverteilungen normiert sind, tritt der Fall vollständiger Überdeckung einer Klasse durch eine andere Klasse nicht auf.

2.2.2 Informationsoptimierung

Da das eigentliche Ziel in der *Information über die Klassenzugehörigkeit* des Vektors $\underline{x}$ besteht, wird im Rahmen dieser Arbeit die von Shannon in [62] definierte Information als Zielfunktion verwendet. Sie kann numerisch angegeben werden mit der Einheit „bit“. Die Information über die Klassenzugehörigkeit eines Vektors verringert die Unsicherheit der Klassenzugehörigkeit, die als informationstheoretische *Entropie* H bezeichnet wird und definitionsgemäß positiv ist[1]. Die Entropie bildet die für das gesamte Kapitel zentrale Zielgröße. Ihr durchschnittlicher Wert pro Vektor kann aus den Wahrscheinlichkeiten berechnet werden. Der Formel liegt die Annahme zugrunde, daß die Einzelereignisse voneinander unabhängig sind, K bedeutet die Anzahl der Klassen[2]:

$$H = -\sum_{k=1}^{K} p_k \cdot log_2 p_k \tag{2.1}$$

Der Wert der Information liegt im Intervall $[0; log_2 K]$. Ihr Maximalwert ergibt sich, wenn alle Teilwahrscheinlichkeiten den Wert $p_k = K^{-1}$ haben. In diesem Fall ist die Unsicherheit über die Klasse vor der Klassifikation maximal. Der minimale Wert 0 ergibt sich, wenn stets die selbe Klasse auftritt, in diesem Fall besteht keine Unsicherheit[3].

1 Die hier verwendete informationstechnische Entropie unterscheidet sich von der thermodynamischen Entropie durch einen Vorfaktor.

2 Um die Allgemeinheit der Darstellung nicht unnötig zu verringern, wird nicht von 2 Klassen, sondern von K Klassen ausgegangen. Der Klassenindex ist das „k“. p_k bedeutet also die Wahrscheinlichkeit, daß ein beliebiger Vektor der Klasse k angehört.

3 Der Begriff der Entropie ist sehr abstrakt. Trotzdem läßt sich die Formel für die Entropie interpretieren. Anschaulich trägt ein Ereignis um so mehr Information, je seltener es ist. Dies wird durch den Logarithmus der Wahrscheinlichkeiten ausgedrückt, der betragsmäßig für sehr kleine Wahrscheinlichkeiten sehr groß wird.

Zur anschaulichen Begründung für die Verwendung der Entropie soll kurz die Verbindung zur Signaltheorie aufgezeigt werden. Ein diskretes Signal bestehe aus einer Zeichenfolge mit K unterschiedlichen Zeichen. Im einfachsten Fall eines binären Signals sind dies 2 verschiedene Zeichen, zum Beispiel $\{1, 2\}$. Die beiden Zeichen haben die Wahrscheinlichkeit p_1 und p_2. Die Information dieser Zeichenfolge berechnet sich nach (2.1). Zur Übertragung der Information von einer Signalquelle zu einer Signalsenke werden die Zeichen codiert, beispielsweise als elektrische Spannungswerte $\{U_1 = 0V, U_2 = 5V\}$. Durch die analoge Übertragungsstrecke zwischen Quelle und Senke werden die Spannungswerte verrauscht. Aufgabe der Signalsenke ist es nun, aus den verrauschten Spannungswerten die korrekte Information zu rekonstruieren. Ein einfacher Klassifikator kann den kontinuierlichen Spannungswerten wieder die diskreten Zeichen zuordnen, beispielsweise nach dem Gesetz $(U < 2,5V \rightarrow 1), (U > 2,5V \rightarrow 2)$. Dieses Gesetz ist bereits ein einfacher Klassifikator. Das Problem eines Klassifikators für die Objekterkennung ist exakt das gleiche. Die Signalquelle enthält hier 2 unterschiedliche Zeichen, es sind die Objektklasse und Zurückweisungsklasse. Diese Klassen werden durch ihre optische Abbildung codiert, ähnlich zu den Spannungswerten im obigen Beispiel eine wertekontinuierliche Codierung. Störungen der Übertragungsstrecke sind hier beispielsweise der unterschiedliche Abstand zwischen Kamera und Objekt oder eine wechselnde Beleuchtung. Die Signalsenke korrespondiert zu dem Objekterkennungssystem, das zur Zeichenrekonstruktion einen Klassifikator verwendet. Die Klassifikation ist dann korrekt durchgeführt, wenn die optische Abbildung eines Objekts korrekt einer Klasse zugeordnet wird. Das oben beschriebene Problem aus der Signaltheorie und das hier diskutierte Problem sind offenbar identisch.

Eine sehr anschauliche und detaillierte Diskussion der Entropie und Herleitung der mathematischen Zusammenhänge findet sich in [58] und im Originalartikel von Shannon [62].

Ziel der Klassifikation ist es also, die in (2.1) definierte Information zu extrahieren. Die Extraktion wird in einem Musterbaum in viele Einzelschritte zerlegt, die einzelnen Klassifikationen in den Knoten. Jede einzelne Klassifikation in einem Musterbaumknoten extrahiert einen kleinen Teil der gesamten Information. Jeder dieser Teilschritte soll für sich genommen so effizient wie möglich sein. Ziel an jedem Knoten, d.h. bei einer Einzelklassifikation im Baum, ist also eine maximale Reduktion der Entropie. Die Differenz zwischen der Entropie H vor einer Klassifikation und der Entropie H^* nach einer Klassifikation kann berechnet werden und soll möglichst groß werden. Die Parameter in einem Knoten sind entsprechend so zu bestimmen, daß die Differenz maximal wird. Die Klassifikation in einem Knoten des Baumes bewirkt eine Teilung der Menge der Muster in eine Teilmenge von Mustern, die dem rechten Zweig folgen und eine Teilmenge, die dem linken Zweig folgen. Beide Teilmengen weisen

jeweils eine durchschnittliche Entropie H_1^* und H_2^* auf. Ist die Wahrscheinlichkeit für ein beliebiges Muster, dem linken bzw. rechten Zweig zu folgen $p(1)$ und $p(2)$, dann gilt für die durchschnittliche Entropie H^* nach der Klassifikation:

$$H^* = p(1) \cdot H_1^* + p(2) \cdot H_2^*$$

Für die Berechnung von H_1^* und H_2^* wird die Wahrscheinlichkeit, daß ein beliebiges Muster der Klasse k angehört und dem linken bzw. rechten Zweig folgt als $p_k(1)$ und $p_k(2)$ bezeichnet. Es gilt der Zusammenhang:

$$p(1) = \sum_{k=1}^{K} p_k(1) \qquad p(2) = \sum_{k=1}^{K} p_k(2)$$

Damit erhält man für H_1^* und H_2^*:

$$H_1^* = -\sum_{k=1}^{K} \frac{p_k(1)}{p(1)} \cdot log_2 \frac{p_k(1)}{p(1)} \qquad H_2^* = -\sum_{k=1}^{K} \frac{p_k(2)}{p(2)} \cdot log_2 \frac{p_k(2)}{p(2)}$$

Es ergibt sich weiter die gesamte Entropie nach der Klassifikation

$$H^* = -p(1)\sum_{k=1}^{K} \frac{p_k(1)}{p(1)} \cdot log_2 \frac{p_k(1)}{p(1)} - p(2)\sum_{k=1}^{K} \frac{p_k(2)}{p(2)} \cdot log_2 \frac{p_k(2)}{p(2)}$$

oder kürzer:

$$H^* = -\sum_{k=1}^{K} p_k(1) \cdot log_2 \frac{p_k(1)}{p(1)} - \sum_{k=1}^{K} p_k(2) \cdot log_2 \frac{p_k(2)}{p(2)} \qquad (2.2)$$

Aufgabe des Klassifikators ist die Minimierung dieses Ausdrucks. Diese Minimierung ist im Vergleich zu einer Fehlerminimierung ein etwas abstrakteres Ziel. Durch die Formulierung der Information als Klassifikationsziel besteht das primäre Ziel nicht mehr darin, daß etwa die Vektoren der Klasse 1 in den linken Knoten und die Vektoren der Klasse 2 in den rechten Knoten sortiert werden. Trotzdem gibt es einen Zusammenhang zwischen Informationsoptimierung und Fehlerminimierung. Die extrahierte Information ist dann am größten, wenn jeder der Folgeknoten nur eine der beiden Klassen aufnimmt, also auch der Fehler zu 0 wird. Allerdings ist der Begriff der

Information allgemeiner. Er ist auch dann definiert, wenn die Anzahl der Klassen und die Anzahl der Folgeknoten nicht übereinstimmen. Der Begriff des Fehlers versagt in solchen Fällen, beispielsweise wenn 3 Klassen und 2 Folgeknoten vorhanden sind.

2.3 Optimale Klassifikatoren

In Abschnitt 2.1 wurden zwei unterschiedliche Repräsentationsformen der Muster als Vektormenge und als kontinuierliche Häufigkeitsverteilung diskutiert, außerdem in Abschnitt 2.2 das Ziel der Klassifikation. Aus beiden muß nun eine geeignete Methode zur Klassifikation abgeleitet werden. Eine solche Klassifikation bestimmt für jeden beliebigen Vektor $\underline{x}$ ein binäres Ergebnis $e \in \{1, 2\}$. Je nach Ergebnis wird in dem Musterbaum verzweigt, für $e = 1$ in den linken Folgeknoten, für $e = 2$ in den rechten Folgeknoten[1].

Diese Klassifikation teilt also den gesamten Vektorraum in zwei Teilräume. Beide Bereiche werden durch eine Hyperfläche voneinander getrennt. Um uns dem Problem eines geeigneten Klassifikators zu nähern, wird zunächst die optimale Trennfläche analysiert. Da die resultierenden Trennflächen für die technische Realisierung Probleme verursachen, wird anschließend als einfachere Form der lineare Klassifikator diskutiert. Die optimale Trennfläche ist eine ausschließlich von der Trainingsstichprobe abhängige Hyperfläche, über deren Struktur zunächst nichts bekannt ist und die nur in wenigen Fällen analytisch darstellbar bzw. berechenbar ist[2]. Für die Wahrscheinlichkeiten gilt die Nomenklatur nach Abschnitt 2.2, relevant sind $p_1(1)$, $p_2(1)$, $p_1(2)$ und $p_2(2)$. Die Wahrscheinlichkeitsdichtefunktionen, multipliziert mit der jeweiligen Klassenwahrscheinlichkeit werden als Häufigkeitsverteilungen $v_k(\underline{x}) = p_k \cdot g_k(\underline{x})$ bezeichnet.

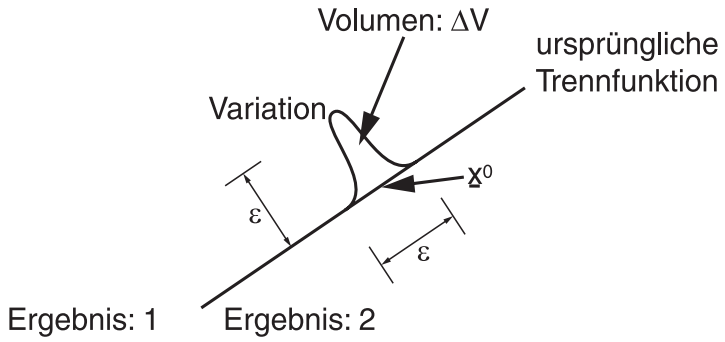

ABBILDUNG 2.3: Lokale Variation der Trennfläche.

1 Im folgenden wird der linke Folgeknoten als Knoten 1, der rechte Folgeknoten als Knoten 2 bezeichnet.

2 Sind die Musterklassen trennbar, verläuft die optimale Trennfläche zwischen den beiden Verteilungsbereichen, ihr genauer Verlauf ist nicht definiert. Interessant ist hier nur der Fall, daß die Klassen aus Mangel an Information nicht trennbar sind und sich überlappen, wie das in den ersten Schichten des Musterbaums aufgrund der geringen Musterauflösung der Fall ist (Abschnitt 4.2.1).

Entsprechend dem Variationsprinzip wird zunächst die optimale Trennfläche betrachtet. Sie wird an der Koordinate $\underline{x}^0$ variiert, wodurch sich die beiden Teilräume beiderseits der Trennfläche ändern. Für die weiteren Überlegungen ist die genaue Form der Variation irrelevant, trotzdem wird eine Einschränkung vorgenommen, um die Darstellung zu vereinfachen. Um innerhalb des Volumens ΔV Konstanz der Wahrscheinlichkeitsdichtefunktionen annehmen zu können, habe die Variation eine Ausdehnung ε. ε sei unendlich klein, aber für die weiteren Überlegungen konstant. Wir verzichten deshalb auf eine Kennzeichnung mit dem Differential-Operator d. Die Variation verändere außerdem die ursprüngliche Trennfunktion so, daß die neue Trennfunktion stetig sei.

Für die weitere Rechnung wird die Variation mit einer Hilfsvariablen e multipliziert. Das Volumen beider Teilräume wird entsprechend Abb. 2.3 um jeweils $e \cdot \Delta V$ größer bzw. kleiner. Durch die Variation ergeben sich nun Änderungen für die Einzelwahrscheinlichkeiten, für Klasse 1 ist dies:

$$p_1^*(1) = p_1(1) + v_1(\underline{x}^0) \cdot \Delta V \cdot e \qquad p_1^*(2) = p_1(2) - v_1(\underline{x}^0) \cdot \Delta V \cdot e$$

Für Klasse 2 ergibt sich

$$p_2^*(1) = p_2(1) + v_2(\underline{x}^0) \cdot \Delta V \cdot e \qquad p_2^*(2) = p_2(2) - v_2(\underline{x}^0) \cdot \Delta V \cdot e\,.$$

Damit erhält man als Ableitung der Wahrscheinlichkeiten nach e:

$$\frac{d}{de}p_1(1) = v_1(\underline{x}^0) \cdot \Delta V \qquad \frac{d}{de}p_1(2) = -v_1(\underline{x}^0) \cdot \Delta V$$

$$\frac{d}{de}p_2(1) = v_2(\underline{x}^0) \cdot \Delta V \qquad \frac{d}{de}p_2(2) = -v_2(\underline{x}^0) \cdot \Delta V$$

Weiterhin gilt gemäß Abschnitt 2.2:

$$\begin{aligned} H^* = \; & -p_1(1)log_2\frac{p_1(1)}{p_1(1)+p_2(1)} - p_1(2)log_2\frac{p_1(2)}{p_1(2)+p_2(2)} \\ & -p_2(1)log_2\frac{p_2(1)}{p_1(1)+p_2(1)} - p_2(2)log_2\frac{p_2(2)}{p_1(2)+p_2(2)} \end{aligned}$$

Damit erhalten wir als Ableitungen nach den Einzelwahrscheinlichkeiten:

$$\begin{aligned}
\frac{dH^*}{dp_1(1)} &= \frac{p_1(1)}{p_1(1)+p_2(1)} - log_2 \frac{p_1(1)}{p_1(1)+p_2(1)} \\
\frac{dH^*}{dp_1(2)} &= \frac{p_1(2)}{p_1(2)+p_2(2)} - log_2 \frac{p_1(2)}{p_1(2)+p_2(2)} \\
\frac{dH^*}{dp_2(1)} &= \frac{p_2(1)}{p_1(1)+p_2(1)} - log_2 \frac{p_2(1)}{p_1(1)+p_2(1)} \\
\frac{dH^*}{dp_2(2)} &= \frac{p_2(2)}{p_1(2)+p_2(2)} - log_2 \frac{p_2(2)}{p_1(2)+p_2(2)}
\end{aligned} \tag{2.3}$$

Daraus berechnet sich die Ableitung von H^* nach e :

$$\frac{dH^*}{de} = \frac{dH^*}{dp_1(1)} \cdot \frac{d}{de} p_1(1) + \frac{dH^*}{dp_2(1)} \cdot \frac{d}{de} p_2(1) + \frac{dH^*}{dp_1(2)} \cdot \frac{d}{de} p_1(2) + \frac{dH^*}{dp_2(2)} \cdot \frac{d}{de} p_2(2)$$

$$\frac{dH^*}{de} = v_1(\underline{x}^0) \cdot \left(\frac{d}{dp_1(1)} H^* - \frac{d}{dp_1(2)} H^* \right) \cdot \Delta V + v_2(\underline{x}^0) \cdot \left(\frac{d}{dp_2(1)} H^* - \frac{d}{dp_2(2)} H^* \right) \cdot \Delta V$$

Ist die Trennfunktion vor der Variation bereits optimal, ist die Größe H^* also in ihrem Minimum, führt jede Variation zu einer Vergrößerung. Dies gilt für eine Variation in beide Richtungen, also für ein positives und ein negatives e . Dies bedeutet aber, daß die Ableitung $\left. \frac{dH^*}{de} \right|_{e=0}$ im Optimum 0 sein muß; es gilt nach Einsetzen von (2.3):

$$\begin{aligned}
&v_1(\underline{x}^0) \cdot \left(\frac{p_1(1)}{p_1(1)+p_2(1)} - log_2 \frac{p_1(1)}{p_1(1)+p_2(1)} - \frac{p_1(2)}{p_1(2)+p_2(2)} + log_2 \frac{p_1(2)}{p_1(2)+p_2(2)} \right) \\
&+ v_2(\underline{x}^0) \cdot \left(\frac{p_2(1)}{p_1(1)+p_2(1)} - log_2 \frac{p_2(1)}{p_1(1)+p_2(1)} - \frac{p_2(2)}{p_1(2)+p_2(2)} + log_2 \frac{p_2(2)}{p_1(2)+p_2(2)} \right) = 0
\end{aligned}$$

Da die Klammerausdrücke für die gesamte Trennfläche Gültigkeit haben, kann man sie mit c_1 und c_2 substituieren und erhält als Ergebnis für jeden Punkt $\underline{x}$ der Trennfläche:

$$v_1(\underline{x}) \cdot c_1 + v_2(\underline{x}) \cdot c_2 = 0$$

oder kürzer:

$$v_1(\underline{x}) / v_2(\underline{x}) = const. \tag{2.4}$$

Diese Gleichung wird im nächsten Abschnitt für den Fall der Normalverteilung diskutiert.

2.4 Quadratische Klassifikatoren

2.4.1 Analytische Näherungslösung

Aus (2.4) kann für zwei Normalverteilungen analytisch eine Trennfläche berechnet werden. Es ergibt sich nach [57] eine quadratische Form:

$$\underline{x}^T A \underline{x} + \underline{w}^T \underline{x} + a = 0 \quad \text{mit} \quad A \in I\!R^{nxn} \quad \underline{w} \in I\!R^n \quad a \in I\!R.$$

Der Vektor $\underline{w}$ und die Matrix A ergeben sich aus den Parametern der Normalverteilungen zu:

$$\underline{w} = 2 \cdot \underline{m}_1^T \cdot C_1^{-1} - 2 \cdot \underline{m}_2^T \cdot C_2^{-1} \quad \text{und} \quad A = C_2^{-1} - C_1^{-1} \tag{2.5}$$

Aus (2.4) läßt sich keine Bedingung für den Parameter a herleiten. Er kann im Sinne einer Näherungslösung beispielsweise so festgelegt werden, daß die Trennfläche die Mitte $\frac{\underline{m}_1 + \underline{m}_2}{2}$ zwischen den Normalverteilungen beinhaltet[1].

Die Klassifikation für einen zunächst unbekannten Vektor $\underline{x}_0$ besteht nun in der Auswertung des Ausdrucks

$$sgn(\underline{x}_0{}^T A \underline{x}_0 + \underline{w}^T \underline{x}_0 + a)$$

Bei negativem[2] Ergebnis wird in den Folgeknoten 1 verzweigt, bei positivem Ergebnis in den Folgeknoten[3] 2. Obwohl der quadratische Klassifikator optimal für die Klassifikation normalverteilter Klassen ist, ist er sicher nicht optimal für die tatsächlich vorhandenen Verteilungen, die durch die Trainings-Vektormengen gegeben sind.

1 Da die Parametrierung ohnehin nur auf der Approximation mittels Normalverteilung beruht, ist diese Methode zulässig. Die anschließende Optimierung erfolgt ohne eine solche Näherung.

2 Der Funktionswert $sgn(0) = 0$ ist für die Praxis bedeutungslos.

3 Der Wert der extrahierten Information ändert sich selbstverständlich nicht, wenn die Knoten vertauscht werden. Der Grund ist, daß die Knotenbezeichnungen ´links´, ´rechts´ bzw. ´Knoten 1´ und ´Knoten 2´ keine weitere Bedeutung haben.

Eine analytische Berechnung eines optimalen Klassifikators anhand der Vektormengen erscheint ausgeschlossen. Statt dessen wird eine numerische Optimierung durchgeführt, die allerdings zur Initialisierung eine analytische Näherungslösung voraussetzt. Dazu werden die Vektormengen zunächst nach Anhang B durch Normalverteilungen approximiert. Anschließend werden aus den Normalverteilungen mittels (2.5) Näherungslösungen berechnet. Mittels dieser Näherungslösungen wird die numerische Optimierung initialisiert. Abb. 2.4 gibt einen Überblick über den Zusammenhang.

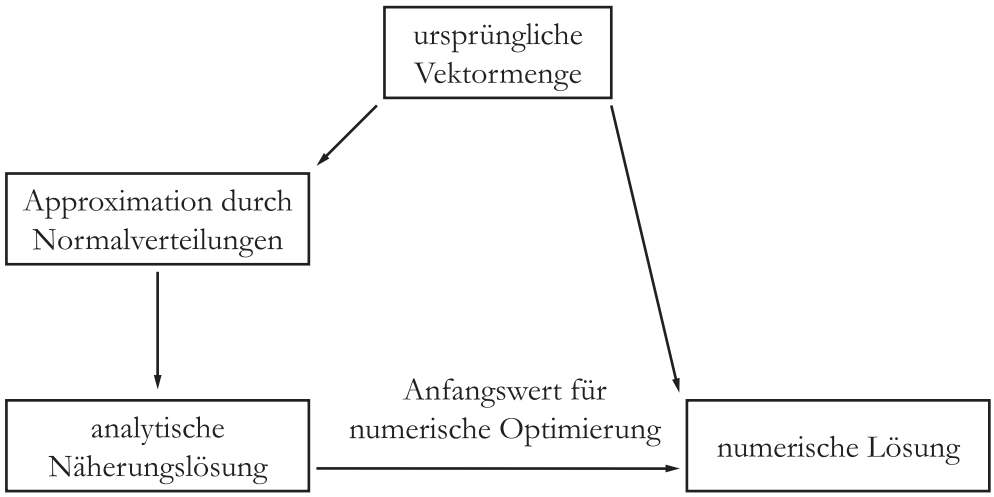

ABBILDUNG 2.4: Berechnung einer Näherungslösung für die numerische Optimierung nach Abschnitt 2.4.3.

Die Diskussion der numerischen Optimierung wird in 2 Abschnitte aufgeteilt. In Abschnitt 2.4.2 werden die Entropie und ihre Ableitung nach den Klassifikatorparametern als analytische Funktionen der Klassifikatorparameter dargestellt. In Abschnitt 2.4.3 wird mittels dieser Funktionen die Entropie minimiert. Im Resultat erhalten wir eine Struktur des Klassifikators aus der Überlegung anhand von Normalverteilungen, während die Parameter in Hinblick auf das Klassifikationsziel anhand der Vektormengen eingestellt werden.

Die numerische Optimierung führt nicht zwangsweise zu einer global optimalen Lösung, sondern nur zu einem lokalen Optimum.

2.4.2 Entropie als Funktion der Klassifikatorparameter

Zur Optimierung der Parameter eines Klassifikators muß ein numerischer Zusammenhang zwischen den Parametern des Klassifikators und dem Ziel der Klassifikation, der Minimierung der Entropie, hergestellt werden. Dazu wird die Entropie als Funktion

der Parameter formuliert. Da sich die Entropie zunächst aus den Einzelwahrscheinlichkeiten nach (2.1) berechnet, müssen diese als Funktionen der Klassifikator-Parameter angegeben werden. Dazu berechnet man für jede Klasse:

$$\Delta_k = \sum_{n=1}^{N_k} sgn(\underline{x}_{kn}^T A \underline{x}_{kn} + \underline{w}^T \underline{x}_{kn} + a)$$

Die Bezeichnung $\underline{x}_{kn}$ bedeutet den n-ten Vektor der Klasse k. Die Anzahl der Vektoren beträgt für jede Klasse N_k. Es ergibt sich mit der Gesamtzahl aller Vektoren

$$N = \sum_{k=1}^{K} N_k$$

für die Wahrscheinlichkeiten:

$$p_k(1) = \frac{1}{2} \cdot \frac{N_k - \Delta_k}{N} \qquad p_k(2) = \frac{1}{2} \cdot \frac{N_k + \Delta_k}{N} \tag{2.6}$$

Damit läßt sich die Entropie entsprechend (2.2) in Abhängigkeit der Klassifikatorparameter angeben. Da für die Minimierung nicht der absolute Wert zählt, wird zur Vereinfachung der Rechnung H^* ersetzt. Es gilt die Beziehung $H^{**} = 2 \cdot N \cdot H^* \cdot ln2$. Wir erhalten durch Einsetzen von (2.6) in (2.2) mit $\Delta = \sum_{k=1}^{K} \Delta_k$:

$$H^{**} = -\sum_{k=1}^{K} (N_k - \Delta_k) ln \frac{N_k - \Delta_k}{N - \Delta} + (N_k + \Delta_k) ln \frac{N_k + \Delta_k}{N + \Delta}$$

Für das numerische Optimierungsverfahren wird neben dem Funktionswert auch seine Ableitung berechnet. Es ergibt sich als partielle Ableitung nach den Δ_k der kompakte Ausdruck:

$$\frac{\partial H^{**}}{\partial \Delta_k} = ln \frac{(N_k - \Delta_k)(N + \Delta)}{(N_k + \Delta_k)(N - \Delta)} \tag{2.7}$$

Ebenfalls zu berechnen sind die Ableitungen der Größen Δ_k nach den Parametern a, $\underline{w}$ und A.

Die Ableitung nach den Parametern $\underline{w}$ und a ergibt sich formal zu

$$\frac{dH^{**}}{d\underline{w}} = \sum_{k=1}^{K} \frac{\partial H^{**}}{\partial \Delta_k} \cdot \frac{d\Delta_k}{d\underline{w}} \quad \text{und} \quad \frac{dH^{**}}{da} = \sum_{k=1}^{K} \frac{\partial H^{**}}{\partial \Delta_k} \cdot \frac{d\Delta_k}{da}.$$

Dieser Ausdruck ist zunächst nicht definiert, da die Ableitung der *sgn* -Funktion nicht möglich ist. Aus diesem Grund wird als Näherung der *sgn* -Funktion die *atan* -Funktion verwendet:

$$\Delta_k^* = \frac{2}{\pi} \sum_{n=1}^{N_k} atan(\eta \cdot (\underline{x}_{kn}^T A \underline{x}_{kn} + \underline{w}^T \underline{x}_{kn} + a)), \qquad \eta > 0 \tag{2.8}$$

Der Parameter η gibt an, wie steil die Funktion durch den Ursprung läuft, die Normierung $2/\pi$ wird zum Anpassen des Wertebereiches der *atan* -Funktion an die *sgn* -Funktion benötigt. Die Ableitung dieser Funktion nach den Klassifikatorparametern $\underline{w}$ und a berechnet sich damit zu

$$\frac{d\Delta_k^*}{da} = \frac{2}{\pi} \sum_{n=1}^{N_k} \frac{\eta}{1 + (\eta \cdot (\underline{w}^T \underline{x}_{kn} + a))^2}$$

$$\frac{d\Delta_k^*}{d\underline{w}} = \frac{2}{\pi} \sum_{n=1}^{N_k} \frac{\eta}{1 + (\eta \cdot (\underline{w}^T \underline{x}_{kn} + a))^2} \underline{x}_{kn}$$

Neben a und $\underline{w}$ ist auch die Matrix A optimal festzulegen. Entsprechend muß neben den bereits bekannten Ableitungen $\frac{d\Delta_k^*}{d\underline{w}}$ und $\frac{d\Delta_k^*}{da}$ auch noch Δ_k nach A abgeleitet werden, symbolisch zu schreiben als $\frac{d\Delta_k^*}{dA}$. Diese Operation, die Ableitung eines Skalars nach einer Matrix, ist nicht definiert. Statt dessen müssen die Δ_k^* nach den Einträgen der Matrix A abgeleitet werden. Um dies zu vereinfachen, wird ohne Beschränkung der Allgemeinheit der quadratische Term anders als üblich mittels eines Skalarproduktes angegeben, also genau so wie der lineare Term. Dazu wird neben dem Vektor $\underline{x}^T = \left[x_1 \ \ldots \ x_n\right]$ der Vektor

$$\underline{z}^T = \left[x_1^2 \ x_1x_2 \ x_1x_3 \ \ldots \ x_1x_n \mid x_2^2 \ x_2x_3 \ x_2x_3 \ \ldots \ x_2x_n \mid \ldots \mid x_n^2\right]$$

definiert und analog zu dem Vektor $\underline{w}$ der zusätzliche Gewichtsvektor $\underline{c}$ eingeführt, es gilt $\underline{x}^T A \underline{x} = \underline{c}^T \underline{z}$. Der Vektor $\underline{c}$ gewichtet den Vektor $\underline{z}$ und hat deshalb dieselbe Dimension. Damit ergibt sich für Δ_k die neue Schreibweise:

$$\Delta_k = \sum_{n=1}^{N_k} sgn(\underline{c}^T \underline{z}_{kn} + \underline{w}^T \underline{x}_{kn} + a) \tag{2.9}$$

Der neueingeführte Vektor $\underline{c}$ wird wie der Vektor $\underline{w}$ behandelt, es ergibt sich damit

$$\frac{d\Delta_k^*}{d\underline{c}} = \frac{\eta}{1 + (\eta \cdot (\underline{c}^T \underline{z}_{kn} + \underline{w}^T \underline{x}_{kn} + a))^2} \underline{c}_{kn}$$

Damit erhalten wir als Ableitung der Entropie nach den Parametern des Klassifikators

$$\frac{dH^{**}}{da} = \frac{2}{\pi} \sum_{n=1}^{N_k} ln \frac{(N_k - \Delta_k)(N + \Delta)}{(N_k + \Delta_k)(N - \Delta)} \frac{\eta}{1 + (\eta \cdot (\underline{c}^T \underline{z}_{kn} + \underline{w}^T \underline{x}_{kn} + a))^2} \tag{2.10}$$

$$\frac{dH^{**}}{d\underline{w}} = \frac{2}{\pi} \sum_{n=1}^{N_k} ln \frac{(N_k - \Delta_k)(N + \Delta)}{(N_k + \Delta_k)(N - \Delta)} \frac{\eta}{1 + (\eta \cdot (\underline{c}^T \underline{z}_{kn} + \underline{w}^T \underline{x}_{kn} + a))^2} \underline{x}_{kn} \tag{2.11}$$

$$\frac{dH^{**}}{d\underline{c}} = \frac{2}{\pi} \sum_{n=1}^{N_k} ln \frac{(N_k - \Delta_k)(N + \Delta)}{(N_k + \Delta_k)(N - \Delta)} \frac{\eta}{1 + (\eta \cdot (\underline{c}^T \underline{z}_{kn} + \underline{w}^T \underline{x}_{kn} + a))^2} \underline{z}_{kn} \tag{2.12}$$

2.4.3 Numerische Optimierung quadratischer Klassifikatoren

Nachdem ein Zusammenhang zwischen der Entropie und den Klassifikatorparametern hergestellt wurde und auf Basis von Normalverteilungen eine Näherungslösung für die Parameter gefunden wurde, kann nun die numerische Optimierung der Parameter durchgeführt werden. Da die Entropie als Funktion der gesuchten Parameter vorliegt und auch die Ableitungen bekannt sind, handelt es sich um ein *statisches Optimierungsproblem*[1]. Da nicht jede einzelne Optimierung eines Knotens aufgrund der großen Anzahl der Knoten manuell überprüft und initialisiert werden kann, wird ein *vollautomatisches Optimierungsverfahren* benötigt. Dieses muß in jedem Fall zu einer Lösung kommen und darf nicht etwa in einen Grenzzyklus münden. Wichtiger als eine hohe

1 Der Suchalgorithmus für das globale Optimum ist nach [54] NP-vollständig.

Konvergenzgeschwindigkeit ist also die *Stabilität des Verfahrens*. Das hier verwendete Gradienten-Abstiegsverfahren wird deshalb so gestaltet, daß in jedem Fall eine Verbesserung der Lösung erzwungen wird[1].

Ausgangspunkt ist eine Initialisierung entsprechend (2.5), die dort gefundene Matrix A wird in den Vektor $\underline{c}$ umgerechnet. Damit erhalten wir als Ausgangspunkt der numerischen Optimierung die als $\underline{c}^0$, $\underline{w}^0$ und a^0 bezeichneten Parameter. Die Ableitung berechnet sich an der betreffenden Stelle nach den Gleichungen (2.10) bis (2.12). Aus diesen Ableitungen berechnen sich neue Parameter mittels der Formeln:

$$\underline{c}^{\nu+1} = \underline{c}^{\nu} + \sigma^{\nu} \cdot \frac{dH^{**}}{d\underline{c}} \qquad \underline{w}^{\nu+1} = \underline{w}^{\nu} + \sigma^{\nu} \cdot \frac{dH^{**}}{d\underline{w}} \qquad a^{\nu+1} = a^{\nu} + \sigma^{\nu} \cdot \frac{dH^{**}}{da}$$

σ^{ν} bezeichnet die Schrittweite. Ziel ist dabei, daß der Funktionswert H^{**} an der neuen Stelle kleiner ist als an der alten. Ist nun die Schrittweite zu groß, kann das Minimum übersprungen werden und der Funktionswert ansteigen. Dies wird überprüft durch Berechnen der beiden Funktionswerte. Ist der neue Funktionswert größer als der alte, wird die Iteration rückgängig gemacht und die Schrittweite verkleinert:

$$\sigma^{\nu+1} = \frac{1}{\alpha} \cdot \sigma^{\nu} \text{ mit } \alpha > 1 .$$

Diese Anpassung der Schrittweite wird solange wiederholt, bis sich tatsächlich der Funktionswert H^{**} verkleinert.

Ist der Funktionswert an der neuen Stelle $\nu + 1$ kleiner als der Funktionswert an der alten Stelle, wird der Schritt nicht rückgängig gemacht. Von der neuen Stelle aus wird die nächste Iteration vorbereitet. Die Schrittweite für die nächste Iteration wird etwas größer gewählt zu

$$\sigma^{\nu+1} = \alpha \cdot \sigma^{\nu} .$$

Diese Regelung der Schrittweite ist sehr effektiv, sie stellt die Schrittweite schnell und *vollkommen selbständig* auf das erforderliche Maß ein. Die Anfangsschrittweite σ^0 kann in weiten Grenzen beliebig gewählt werden. Ein zu großer oder zu kleiner Wert führt lediglich zu zusätzlichen Rechenoperationen, bis die Schrittweite passend eingestellt

1 Selbstverständlich können auch andere, komplexere Suchmethoden verwendet werden, die eine Verbesserung der Lösung erzwingen.

ist, eine Festlegung auf ein Zehntel des Betrages des Parametervektors $\left|\left[\underline{c}^T\ \underline{w}^T\ a\right]\right|$ ergab ein schnelles Einschwingen. Die Wahl des Parameters $\alpha > 1$ ist unkritisch, ein Wert von *1,1* ergibt in der Praxis gute Ergebnisse.

Dieser Algorithmus stellt sicher, daß die Optimierung nicht in einen Grenzzyklus mündet, da zwangsweise eine Verkleinerung der Entropie stattfindet.

Ein Kriterium für den Abbruch der iterativen Minimumsuche ergibt sich aus dem Pixelrauschen der Kamera. Ein beliebiger Vektor $\underline{x}_{kn}$ und damit der daraus abgeleitete Vektor $\underline{z}_{kn}$ sind mit dem Pixelrauschen der Kamera behaftet. Dieses Rauschen geht in den Ausdruck $\underline{c}^T\underline{z}_{kn} + \underline{w}^T\underline{x}_{kn} + a$ ein. Eine sinnvolle Grenze für die Genauigkeit des Parametervektors $\left[\underline{c}^T\ \underline{w}^T\ a\right]$ besteht nun dadurch, daß der Ausdruck $\underline{c}^T\underline{z}_{kn} + \underline{w}^T\underline{x}_{kn} + a$ durch die Ungenauigkeit des Parametervektors nicht mehr beeinflußt wird als durch das Rauschen der Vektoren $\left[\underline{z}^T\ \underline{x}^T\ 1\right]$. Für den Einfluß des Rauschens und der Ungenauigkeit des Parametervektors gilt:

$$\begin{bmatrix}\underline{z}\\ \underline{x}\\ 1\end{bmatrix}^T_{Rauschen} \cdot \begin{bmatrix}\underline{c}\\ \underline{w}\\ a\end{bmatrix} \text{ und } \begin{bmatrix}\underline{c}\\ \underline{w}\\ a\end{bmatrix}^T_{Ungenauigkeit} \cdot \begin{bmatrix}\underline{z}\\ \underline{x}\\ 1\end{bmatrix}$$

Konservativ[1] abgeschätzt muß die Ungleichung

$$\left|\begin{bmatrix}\underline{c}\\ \underline{w}\\ a\end{bmatrix}_{Ungenauigkeit}\right| < \frac{\left|\left[\underline{z}^T\ \underline{x}^T\ 1\right]_{Rauschen}\right|}{\left|\left[\underline{z}^T\ \underline{x}^T\ 1\right]\right|} \cdot \left|\begin{bmatrix}\underline{c}\\ \underline{w}\\ a\end{bmatrix}\right|$$

eingehalten werden, und zwar für alle in der Trainingsstichprobe enthaltenen Vektoren $\left[\underline{z}^T\ \underline{x}^T\ 1\right]^T$. Daraus kann direkt ein Kriterium für das Terminieren der Minimumsuche angegeben werden. Die Schrittweite σ wird um so kleiner, je besser man sich dem Minimum nähert; sie kann in der Nähe des Minimums als Maß für den Abstand verwendet werden.

1 Die Abschätzung nimmt für die Vektoren $\underline{z}$ und $\underline{x}$ unabhängiges Rauschen an und führt deshalb zu einem zu restriktiven Ergebnis.

Damit ergibt sich als Abbruchkriterium der Ausdruck:

$$\sigma^{Schwelle} = \frac{\left|\left[\underline{z}^T\ \underline{x}^T\ 1\right]_{Rauschen}\right|}{\left|\left[\underline{z}^T\ \underline{x}^T\ 1\right]\right|} \cdot \left|\begin{bmatrix}\underline{c}\\ \underline{w}\\ a\end{bmatrix}\right|$$

Es fällt auf, daß nur der relative Wert[1] des Vektors $\left[\underline{c}^T\ \underline{w}^T\ a\right]^T$, nicht aber sein Absolutwert in die Rechnung eingehen.

Wird der Vektor mit einer beliebigen, positiven Konstante multipliziert, ändert sich der Ausdruck $sgn(\underline{c}^T\underline{z} + \underline{w}^T\underline{x} + a)$ nicht. Folglich kann der Parametervektor im Verlauf der Optimierung betragsmäßig sehr groß oder sehr klein werden. Deshalb wird er nach jeder Iteration auf den Betrag 1 normiert. Diese Maßnahme dient ausschließlich einer besseren, numerischen Konditionierung der Zahlenwerte, um große Exponenten und im Extremfall einen Zahlenüberlauf zu verhindern.

Bei den bisherigen Betrachtungen wurde die Festlegung des Parameters η ausgeklammert. Dieser Parameter wurde in (2.8) eingeführt, um die nicht stetige *sgn* -Funktion mit der stetig differenzierbaren *atan* -Funktion zu approximieren:

$$sgn\left(\left[\underline{c}^T\ \underline{w}^T\ a\right]\begin{bmatrix}\underline{z}\\ \underline{x}\\ 1\end{bmatrix}\right) \rightarrow \frac{2}{\pi} \cdot atan\left(\eta \cdot \left[\underline{c}^T\ \underline{w}^T\ a\right]\begin{bmatrix}\underline{z}\\ \underline{x}\\ 1\end{bmatrix}\right)$$

Liegt ein bestimmter Vektor $\underline{x}$ genau auf der Trennfläche, ergibt sich für diesen Vektor $\eta \cdot \left[\underline{c}^T\ \underline{w}^T\ a\right]\left[\underline{z}^T\ \underline{x}^T\ 1\right]^T = 0$. Eine Änderung des Gewichtsvektors soll nun das Argument nur im Bereich [-1,1] verändern, damit der Vektor nicht aus dem mittleren Teil der *atan* -Funktion in den sehr flachen Bereich verschoben wird. Nach einem Optimierungsschritt

$$\left[\underline{c}^T\ \underline{w}^T\ a\right] \rightarrow \left[\underline{c}^T\ \underline{w}^T\ a\right] + \left[\Delta\underline{c}^T\ \Delta\underline{w}^T\ \Delta a\right]$$

soll also gelten:

1 Für die heute handelsüblichen Kameras mit einem Signal/Rauschabstand von ca. 45 db und die übliche Skalierung der Pixelwerte auf [0...256] ergibt sich: $\left|\left[\underline{c}^T\ \underline{w}^T\ a\right]_{Ungenauigkeit}\right| < 10^{-2} \cdot \left|\left[\underline{c}^T\ \underline{w}^T\ a\right]\right|$. Der Wert ist unabhängig von der Dimension des Vektors $\left[\underline{z}^T\ \underline{x}^T\ 1\right]$.

$$\left| \eta \cdot \left[\Delta \underline{c}^T \; \Delta \underline{w}^T \; \Delta a \right] \begin{bmatrix} \underline{z} \\ \underline{x} \\ 1 \end{bmatrix} \right| < 1$$

Daraus ergibt sich als Bedingung für den Parameter η :

$$\eta < \left| \frac{1}{\left[\Delta \underline{c}^T \; \Delta \underline{w}^T \; \Delta a \right] \begin{bmatrix} \underline{z} \\ \underline{x} \\ 1 \end{bmatrix}} \right|$$

Der Parameter wird im Anschluß an eine Iteration jeweils neu berechnet und bleibt für eine Iteration konstant. Die Ungleichung wird für alle Vektoren $\underline{x}_{kn}$ ausgewertet und der restriktivste Wert gewählt. Im Verlauf einer Optimierung wird die Veränderung des Parametervektors immer kleiner, dementsprechend wächst der Parameter η . Dies entspricht dem Ziel, der *sgn*- Funktion näher zu kommen. Gleichzeitig wird die Gefahr für eine Konvergenz in einem lokalen Minimum verringert. Das Anwachsen von η wird nach Abschnitt 4.1.4 beschränkt.

2.4.4 Probleme quadratischer Klassifikatoren

2.4.4.1 Technische Umsetzung

Der quadratische Klassifikator nach (2.9) enthält im Fall eines n-dimensionalen Vektorraums $\frac{n^2 + 3n + 2}{2}$ Parameter. Bei einem Bildausschnitt der Größe 31x31 ergibt sich n zu 961 und damit die Anzahl der Parameter zu 463203. Bei einer Speicherung im Fließkommaformat benötigt man dazu 3,7 MByte Speicher. Da dies der Speicher für nur einen einzigen Knoten ist, beträgt der Speicheraufwand für den gesamten Baum ein Vielfaches.

Ein weiteres Problem besteht in der notwendigen Rechenleistung. Um eine quadratische Klassifikation durchzuführen, sind $n^2 + 2n$ Multiplikationen und $\frac{n^2 + 3n + 1}{2}$ Additionen notwendig. Für große n kann zur Vereinfachung von n^2 Multiplikationen ausgegangen werden; dies sind für $n = 961$ ca. 10^6 Multiplikationen, die selbst auf einem schnellen Prozessor eine Zeit von $> 1ms$ benötigen. Dies ist wiederum die Zeit für einen einzelnen Knoten, die Auswertung für den gesamten Baum beträgt auch hier ein Vielfaches.

2.4.4.2 Invarianz gegen Schwankungen der Beleuchtung

Für viele Anwendungen ist eine Klassifikation wichtig, die einen *veränderlichen Offset oder einen veränderlichen Verstärkungsfaktor* kompensiert. Je nach Aufgabe sollen auch beide Störgrößen kompensiert werden. Ein Beispiel für einen veränderlichen Verstärkungsfaktor sind im Bereich der Bildanalyse unterschiedliche Beleuchtungsstärken, die in der Wirkung einen Vektor $\underline{x}$ mit einer Konstanten κ_1 multiplizieren[1]:

$$\underline{x} \rightarrow \kappa_1 \cdot \underline{x} \qquad \kappa_1 > 0$$

Eine Klassifikation sollte so parametrisierbar sein, daß sich ein von κ_1 unabhängiges Ergebnis ergibt. In diesem Fall wird nicht die Helligkeit des Musters bewertet, sondern seine *innere Struktur*. Für einen quadratischen Klassifikator muß also gelten:

$$sgn(\kappa_1^2 \underline{x}^T A \underline{x} + \underline{w}^T(\kappa_1 \cdot \underline{x}) + a) = const. \qquad \kappa_1 > 0$$

Invarianz wird erfüllt für die 3 Fälle $A = 0 \quad a = 0$, $A = 0 \quad \underline{w} = \underline{0}$ und $\underline{w} = \underline{0} \quad a = 0$.
Andererseits kann auch die Helligkeit Information enthalten und muß geeignet bewertet werden können. Es soll deshalb von den Daten abhängen, ob die Parameter des Klassifikators entsprechend eingestellt werden. Die Optimierung wird dies wieder so durchführen, daß die extrahierte Information maximal wird. Allerdings kann Helligkeitsinvarianz nur dann erzeugt werden, wenn die *Struktur* des Klassifikationsgesetzes dies zuläßt.

Neben der Verstärkung kann zusätzlich auch ein störender und eventuell auch schwankender *Offset* vorhanden sein. Ursache können ebenfalls unterschiedliche Beleuchtungsstärken sein in Kombination mit einer die Helligkeitswerte logarithmierenden Kamera. Der Offset bewirkt die Addition des Vektors $\underline{1} = \left[1 \ \ldots \ 1\right]^T$ zu dem Mustervektor $\underline{x}$, multipliziert mit einer Konstanten κ_2 :

$$\underline{x} \rightarrow \underline{x} + \kappa_2 \cdot \underline{1} \qquad \kappa_2 \ beliebig$$

Für beliebigen, aber festen Vektor $\underline{x}$ soll entsprechend gelten können:

$$sgn((\underline{x} + \kappa_2 \cdot \underline{1})^T A(\underline{x} + \kappa_2 \cdot \underline{1}) + \underline{w}^T(\underline{x} + \kappa_2 \cdot \underline{1}) + a) = const. \qquad \kappa_2 \ beliebig$$

1 Unter der Voraussetzung einer homogenen Veränderung der Beleuchtung im betrachteten Bildausschnitt.

Dies wird für $A = 0$ und $\underline{w}^T \underline{1} = 0$ erreicht, in diesem Fall muß der Gleichanteil $\sum_{i=1}^{n} w_i$ des Vektors 0 sein. In der Konsequenz kann der quadratische Klassifikator so dimensioniert werden, daß er die beiden wichtigsten Störgrößen vollkommen kompensiert. Dies gelingt auch, wenn beide Störgrößen gleichzeitig wirksam sind, in diesem Fall muß $a = 0$, $A = 0$ gelten und der Gleichanteil des Vektors 0 sein.

Das Nullsetzen von Offset, Gleichanteil und der Matrix des quadratischen Teils muß selbstverständlich nicht der Entropieoptimierung überlassen werden. Es kann auch durch algorithmische Eingriffe als mathematische Nebenbedingung des Optimierungsproblems beim Training erzwungen werden. Es kann dann allerdings nicht vorab entschieden werden, ob dadurch zuviel Information verloren geht.

Neben der *impliziten* Kompensation durch den Klassifikator gibt es natürlich grundsätzlich die Möglichkeit, die zu klassifizierenden Vektoren *explizit* auf einen bestimmten Offset, beispielsweise 0, und auf eine bestimmte Varianz der Vektorelemente zu normieren[1]. Dies entspricht ebenfalls der Kompensation der beiden Störgrößen κ_1 und κ_2. Ein Vorteil der Methode besteht in der möglichen Kombination von Verstärkungs- und Offsetinvarianz mit quadratischen Klassifikatoren, also $A \neq 0$. Nachteilig ist hier, daß diese Art der Kompensation zusätzliche Rechenoperationen bei der Erkennung benötigt. Ein weiterer Nachteil der Methode besteht in dem Verlust von Information über Offset und Varianz.

2.5 Lineare Klassifikatoren

2.5.1 Motivation

Die obigen Ausführungen zur Invarianz gegen Beleuchtungsschwankungen legen nahe, die Matrix A des quadratischen Klassifikators auf 0 zu setzen. Dies ermöglicht einerseits eine Klassifikation mit *erheblich geringerem Aufwand* an Rechenzeit und Speicher. Andererseits wird bei entsprechender Variabilität der Beleuchtung der quadratische Anteil des Klassifikators an Bedeutung verlieren und die Einträge der Matrix A werden folglich ohnehin kleine Werte annehmen.

Als weiteres Problem quadratischer Klassifikatoren erscheint eine hyperbolische Trennfläche. Eine solche Trennfläche ist nicht zusammenhängend. Ist für die Klassifikation der Trainingsmenge nur ein Ast der Hyperbel relevant, kann der zweite Ast bei

1 In der Literatur oft als Helligkeits- und Kontrastnormierung bezeichnet.

der Erkennung zu Problemen führen; in unüblichen Situationen werden Muster als 'Objekt' erkannt, die mit den Objektmustern in der Trainingsmenge keinerlei Ähnlichkeit besitzen.

Demgegenüber ist ein linearer Klassifikator weniger mächtig, er zeigt aber ausgehend von der Trennebene ein stetiges Extrapolationsverhalten und läßt deshalb bessere Generalisierungseigenschaften des gesamten Erkenners erwarten. Durch eine entsprechende Anzahl linearer Klassifikatoren innerhalb des Musterbaumes kann jede beliebige Trennfläche stückweise linear approximiert werden, der lineare Klassifikator stellt somit keine Einschränkung für die gesamte Vorgehensweise dar. Er lautet:

$$sgn(\underline{w}^T \underline{x} + a)$$

Die Trennfläche ist eine Hyperebene, die auf dem Vektor $\underline{w}$ senkrecht steht. Ähnlich wie bei quadratischen Klassifikatoren wird im folgenden ein mehrstufiges Verfahren vorgelegt. Unter Annahme normalverteilter Wahrscheinlichkeitsdichtefunktionen und ausgehend von einer sehr einfachen Initialisierung wird zunächst halbanalytisch eine Näherungslösung für die Parameter berechnet. Diese dient in einem 2. Schritt zur numerischen Lösung anhand der originalen Vektormengen.

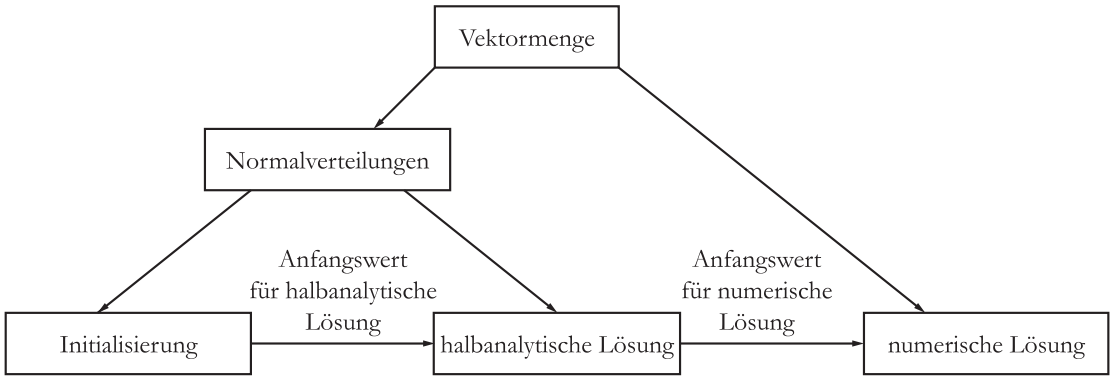

ABBILDUNG 2.5: Mehrstufige Berechnung einer Lösung. Das Resultat der Initialisierung dient als Anfangsnäherung der halbanalytischen Berechnung. Deren Resultat wird als Anfangsnäherung für die numerische Lösung anhand der originalen Vektormenge verwendet.

2.5.2 Initialisierung

Als Anfangswert für die Nullstellensuche wird jene Ebene verwendet, die auf der Verbindungsgeraden der beiden Klassenmittelpunkte senkrecht steht und von beiden Punkten denselben Abstand besitzt:

$$\underline{w} = \underline{m}_1 - \underline{m}_2 \qquad a = \frac{\underline{m}_2^T \underline{m}_2 - \underline{m}_1^T \underline{m}_1}{2}$$

Alternativ kann auch das Klassifikationsgesetz nach Fisher [20] angewandt werden; es benötigt aber die Invertierung einer Matrix.

2.5.3 Halbanalytische Näherungslösung

Die pragmatische Festlegung der Parameter im vorhergehenden Abschnitt wird hier verbessert. Da der Klassifikator ohnehin in einem weiteren Schritt anhand der Vektormengen optimiert wird, wird hier eine vereinfachte Lösung betrachtet:

- Normalverteilungen $g_1(\underline{x})$ und $g_2(\underline{x})$.
- Fehlerminimierung statt Entropieoptimierung.
- Um die Existenz einer Lösung sicherzustellen und den Fall nach Abb. 2.2 auszuschließen, wird $p_1 = p_2 = 0.5$ gesetzt. Für größtmögliche Allgemeinheit der Darstellung wird die Herleitung mit beliebigen Wahrscheinlichkeiten p_1, p_2 angegeben.

Ein zunächst unbekannter Vektor $\underline{x}$ wird mit dem linearen Klassifikator $sgn(\underline{w}^T \underline{x} + a)$ bewertet. Ist das Ergebnis -1, wird der Vektor der Klasse 1 zugeordnet, andernfalls der Klasse 2. Die Wahrscheinlichkeit für eine falsche Zuordnung wird als Fehlerwahrscheinlichkeit p_F bezeichnet, sie ergibt sich zu

$$p_F = p_1(2) + p_2(1)$$

Dies ist bereits die zu minimierende Zielfunktion. Ihren Zusammenhang mit den Parametern des Klassifikators erhält man, indem die Wahrscheinlichkeiten $p_1(2)$ und $p_2(1)$ mittels Integralen über dem Vektorraum $\mathbb{R}^n$ angeschrieben werden:

$$p_1(2) = \frac{1}{2} p_1 \left(1 + \int_{\mathbb{R}^n} g_1(\underline{x}) sgn(\underline{w}^T \underline{x} + a) d\underline{x} \right)$$

und

$$p_2(1) = \frac{1}{2} p_2 \left(1 - \int_{\mathbb{R}^n} g_2(\underline{x}) sgn(\underline{w}^T \underline{x} + a) d\underline{x} \right)$$

Damit ergibt sich für die Gütefunktion der Ausdruck

$$p_F = \frac{1}{2}p_1\left(1 + \int_{I\!R^n} g_1(\underline{x})\,sgn(\underline{w}^T\underline{x} + a)d\underline{x}\right) + \frac{1}{2}p_2\left(1 - \int_{I\!R^n} g_2(\underline{x})\,sgn(\underline{w}^T\underline{x} + a)d\underline{x}\right)$$

Der konstante Anteil ist für die Minimierung irrelevant, es ergibt sich also folgendes Optimierungsproblem:

$$p_F^* = \frac{1}{2}\int_{I\!R^n} (p_1 g_1(\underline{x}) - p_2 g_2(\underline{x}))\,sgn(\underline{w}^T\underline{x} + a)d\underline{x} \rightarrow min$$

Eine analytische Lösung des Integrales ist nicht bekannt, allerdings können die Ableitungen nach den Klassifikatorparametern $\underline{w}$ und a analytisch berechnet werden, diese sind:

$$\frac{d}{da}p_F^* = \int_{I\!R^n} (p_1 g_1(\underline{x}) - p_2 g_2(\underline{x}))\delta(\underline{w}^T\underline{x} + a)d\underline{x} \tag{2.13}$$

und

$$\frac{d}{d\underline{w}}p_F^* = \int_{I\!R^n} (p_1 g_1(\underline{x}) - p_2 g_2(\underline{x}))\delta(\underline{w}^T\underline{x} + a)\underline{x}d\underline{x} \tag{2.14}$$

Die beiden Integrale können analytisch für Normalverteilungen $g_1(\underline{x})$ und $g_2(\underline{x})$ gelöst werden. Die Herleitungen sind verhältnismäßig umfangreich und in Anhang C angegeben. Es gilt:

$$\int_{I\!R^n} g(\underline{x})\delta(\underline{w}^T\underline{x} + a)d\underline{x} = \frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C\underline{w}}}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m} + a)^2}{\underline{w}^T C\underline{w}}}$$

und

$$\int_{I\!R^n} g(\underline{x})\delta(\underline{w}^T\underline{x} + a)\underline{x}d\underline{x} = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m} + a)^2}{\underline{w}^T C\underline{w}}} \cdot \frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C\underline{w}}^3}(\underline{m}\underline{w}^T - (\underline{w}^T\underline{m} + a)E)C\underline{w}$$

Die Ableitungen berechnen sich damit zu

$$\frac{d}{d\underline{w}}p_F^* = \left(p_1 \cdot e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m}_1 + a)^2}{\underline{w}^T C_1 \underline{w}}} \cdot \frac{(\underline{m}_1\underline{w}^T - (\underline{w}^T\underline{m}_1 + a)E)C_1\underline{w}}{\sqrt{\underline{w}^T C_1 \underline{w}}^3} - p_2 \cdot e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m}_2 + a)^2}{\underline{w}^T C_2 \underline{w}}} \cdot \frac{(\underline{m}_2\underline{w}^T - (\underline{w}^T\underline{m}_2 + a)E)C_2\underline{w}}{\sqrt{\underline{w}^T C_2 \underline{w}}^3} \right) \frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{2\pi}} \tag{2.15}$$

und

$$\frac{d}{da}p_F^* = p_1 \cdot \frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C_1 \underline{w}}}\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m}_1 + a)^2}{\underline{w}^T C_1 \underline{w}}} - p_2 \cdot \frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C_2 \underline{w}}}\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m}_2 + a)^2}{\underline{w}^T C_2 \underline{w}}} . \tag{2.16}$$

Im Optimum muß $\frac{d}{d\underline{w}}p_F^* = \underline{0}$ und $\frac{d}{da}p_F^* = 0$ gelten; eine analytische Lösung ist allerdings nicht bekannt. Die optimale Lösung kann aber numerisch bestimmt werden. Die analytische Lösung der Integrale erspart also nicht die numerische Lösung des nichtlinearen Gleichungssystems in $n + 1$ Variablen; sie ersetzt aber das in dem hochdimensionalen Raum kaum mögliche, numerische Integrieren. Für die folgende iterative Lösung steht nicht der Funktionswert der Zielfunktion zur Verfügung, statt dessen nur seine Ableitung. Allerdings ist das Optimierungsproblem gut konditioniert. Eine Optimumssuche in Richtung des maximalen Abstiegs führt dementsprechend schnell zu einer Lösung.

Stabilität und Konvergenzgeschwindigkeit der Optimierung hängen von einer geeigneten Schrittweitensteuerung ab. Für eine effiziente Optimierung muß die Anfangsschrittweite in der selben Größenordnung sein wie der Parametervektor selbst. Damit sich dieser nicht aus dem Einzugsbereich des von der Initialisierung vorgegebenen Minimums bewegt, wird die Änderung des Parametervektors kleiner gewählt, beispielsweise auf den 0.1-fachen Wert.

Die Schrittweite wird im Verlauf der Optimierung angepaßt. Kriterium für die Verkleinerung ist der Betrag der Ableitung. Da der Betrag der Ableitung im Minimum 0 ist und von dort in alle Richtungen ansteigt, kann er als Abstandsmaß verwendet werden. Die Schrittweite wird dann verkleinert, wenn dieses Abstandsmaß von einer Iteration zur nächsten ansteigt. Die Schrittweite wird geringfügig erhöht, wenn dieses Abstandsmaß kleiner wird. Als Maximalwert wird hier die Schrittweite aus dem 1. Schritt der Iteration vorgegeben. Die genauen Werte für Verkleinerung und Erhöhung der Schrittweite sind unkritisch, eine Verkleinerung um den Faktor 2 und eine Vergrößerung um den Faktor 1,1 ergeben in der Praxis gute Ergebnisse.

Diese Schrittweitenregelung erlaubt auch die Formulierung eines Abbruchkriteriums. Bei einer Annäherung an das Minimum werden der Betrag der Ableitung und die Schrittweite kleiner. Fällt die Schrittweite unter eine bestimmte Schwelle, wird die Optimierung abgebrochen. Da das Verfahren schnell konvergiert, kann eine sehr kleine Schwelle angegeben werden. Andererseits ist das mit diesem Verfahren berechnete Minimum ohnehin nur die Initialisierung der folgenden, anhand der Vektormengen durchgeführten Optimierung. Ähnliche Überlegungen wie in Abschnitt 2.4.3 führen auf eine relative Schwelle von $10^{-3} \cdot \left| \left[\underline{w}^T \; a \right] \right|$.

2.5.4 Numerische Lösung

Die numerische Optimierung des linearen Klassifikators anhand der Vektormengen erfolgt vollkommen analog zu der Optimierung des quadratischen Klassifikators. Unterschiedlich ist die Initialisierung des Verfahrens und selbstverständlich die hier zu 0 gesetzte Matrix A. Auch der Algorithmus zur Optimierung und die Schrittweitenregelung sind identisch.

2.6 Lineare Invarianz

In vielen praktisch relevanten Fällen kann eine bestimmte *lineare Vortransformation* der Eingangsdaten gewählt werden. Ein typisches Beispiel ist die Repräsentation farbiger Bilder, deren Pixel üblicherweise im RGB-, YHS- oder CMK-Format dargestellt werden. Jedes dieser Formate besteht aus einem Vektor der Dimension 3. Die verschiedenen Darstellungen können durch Multiplikation mit geeigneten, quadratischen Transformationsmatrizen ineinander überführt werden. Es ergibt sich also eine lineare Transformation der Form

$$\underline{z} = T\underline{x}$$

mit einer repräsentationsabhängigen Matrix T und den zwei unterschiedlichen Repräsentationsformen $\underline{z}$ und $\underline{x}$. Diese Transformation beinhaltet als Sonderfall auch unterschiedliche Skalierung der Eingangsdaten, in diesem Fall ergibt sich als Transformationsmatrix T eine Diagonalmatrix. Dieser Sonderfall ist besonders von Interesse, da in vielen Farbmodellen wie dem YHS-Format (Helligkeit, Farbton, Sättigung) die Skalierung aus rein technischen Gesichtspunkten festgelegt wird. Eine weitere, lineare Vortransformationen ist die beispielsweise in [43] gewählte Cosinustransformation. Insgesamt stellt sich die Frage, welche Repräsentationsform gewählt werden soll und im folgenden wird gezeigt, daß dies bei Verwendung der Entropieminimierung irrelevant ist.

Gibt es für eine Darstellung $\underline{x}$ und den quadratischen Klassifikator optimale Parameter A, $\underline{w}$ und a, so lassen sich für die Repräsentation in $\underline{z}$ ebenfalls optimale Parameter finden. Es gilt:

$$\underline{x}^T A\underline{x} + \underline{w}^T\underline{x} + a = (T^{-1}\underline{z})^T A(T^{-1}\underline{z}) + \underline{w}^T(T^{-1}\underline{z}) + a = \underline{z}^T(T^{-1})^T AT^{-1}\underline{z} + \underline{w}^T T^{-1}\underline{z} + a \quad (2.17)$$

Offensichtlich gibt es also für die Darstellung in $\underline{z}$ ebenfalls Parameter

$$A^* = (T^{-1})^T AT^{-1}\,,\ \underline{w}^* = (T^{-1})^T\underline{w} \text{ und } a^* = a,$$

die das selbe Klassifikationsgesetz realisieren und dementsprechend die selbe Information extrahieren. Der lineare Klassifikator ist als Sonderfall $A = 0$ in (2.17) enthalten.

3 Musterbäume

3.1 Das Klassifikationsproblem

3.1.1 Klassifikation mit Musterbäumen

Das allgemeine Vektor-Klassifikationsproblem besteht darin, einen Vektor $\underline{x} \in \mathbb{R}^n$ einer von mehreren Klassen zuzuordnen, ein Überblick über die Thematik findet sich in [45]. Jede dieser Klassen ist durch eine Trainingsstichprobe repräsentiert. Das Problem taucht auch in vielen anderen Aufgabenstellungen der Informatik auf, neben der Objekterkennung beispielsweise auch bei der Erkennung natürlicher Sprache oder bei der Fehler-Erkennung in industriellen Anlagen.

Zur Lösung des Problems existieren mehrere Verfahren; das NN-Verfahren (Nearest Neighbor[1]) führt im Fall der Trennbarkeit der Klassen zur korrekten Lösung. Sind die Klassen nicht trennbar, ergibt das k-NN-Verfahren[2] die im Sinne eines Bayesklassifikators optimale Lösung. Diese Verfahren sind aufwendig und werden angewandt, wenn die Trainingsmenge klein und die verfügbare Rechenzeit groß ist. Für die schnelle Objekterkennung in Videosequenzen sind beide Voraussetzungen nicht erfüllt.

Eine sehr schnelle Klassifikation erlauben Entscheidungsbäume; für die Vektorklassifikation wird ein binärer Musterbaum nach Abb. 1.4 verwendet. Im Wurzelknoten findet eine Klassifikation entsprechend Kapitel 2 statt. Je nach Ergebnis der Klassifikation wird als Folgeknoten Knoten 1 oder Knoten 2 ausgewählt und dort wieder eine Klassifikation mit dem entsprechenden Klassifikator durchgeführt. Dies wird rekursiv so lange fortgesetzt, bis die Suche einen Endknoten des Baums erreicht. Das Verfahren grenzt im Verlauf der Klassifikation das Suchproblem immer weiter ein, d.h. jede einzelne Klassifikation wird nicht nur zur Extraktion von Information genützt. Daraus resultiert die nur *logarithmische* Zunahme der Suchzeit mit der Größe des Baums und die für die Objekterkennung notwendige Geschwindigkeitssteigerung. Während die optimale Dimensionierung eines einzelnen Klassifikators bereits gezeigt wurde, soll im folgenden die Erzeugung des Baums erläutert werden.

---

1 Beim ´Nearest Neighbor´-Verfahren wird die Entfernung des unbekannten Vektors zu jedem Vektor im Trainingsset bestimmt. Der Trainingsvektor mit der kleinsten Entfernung, der ´Nächste Nachbar´ wird ausgewählt und der unbekannte Vektor der Klasse dieses Vektors zugeordnet. Die wesentliche Herausforderung besteht bei diesem Verfahren in der Suche eines möglichst kleinen Trainingssets.

2 Beim k-NN-Verfahren werden die k nächsten Nachbarn bestimmt, also der nächste, der 2.-nächste bis zum k-nächsten Nachbarn. Aus diesen k nächsten Nachbarn wird die häufigste Klasse bestimmt und dem unbekannten Vektor als Klassifikationsergebnis zugewiesen. Das Verfahren ist von Vorteil, wenn sich die Trainingsklassen im Vektorraum überlappen. In diesem Fall wäre es beim NN-Verfahren Zufall, welcher Klasse ein unbekannter Vektor zugewiesen wird. Das k-NN-Verfahren konvergiert für unendlich große Trainingsmenge und unendlich großes k gegen den idealen Bayes-Klassifikator. Im übrigen ist das NN-Verfahren der Sonderfall $k = 1$ des k-NN-Verfahrens.

3.1.2 Baumerzeugung

Voraussetzung für die Baumerzeugung (Training) ist eine vorgegebene Trainingsmenge von Vektoren mindestens zweier Klassen, also $K \geq 2$. Jede Klasse k enthält N_k Vektoren. Es wird angenommen, daß diese Vektoren aufgabenorientiert gewählt wurden und daß deshalb das Problem gelöst werden kann, wenn diese Trainingsvektoren korrekt klassifiziert werden. Die Auswahl dieser Vektoren selbst ist abhängig von der jeweiligen Anwendung und wird in Kapitel 4 diskutiert.

Zu Beginn der Trainings[1] ist die Struktur des Baums nicht bekannt. Entsprechend wird zunächst lediglich der Wurzelknoten und die zwei Folgeknoten 1 und 2 angelegt, entsprechend Abb. 3.1.

ABBILDUNG 3.1: Musterbaum zu Beginn des Verfahrens.

Für die Parametrierung des Klassifikators im Wurzelknoten wird die gesamte Trainingsmenge verwendet. Anschließend wird die gesamte Trainingsmenge testweise klassifiziert und geprüft, ob einer der beiden Knoten 1 oder 2 bereits nur noch Vektoren einer einzigen Klasse aufnimmt. Der Knoten wird in diesem Fall ein Endknoten und erhält für die Klassifikation den Namen derjenigen Klasse, die bei der testweisen Klassifikation von ihm aufgenommen wurde. Wurden bei der testweisen Klassifikation Vektoren unterschiedlicher Klassen von einem Knoten aufgenommen, erhält er keinen Klassennamen, statt dessen wiederum zwei Folgeknoten. Dies ist im Beispiel nach Abb. 3.2 für beide Knoten der Fall, wir erhalten also insgesamt 4 Knoten in der 3. Schicht.

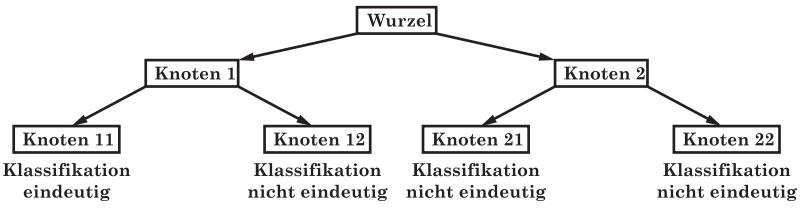

ABBILDUNG 3.2: Baum nach Erzeugung der 3. Schicht.

1 Das Prinzip der Baumerzeugung wird erstmalig in [38] beschrieben.

Es müssen also zusätzliche Klassifikatoren in den Knoten 1 und 2 angelegt und parametriert werden. Dazu werden für den Klassifikator diejenigen Vektoren verwendet, die bei der testweisen Klassifikation auch in den betreffenden Knoten gelangen. Dies bedeutet also, daß alle Trainingsvektoren, die in Knoten 1 gelangen, zur Parametrierung des Klassifikators in diesem Knoten verwendet werden, die Trainingsmenge wird also aufgeteilt. Das Verfahren wird in jeder Schicht fortgesetzt und führt zu Endknoten, wie in Abb. 3.3 dargestellt.

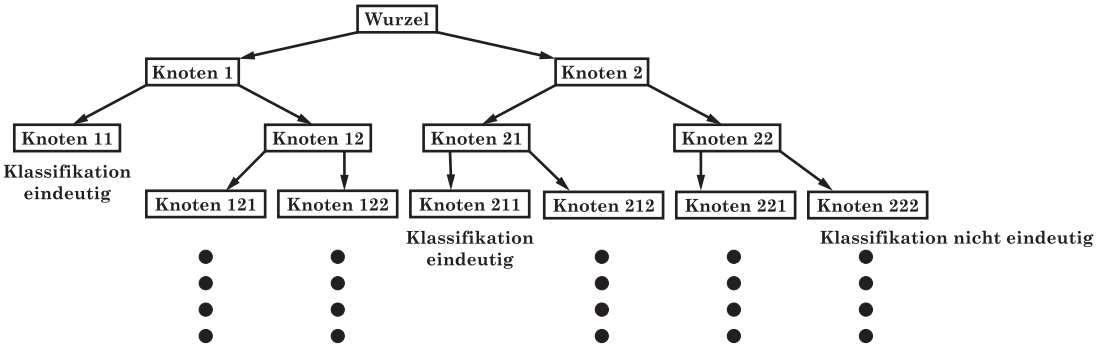

ABBILDUNG 3.3: Anlegen der Folgeknoten für nicht eindeutige Knoten und Bildung von eindeutigen Endknoten.

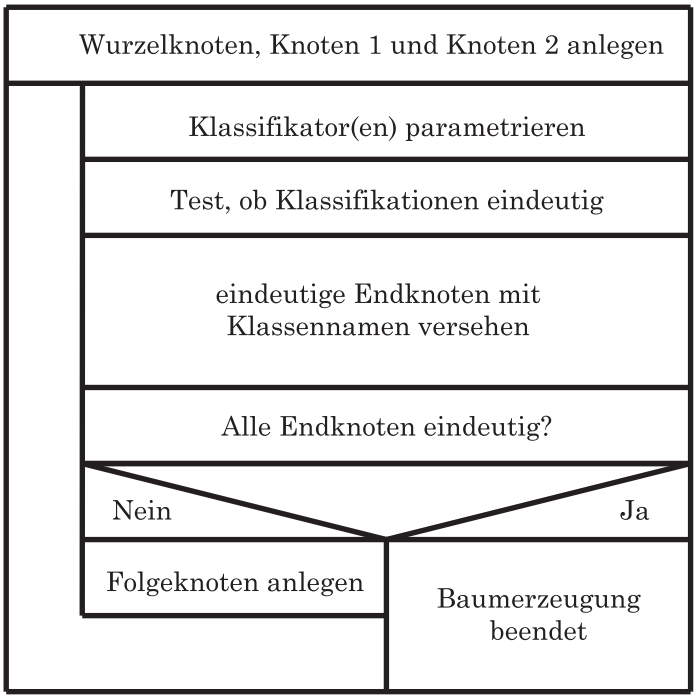

ABBILDUNG 3.4: Der Algorithmus zur Baumerzeugung.

Das gesamte Verfahren terminiert dann, wenn in alle Endknoten bei testweiser Klassifikation nur noch Vektoren einer einzigen Klasse fallen. Für den Algorithmus ergibt sich das Struktogramm nach Abb. 3.4.

Der vollständige Baum extrahiert die gesamte, gesuchte Information aus der Trainingsmenge.

3.2 Diskussion

3.2.1 Komplexität von Training und Erkennung

Die für die Objekterkennung wichtigste Eigenschaft der Musterbäume ist ihre geringe Komplexität bei der Erkennung, also die nur logarithmische Zunahme des Rechenaufwands mit der Größe des Baums. Die maximal notwendige Anzahl an Klassifikationen ergibt sich bei einer Knotenanzahl G in ihrer Größenordnung zu $log_2(G)$. Die Anzahl E der Endknoten eines Baumes berechnet sich daraus zu

$$E = \frac{G+1}{2}. \tag{3.1}$$

Die notwendige Anzahl an Klassifikationen ergibt sich für einen *symmetrischen* Baum also zu nur $log_2(2 \cdot E - 1)$. Da jeder Endknoten einem Teilraum oder Fragment des gesamten Vektorraumes entspricht, dem individuell ein Erkennerergebnis zugeordnet werden kann, steigt der Aufwand für die Erkennung nur logarithmisch mit der Fragmentierung des Raumes. Insgesamt kann also mit x Klassifikationen zur Laufzeit eine Auswahl aus 2^x Teilräumen getroffen werden.

Komplexere Objekte führen zu einer stärkeren Untergliederung des Vektorraumes und damit zu größeren Bäumen. Da diese Untergliederung aber exponentiell von der Baumtiefe abhängt, wird letztere kaum größer; dies zeigen auch die Experimente in Kapitel 6. Dementsprechend steigt die Dauer für die Erkennung auch nur unwesentlich, da diese von der Tiefe abhängig ist. Dies ist ein im Vergleich zu merkmalsbasierten Methoden insofern überraschendes Ergebnis, da dort die Rechenzeit für die Zuordnung der Merkmale mit der Anzahl und damit der Komplexität des Objekts steigt.

Im Gegensatz zur Baumtiefe steigt die Anzahl der Knoten mit der Komplexität des Objekts stark an. Diese ist für die Rechenzeit des Trainings wesentlich, so daß das Training eines komplexen Objekts bedeutend länger dauert als das Training eines einfachen Objekts. Während also die Rechenzeit für die Erkennung für alle Objekte ähnlich ist, besteht die Limitierung des Verfahrens in der hohen Trainingszeit.

Komplexere Objekte führen zu einer größeren Trainingsmenge. Mit der Größe der Trainingsmenge steigt die Komplexität des Trainings mehr als linear. Eine lineare Erhöhung der Trainingszeit ergäbe sich, wenn jeder Trainingsvektor 2 mal in jedem Schritt verwendet wird. Das würde die Struktur des Baums nicht verändern, und deshalb genau zu einer Verdoppelung des Rechenaufwandes führen. Wird durch die Vergrößerung der Trainingsmenge auch die Komplexität des Problems vergrößert, also weitere Ansichten des Objekts in die Trainingsmenge aufgenommen, vergrößert sich in der Folge die Anzahl der Knoten. Es ergibt sich somit eine mehr als lineare Zunahme der Rechenzeit. Wie stark dieser Anstieg ist, hängt vom Problem ab; er wächst aber mindestens linear mit der Trainingsmenge.

Allgemein muß festgestellt werden, daß durch die mindestens lineare Abhängigkeit der Trainingszeit und der nur logarithmischen Abhängigkeit der Erkennungszeit das Training größere Probleme bereitet als die Erkennung. Dieses Problem verschärft sich bei größeren Trainingsmengen; es ist bereits jetzt die eigentliche Grenze für diese Art der Objekterkennung.

3.2.2 Generalisierungsfähigkeit

Das Verfahren stellt sicher, daß die *Trainingsmenge* korrekt getrennt wird. Dies gilt selbstverständlich nur, wenn die Menge getrennt werden kann, wenn also nicht identische Vektoren in beiden Klassen enthalten sind. Eine Aussage bezüglich der Verallgemeinerungsfähigkeit, also eine Aussage für *andere Vektoren* als die in der Trainingsmenge enthaltenen, ist allerdings nicht möglich. Trotzdem läßt sich diese Eigenschaft über die Informationsoptimierung beeinflussen. Vergleicht man ein Training mit und ohne Informationsoptimierung, ergeben sich sehr unterschiedliche Größen für die erzeugten Bäume. Die Optimierung erhöht die Information, die ein einzelner Knoten extrahiert. Da die gesamte Information nur von der Trainingsmenge abhängt und vom Trainingsalgorithmus unabhängig ist, werden also zur Extraktion der gesamten Information bei Optimierung weniger Knoten benötigt. Entsprechend (3.1) ergeben sich damit weniger Endknoten und in weiterer Konsequenz folgt eine geringere Fragmentierung des Musterraumes.

Nun ist die Verallgemeinerungsfähigkeit der Klassifikation größer, wenn es weniger Fragmente mit einer jeweils größeren Anzahl an enthaltenen Vektoren gibt. Die durch weniger Knoten oder kleinere Bäume erzeugte Trennfläche ist also vergleichsweise ´glatt´ bei identischer Trainingsmenge. Dies entspricht der Vermutung, daß die realen Klassengrenzen ebenfalls glatt sind und eine Approximation mit weniger Trennflächen bzw. Knoten den tatsächlichen Gegebenheiten näher kommt. Eine Diskussion dieses Sachverhalts findet sich in [71].

Neben der Betrachtung der Anzahl an Fragmenten kann man umgekehrt auch argumentieren, daß bei geringerer Anzahl der Fragmente jedes einzelne im Durchschnitt eine größere Anzahl an Vektoren enthält. Jedes Fragment wird also durch eine größere Anzahl von Trainingsvektoren definiert. Dadurch steigt die Wahrscheinlichkeit, daß eine dem speziellen Klassifikationsproblem immanente Trennfläche gefunden wird und nicht eine zufällige. Die Optimierung entspricht also der Suche einer natürlichen Trennfläche. Ohne Optimierung werden zunächst fast beliebige Trennflächen in den Musterraum gelegt, bis dieser Raum so fein untergliedert ist, daß nur noch Vektoren einer Klasse in ein Fragment fallen. Diese kleinen Fragmente werden anschließend mit dem entsprechenden Klassennamen versehen. Dementsprechend sind die so erzeugten Fragmente im wesentlichen zufällig und die Erkennung wenig robust, was sich experimentell leicht zeigen läßt. Mit Optimierung werden die Trennflächen der Fragmente gezielt so verschoben, daß sie den realen Trennflächen der Klassen nahekommen.

Daß Klassifikatoren mit wenigeren Parametern besser verallgemeinern, ist ein vor allem im Bereich der Neuronalen Netze bekannte Tatsache. Bei Musterbäumen darf dafür aber nicht die Anzahl aller Skalare innerhalb der Gewichtungsvektoren betrachtet werden, sondern nur die Anzahl der Knoten. Aber auch die absolute Zahl der Knoten alleine läßt noch keine Aussage über die Verallgemeinerungsfähigkeit zu. Die Anzahl der Knoten hängt von der Komplexität des Problems ab, genauer von der Verteilung der Klassen im Musterraum.

Die Generalisierungsfähigkeit entspricht direkt der Robustheit des Systems gegen Störungen, dies sind zur Zeit des Trainings unbekannte Situationen. Für diese Robustheit kann kein direktes Maß angegeben werden, lediglich die Fehlerwahrscheinlichkeit für eine gegebene Testmenge.

3.2.3 Baumstruktur

3.2.3.1 Entropieoptimierung begünstigt Bildung von Endknoten

Falls ein Knoten Endknoten wird und keine weiteren Folgeknoten erhält, ist dies selbstverständlich für die Klassifikationsgeschwindigkeit von Vorteil. Es muß keine weitere Klassifikation durchgeführt werden. Kann ein Knoten durch eine *geringfügige* Parameteränderung (des Vorgängerknotens) zu einem Endknoten gemacht werden, ist dies für die gesamte Klassifikation günstig. Nimmt einer von 2 Folgeknoten (in der folgenden Betrachtung der Knoten 1) nur noch Muster einer Klasse auf, beispielsweise nur noch Muster der Klasse 2, wird $p_1(1) = 0$. Der Übergang von einem Knoten, der Muster von 2 Klassen aufnimmt, zu einem Endknoten, der nur noch Muster einer Klasse aufnimmt, wird also mathematisch durch den Grenzübergang $p_1(1) \to 0$ beschrieben.

Entsprechend folgt aus (2.6) der Zusammenhang $N_k = \Delta_k$. Zur Berechnung von $\frac{\partial H^{**}}{\partial \Delta_k}$ in (2.7) eingesetzt, ergibt sich:

$$\frac{\partial H^{**}}{\partial \Delta_k} = ln(p_1(1) \cdot Const)$$

Der konstante Anteil ist nicht von Interesse; er ändert sich während dem Grenzübergang nicht. Damit erhalten wir offensichtlich für den Betrag der Ableitung:

$$\lim_{p_1(1) \to 0} \left| \frac{\partial H^{**}}{\partial \Delta_k} \right| = \infty .$$

Ist also ein Knoten an der Grenze, Endknoten zu werden oder nicht, strebt die Ableitung durch die Entropieoptimierung gegen unendlich. In der Konsequenz wird die Optimierung dafür sorgen, daß der Knoten zu einem Endknoten wird[1].

Die Entropieoptimierung entspricht also dem anschaulichen Ziel, daß Knoten nach Möglichkeit zu Endknoten werden. Trotz der Behandlung des Problems als *kontinuierliches* Optimierungsproblem ergibt sich eine Begünstigung der *diskreten, eindeutigen Zuordnung* der Knoten zu einer der Klassen.

Für die numerische Optimierung ergeben sich durch die hier als unendlich berechneten Werte keine Probleme. Der Grund ist die Näherung der *sgn* -Funktion durch die $(2/\pi) \cdot atan$ -Funktion. Sie erreicht nicht die Werte $\{-1, 1\}$. Dadurch werden die Wahrscheinlichkeiten $p_k(1)$ und $p_k(2)$ nicht 0, und damit die Ableitungen $\frac{\partial H^{**}}{\partial \Delta_k}$ nicht unendlich. Es müssen allerdings die aus der Numerik bekannten Mechanismen angewandt werden, um Zahlenüberläufe zu vermeiden oder gegebenenfalls geeignet zu behandeln.

3.2.3.2 Asymmetrie der Musterbäume

Die Bildung von Endknoten kann experimentell bereits in den ersten Baumschichten beobachtet werden. Die Musterbäume sind also stark asymmetrisch. Die Endknoten in den ersten Schichten sind praktisch ausschließlich Knoten der Zurückweisungsklasse. Dieser Effekt beruht auf der unterschiedlichen Verteilung der Objektmuster und der Zurückweisungsmuster im Pixelraum. Während die Menge der Objektmuster in einem kleinen Gebiet des Pixelraumes liegt, erstreckt sich die Verteilung der Zurückweisungsmuster über ein vergleichsweise großes Gebiet. Es ist also relativ leicht möglich,

1 Dies ist insbesondere im Vergleich zur Fehlerminimierung ein Vorteil, die diese Eigenschaft nicht besitzt.

Teilräume vom gesamten Pixelraum zu trennen, die keine Objektmuster enthalten. Dies wird durch den in Abschnitt 3.2.3.1 beschriebenen Effekt begünstigt. Tabelle 3.1 zeigt eine typische Verteilung der Endknoten, abhängig von der jeweiligen Schicht.

Die Identifikation von Zurückweisungsmustern bereits in geringer Baumtiefe trägt deshalb zur hohen Geschwindigkeit der Klassifikation bei, weil es sehr viele Zurückweisungsmuster gibt und deren durchschnittliche Klassifikationszeit wesentlich die Gesamtzeit der Objektsuche bestimmt. Die Verwendung der Information als Gütefunktion kommt also auch dem Ziel einer schnellen Objekterkennung entgegen.

Schicht	Knoten pro Schicht	Endknoten für Zurückweisungsklasse	Endknoten für Objektklasse
0	1	0	0
1	2	0	0
2	4	0	0
3	8	1	0
4	14	3	0
5	22	5	0
6	34	5	0
7	58	12	0
8	92	25	0
9	134	44	0
10	180	66	0
11	228	99	0
12	258	130	0
13	256	132	0
14	248	130	7
15	222	74	60
16	176	77	59
17	80	39	32
18	18	9	9
Summe:	2035	851	167

TABELLE 3.1: Baumstruktur am Beispiel des Objekts ´Spitzer´. Weitere Daten zu diesem Objekt sind in Tabelle 6.1 angegeben.

Die Klassifikation von Objektmustern terminiert stets in *tiefen Baumschichten*, im Beispiel nach Tabelle 3.1 erst in den Schichten 14 bis 18. Erklärt werden kann das dadurch, daß Objektmuster für ihre Klassifikation von vielen, sehr ähnlichen Mustern der Zurückweisungsklasse unterschieden werden müssen. Dazu sind bei der Erken-

nung vergleichsweise viele Einzelklassifikationen notwendig. Dieser großen Anzahl von Einzelklassifikationen müssen auch jene Zurückweisungsmuster unterzogen werden, die dem Objekt ähnlich sind. Je ähnlicher die Muster dem Objekt sind, desto aufwendiger muß also der gesamte Klassifikationsvorgang sein, desto seltener sind diese Muster aber auch; sie erhöhen deshalb nicht wesentlich die Zeit für die Objektsuche in einem Bild. Andererseits zeigen diese Überlegungen, daß die Zeit für die Objekterkennung auch vom Bildinhalt abhängt.

Ein weiterer Schluß kann aus Tabelle 3.1 gezogen werden. Ein Objektknoten entspricht einer bestimmten Region im Pixelraum. Eine solche Objektregion wird im Beispiel mit maximal 18 Hyperebenen vom restlichen Pixelraum abgegrenzt. Dieser Raum hat jedoch eine viel höhere Dimension, beispielsweise $31x31 = 961$. Die Abgrenzung eines endlichen Gebietes in einem n-dimensionalen Raum benötigt minimal $n + 1$ Ebenen, hier also 962. Daraus folgt zunächst, daß die durch einen Endknoten definierte Region unendlich groß ist. Darüber hinaus werden indirekt auch durch die Amplitudenbeschränkung der Pixel auf ihren [schwarz; weiß]-Wertebereich Grenzen gezogen. Diese Grenzen stellen einen Hyperwürfel mit einer Kantenlänge dar, die dem Wertebereich der Pixelwerte entspricht. Dies schränkt den Raum weiter ein. Jeder Teilraum jedes Knotens erstreckt sich deshalb nicht bis ins Unendliche, tangiert aber die Flächen des Hyperwürfels.

3.2.4 Geschwindigkeit entropieoptimierter Musterbäume

Durch die Entropieminimierung in jedem einzelnen Knoten wird in jeder Klassifikation mehr Information extrahiert als ohne diese Optimierung. Die extrahierte Information pro Rechenoperation wird also größer. Für die gesuchte Information müssen folglich weniger Rechenoperationen durchgeführt werden, die Objekterkennung wird schneller[1].

Die Vergrößerung der extrahierten Information bezieht sich lediglich auf einen einzelnen Knoten. Allerdings kann sich durch die Optimierung auch die Struktur des Baums ändern. Insbesondere kann seine maximale Tiefe zunehmen. In diesem Fall wird ein größerer Anteil der Information in tieferen Schichten extrahiert mit der Konsequenz, daß dort aufgrund der höheren Musterauflösung zusätzlicher Aufwand entsteht und deshalb die Effektivität abnimmt. Auch wenn im obigen Beispiel durch die Optimierung die Anzahl der Knoten reduziert wurde, kann eine Verlangsamung nicht grundsätzlich ausgeschlossen werden. Allerdings wurde in allen Experimenten eine spürbare Erhöhung der Rechengeschwindigkeit festgestellt, die aber von Objekt zu Objekt unterschiedlich ist.

1 Der Wert 'Information pro Rechenoperation' entspricht der 'Information Value' nach [64].

Die Informationsoptimierung führt allgemein zu stark asymmetrischen Bäumen. Dies gilt allgemein als Nachteil für Entscheidungsbäume, da einzelne Klassifikationen sehr lange dauern können. Die maximale Zeit für eine Klassifikation ist die Zeit für diejenigen Vektoren, die in den Endknoten der letzten Schicht klassifiziert werden.

Für die Bildanalyse ist dies an sich kein Nachteil, da nicht die maximale Zeit für eine einzelne Klassifikation zählt, sondern die *durchschnittliche* Zeit für die Klassifikation aller Vektoren eines Bildes. Die Minimierung der durchschnittlichen Zeit erfordert sogar einen asymmetrischen Baum, der viele Vektoren in geringer Tiefe klassifiziert. Dies ist die große Menge der Zurückweisungsmuster, die dem gesuchten Objekt sehr unähnlich sind. Die Entropieopimierung mit ihrer in Abschnitt 3.2.3.1 diskutierten Eigenschaft führt also auch über diesen Effekt zu einer wirkungsvollen Geschwindigkeitssteigerung.

4 Ansichtsbasierte Objekterkennung mit Musterbäumen

4.1 Trainingsmenge

4.1.1 Vorüberlegungen

Die bisher diskutierten Musterbäume werden auf die Erkennung von Objekten in Bildern angewandt, indem entsprechend Abb. 1.3 ein Klassifikationsfenster über das Bild geschoben und nach jeder Verschiebung der Inhalt des Fensters klassifiziert wird. Dazu werden die Pixel des Suchfensters in einen Vektor sortiert und sind Eingang des Musterbaums. Der Vergleich zwischen Objekt und Modell basiert also nur auf der Objektansicht, deshalb der Name der Verfahren. Es handelt sich also um einen reinen 2D-2D-Vergleich.

Das Objekt kann nur in Situationen erkannt werden, die auch in der Trainingsmenge enthalten sind. Eine allgemeingültige Regel zur Auswahl der Trainingsmenge ist nicht bekannt. Diese Regel müßte alle möglichen Variationen in der geometrischen Relation Kamera-Objekt, in der Beleuchtung und im Hintergrund des Objekts berücksichtigen. Diese Problematik ist sehr allgemeiner Natur und taucht auch bei merkmalsbasierten Algorithmen auf. Das Problem scheint grundsätzlich keine Lösung zu besitzen, in [42] wird es kurz diskutiert.

Aus einem ähnlichen Grund ist es prinzipiell unmöglich, *situationsunabhängige* Wahrscheinlichkeiten für die Erkennung eines Objekts anzugeben. Für eine solche Angabe müßten ebenfalls alle möglichen Variationen berücksichtigt werden. Außerdem ist es, abgesehen von trivialen Fällen, kaum möglich, Situationen zu beschreiben und so wenigstens ein reproduzierbares, *situationsabhängiges* Ergebnis zu dokumentieren. Bereits eine durchschnittliche Laborumgebung mit mehreren Lichtquellen, Gegenständen unterschiedlicher optischer Eigenschaften, Schattierungen und Reflexionen ist exakt kaum zu erfassen. Tageslicht mit seiner großen Dynamik, der stark gerichteten Sonnenstrahlung und entsprechenden Schlagschatten enthält im allgemeinen noch größere Variationen.

Trotz unbekannter Variationen ist es möglich, verschiedene Erkennungsverfahren zur Objekterkennung objektiv zu vergleichen. Dazu werden die Erkenner mit denselben Trainingsbildern trainiert und anschließend mit jeweils identischen Testbildern die Erkennungswahrscheinlichkeiten gemessen. Bei diesem Vorgehen wird aber nur das Erkennungsverfahren getestet. Ob die vorgegebenen Trainingsbilder für den praktischen Einsatz des Systems geeignet sind, wird auch mit diesem Test nicht geprüft.

Da bei dem hier vorgestellten Verfahren sehr große Trainingsmengen verwendet werden können, ist trotzdem ansichtsbasierte Objekterkennung in variabler Umgebung möglich; der Anwender hat allerdings die grundsätzliche Einschränkung zu beachten, daß die Objekterkennung besonders bei Tageslicht nicht deterministisch ist. In Abschnitt 4.1.2 bis Abschnitt 4.1.4 werden - teilweise heuristische - Verfahren angegeben, wie ein geeignetes Trainingsset aus einer Aufgabenstellung erzeugt werden kann.

4.1.2 Erzeugen der Trainingsmenge

Für die Auswahl der Trainingsbilder der Objektklasse können folgende Regeln angegeben werden:

- Einschränken der Variablilität. Aus der Anwendung heraus sollte die Variabilität soweit eingeschränkt werden wie möglich, beispielsweise durch Einsatz einer künstlichen, konstanten Beleuchtung und Vermeiden vieler geometrischer Freiheitsgrade.
- Geometrische Freiheitsgrade. Die geometrischen Freiheitsgrade der relativen Lage des Objekts zur Kamera sollen gleichmäßig gerastert werden. An jedem Rasterpunkt wird ein Beispielbild aufgenommen[1], das für eine Umgebung dieses Punktes repräsentativ ist.
- Beleuchtung des Objekts. Bei Änderungen der Beleuchtung ist bereits nicht mehr von Freiheitsgraden im klassischen Sinn zu sprechen, da dort oft nicht nur die Helligkeitsstärke einer bestimmten Lichtquelle variiert wird, sondern sich auch die Struktur der Beleuchtungsquelle ändert. Das gilt besonders für das Serviceszenario, das durch seine Vielfalt auch an flächigen Beleuchtungsquellen wie Fenstern, diffuser Reflexion an Gegenständen des täglichen Gebrauchs und wechselnden Situationen wie sonnig/bewölkt kaum modellierbar ist. Trotzdem kann eine Einteilung vorgenommen werden, beispielsweise in Kunstlicht, Tageslicht, Beleuchtung von rechts, links, usw. Die unterschiedlichen Fälle sind jeweils wie Rasterpunkte geometrischer Freiheitsgrade zu behandeln.
- Markieren des Objekts. Innerhalb der Beispielbilder für das Objekt wird die Position des Objekts vom Anwender markiert; an der markierten Stelle wird ein Muster der entsprechenden Größe ausgeschnitten und als Objektmuster abgespeichert. Der Anwender muß dafür Sorge tragen, daß er stets den selben Punkt auf der Objektoberfläche markiert; dieser Punkt wird später im Bild als ´Objekt´ erkannt. Außerdem wird durch eine Markierung immer des gleichen Punktes auf dem Objekt die Variabilität der Objektmuster eingeschränkt und dadurch das Training und die Erkennung beschleunigt.

1 Durch die Suche im 2-dimensionalen Bild werden bei der Erkennung 2 zusätzliche Freiheitsgrade abgedeckt, die aber für das Training irrelevant sind.

Die Erzeugung der Zurückweisungsklasse ist an sich sehr einfach, es müssen lediglich Bilder *ohne* das gesuchte Objekt aus der typischen Umgebung aufgenommen werden; aus diesen Bildern werden alle Muster der entsprechenden Größe herausgeschnitten. Beispielsweise ergeben sich bei einem Bildformat von 384x288 Pixel ca. 100000 Muster, in 300 Bildern zusammen ca. 30 Mio. Muster. Diese Muster beinhalten eine sehr große Menge unterschiedlichster Objekte, zufälliger Objektanordnungen und Ausschnitte größerer Gegenstände, die für die Szene typisch sind.

4.1.3 Iterative Vervollständigung und Validierung der Trainingsmenge

Nachdem also über die notwendigen Trainingsmuster keine systematischen Aussagen anhand der Problemstellung getroffen werden können, ist ihre Vollständigkeit experimentell festzustellen. Dazu wird ein trainierter Klassifikator auf eine festgelegte Testmenge angewandt. Die Fehlerhäufigkeit ist dann ein Maß für die Güte des Klassifikators, wobei zwei Fehler zu unterscheiden sind. Zum einen kann ein vorhandenes Objekt nicht erkannt werden, zum anderen kann ein anderes Muster als das gesuchte Objekt identifiziert werden.

Ein Nicht-Erkennen des Objekts bedeutet ein fehlendes Bild der Objektklasse, ein Erkennen des Hintergrundes als Objekt bedeutet eine zu kleine Zurückweisungsklasse. Daraus läßt sich eine iterative Methodik zur Generierung einer vollständigen Trainingsmenge ableiten. Ausgehend von einem bestimmten Trainingsset wird ein Klassifikator erstellt und anhand einer realen Szene getestet. Bilder mit Nicht-Erkennen des Objekts werden gespeichert, die Objekte manuell identifiziert und der Objekt-Trainingsmenge hinzugefügt. Bilder mit einer Erkennung eines falschen Musters als Objekt werden ebenfalls gespeichert und der Zurückweisungsklasse hinzugefügt. Anschließend wird aus dem erweiterten Trainingsset ein neuer Klassifikator generiert und das Vorgehen wiederholt. Nach 1 bis maximal 2 Iterationen geht die Fehlerhäufigkeit praktisch gegen 0, der fehlerfreie Fall läßt sich allerdings auch mit diesem Verfahren nicht garantieren.

Eine Methode zur Identifizierung schlecht trainierter Situationen anhand des Trainingsmaterials, also ohne einen Test mittels zusätzlichem Bildmaterial, besteht in der Analyse des Musterbaums. Ist eine Situation durch zu wenige Objektbilder repräsentiert, werden die entsprechenden Endknoten des Musterbaums nur durch jeweils wenige Trainingsansichten des Objekts erzeugt. Dies ist ein Indiz, daß in der betreffenden Situation zusätzliche Objektansichten benötigt werden. Allerdings liefert auch diese Methode nur einen qualitativen Hinweis auf zu wenige Ansichten und keine Aussage, wieviele und besonders welche zusätzlichen Ansichten tatsächlich nötig sind.

4.1.4 Künstliche Erweiterung der Trainingsmenge

Neben der Aufnahme zusätzlicher Bilder läßt sich die Trainingsmenge auch künstlich erweitern. Dazu werden bereits vorhandene Bilder unterschiedlichen Transformationen unterworfen:

- Um unterschiedliche Entfernung zum Objekt zu simulieren, kann die Größe des Musters verändert werden und damit eine größere oder kleinere Entfernung zum Objekt simuliert werden.
- Um eine Drehung von Objekt oder Kamera um die Verbindungsgerade Objekt-Kamera auszugleichen, kann das Muster in der Bildebene gedreht werden.
- Eine wechselnde Beleuchtung kann simuliert werden, indem die Grauwerte mit einem Faktor multipliziert werden oder indem ein Offset addiert wird.

Für die nichttrivialen, geometrischen Transformationen stehen leistungsfähige Bibliotheken zur Verfügung. Die 3 Transformationen modellieren die zugehörigen Situationsänderungen exakt, lediglich durch die Diskretisierung muß der tatsächliche Zusammenhang approximiert werden. Daneben können Transformationen angegeben werden, die beispielsweise eine Drehung des Objekts um eine Achse senkrecht zur Geraden Objekt-Kamera annähern. Diese Drehung läßt sich durch eine Skalierung in nur einer Richtung erreichen, gilt aber nur näherungsweise für ebene Objekte, die im Vergleich zur Entfernung eine kleine Ausdehnung haben.

Eine sehr detaillierte Beschreibung des Verfahrens findet sich bei Hagar, [26]; es ist Grundlage der dort beschriebenen Methode zur Objektverfolgung. Da das Verfahren verhältnismäßig einfach und robust ist, wird es in [43] bereits bei der Objekterkennung eingesetzt.

Eine weitere, von Nayar in [43] veröffentlichte Methode geht von 2 Objektansichten aus, die in kleinem zeitlichen und örtlichen Abstand voneinander gewonnen worden sind. Es wird angenommen, daß eine Mittelung der beiden Ansichten mit unterschiedlicher Gewichtung wieder zu einer Objektansicht führt, wenn beide Aufnahmen ähnlich sind. Repräsentieren $\underline{x}_1$ und $\underline{x}_2$ die beiden Objektansichten, jeweils in Vektoren sortiert, gilt für die zusätzliche, interpolierte Ansicht $\underline{x}_\gamma$:

$$\underline{x}_\gamma = \gamma \cdot \underline{x}_1 + (1-\gamma) \cdot \underline{x}_2 \qquad \gamma \in [0, 1]$$

Allgemeingültige Grenzen für den zeitlichen und örtlichen Abstand zwischen den Aufnahmen können nicht angegeben werden. Auch eine Extrapolation kann durchgeführt werden, γ liegt dann außerhalb des Intervalls $[0, 1]$. Maximale Grenzen für γ lassen sich dafür ebenfalls nicht angeben, sondern allenfalls experimentell bestimmen.

Neben der linearen Inter- und Extrapolation zwischen zwei Aufnahmen können die Aufnahmen auch künstlich verrauscht werden, zu den Pixeln werden zufällige, beispielsweise normalverteilte Zahlenwerte addiert. Dies erzeugt eine robustere Klassifikation, da neben den originalen Objektansichten auch ein zusätzlicher Raum um diese als Objekt-Wolke gebildet wird. Als Richtwert für die Amplitude des Rauschens kann die Amplitude des Pixelrauschens der Kamera verwendet werden; Versuche legen einen etwas größeren Wert nahe, auch hier kann keine allgemeingültige Grenze angegeben werden.

Das Rauschen kann durch zweierlei Methoden simuliert werden. Zum einen können durch Verrauschen der Vektoreinträge neue Vektoren gebildet und der Trainingsmenge hinzugefügt werden. Zum anderen kann das Rauschen auch dadurch simuliert werden, daß bei der numerischen Entropieoptimierung aus Abschnitt 2.4 die $sgn(x)$ -Funktion durch eine Sigmoid-Funktion angenähert wird. Dies kann nicht nur als Verflachen der $sgn(x)$-Funktion interpretiert werden, sondern auch als Verrauschen des Arguments. Für die Optimierung bedeutet dies, daß der Parameter η während der Optimierung erhöht wird entsprechend dem Algorithmus nach Abschnitt 2.4.3, aber nur bis zu einer maximal vorgegebenen Grenze. Diese Grenze wird so bemessen, daß die resultierende Wahrscheinlichkeitsdichteverteilung in etwa dem Pixelrauschen der Kamera entspricht.

Die vorgestellten Verfahren zur Generierung künstlicher Muster lassen sich auch kombinieren. Allerdings sind die Verfahren nur mit Vorsicht anzuwenden, da lediglich Heuristiken für bestimmte Einzelfälle, kaum aber systematische Parametrierungen für die unterschiedlichen Methoden angegeben werden können. In den meisten Fällen wird deshalb die Aufnahme einer größeren Menge an Bildern bessere Resultate liefern als eine kleine, nachträglich nach Abschnitt 4.1.4 erweiterte Menge. Dies gilt für Objekt- und Zurückweisungsklasse. Letztere kann prinzipiell auch künstlich vergrößert werden, doch ist hier die Aufnahme zusätzlicher Bilder fast immer einfacher. Die Erzeugung künstlicher Muster ist nur dann sinnvoll, wenn die Aufnahme neuer Objektansichten sehr aufwendig ist.

4.2 Praktischer Einsatz der Musterbäume

4.2.1 Unterabtastung

Um das oben beschriebene Klassifikationsverfahren auf die Suche eines Objekts in einem Bild anzuwenden, wird ein Suchfenster mit der Größe der Muster über das Bild geschoben. Nach jeder Verschiebung wird der Inhalt des Suchfensters mit dem Musterbaum als Objekt oder Nicht-Objekt klassifiziert. Der Aufwand dieses Verfahrens ist relativ groß, da an jeder Position des Bildes, beispielweise an jedem Pixel eine

Klassifikation durchgeführt wird. Allerdings ist der Aufwand für dieses Vorgehen mehr zufällig durch die Auflösung der Kamera bestimmt als durch das Problem selbst. Entsprechend muß die Auflösung künstlich auf das notwendige Maß reduziert werden, woraus sich dann die für den praktischen Einsatz notwendige Geschwindigkeitssteigerung ergibt.

Eine naheliegende Methode ist eine Steuerung der Musterauflösung in Abhängigkeit von der Baumtiefe. Es ist offensichtlich, daß in den ersten Schichten des Baums nicht die detaillierte Struktur der Muster relevant ist und dementsprechend nicht die volle Bildauflösung notwendig ist. Statt dessen zählt dort für die grobe Einordnung des Musters besonders seine gesamte Helligkeit und allenfalls die grobe Struktur. Mit zunehmender Schichttiefe innerhalb des Musterbaums werden an einen bestimmten Knoten nur ähnliche Muster gelangen, die anschließend durch weitere Detaillierung und damit zusätzliche Information voneinander unterschieden werden.

Für die meisten Objekte gilt, daß die hochfrequenten Anteile im Ortsspektrum der Muster beider Klassen eine große Variabilität haben, insbesondere die hochfrequenten Anteile der Zurückweisungsklasse. Dementsprechend überlappen sich diese stark und tragen wenig Information[1]. Um den hochfrequenten Anteil der unterabgetasteten Muster zu reduzieren, werden die Rohbilder zunächst gefiltert. Dies erfolgt zweckmäßiger Weise mit einem Tiefpaßfilter. Wir erhalten bei drei unterschiedlichen Auflösungen die Struktur nach Abb. 4.1.

Als wesentliche Konsequenz der Unterabtastung ergeben sich zwei Effekte. Zum einen wird die benötigte Rechenzeit durch die geringere Anzahl der ausgewerteten Pixel reduziert. Beispielsweise ergibt sich bei zweimaliger Unterabtastung um den Faktor 3 eine Reduktion der Pixelanzahl in einem Muster um den Faktor 81. Zum anderen wird sich die Mehrzahl der in einem Bild klassifizierten Muster von dem gesuchten Objekt sehr stark unterscheiden und deshalb in einer niedrigen Schichttiefe bei bereits sehr geringer Auflösung als Nicht-Objekt identifiziert. Da dies vor allem Zurückweisungsmuster sind, ergibt sich also der doppelte Vorteil, daß viele Muster nach nur wenigen Schichten klassifiziert werden und daß dies bei sehr geringer Auflösung erfolgt. Dies entspricht der Vorstellung, daß sich Bildbereiche, die mit dem gesuchten Objekt nur eine geringe Ähnlichkeit besitzen, schnell zurückweisen lassen, beispielsweise eine helle Fläche, wenn ein dunkler Gegenstand gesucht wird.

Die oben beschriebene Tiefpaßfilterung und Unterabtastung stellt natürlich einen Zusatzaufwand dar, der dem Ziel der Geschwindigkeitssteigerung entgegensteht. Andererseits müssen diese Operationen nur *einmal für das gesamte Bild* durchgeführt wer-

1 Dies ist nur ein Erfahrungswert für unsere Anwendungen; grundsätzlich könnte die wesentliche Information auch ausschließlich in den hochfrequenten Anteilen enthalten sein.

den und nicht in den einzelnen Analysefenstern. Da sich die Analysefenster überlappen, ist der Aufwand außerordentlich gering verglichen mit der eigentlichen Klassifikation.

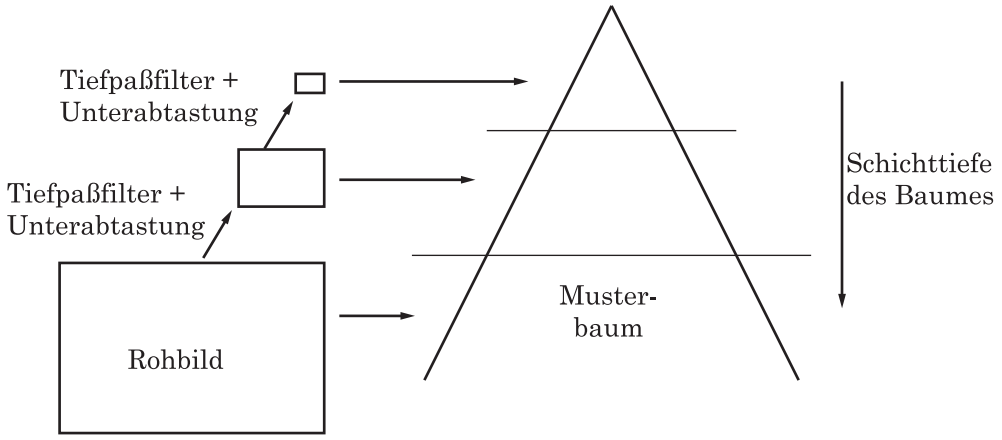

ABBILDUNG 4.1: Tiefpaßgefilterte Eingangsbilder für unterschiedliche Schichten des Musterbaums.

Neben der Unterabtastung eines einzelnen Musters kann eine ähnliche Methode auch auf den Abstand zweier Suchfenster angewendet werden. Wird die Auflösung eines Musters um einen bestimmten Faktor reduziert, ist ein Suchfensterabstand naheliegend, der diesem Faktor entspricht. Dies führt zu dem Problem, daß während einer Klassifikation, also innerhalb eines Musterbaums an einer bestimmten Schicht *zusätzliche Analysefenster* eingeführt werden. Zur genaueren Diskussion diene Abb. 4.2. Von der Schicht n zur Schicht $n+1$ sollen zusätzliche Suchfenster eingefügt werden, die den Bereich feiner untergliedern. Dies impliziert, daß ein Knoten in Schicht n erreicht worden ist, der kein Endknoten ist. Nun werden an der Stelle des alten Analysefensters und an den 8 benachbarten, neu eingefügten Analysefenstern insgesamt 9 Klassifikationsvorgänge gestartet, die alle mit dem bereits erreichten Knoten aus Schicht n beginnen. Der auf diesen Knoten folgende Teilbaum wird also als Musterbaum insgesamt 9 mal an der alten und den neuen Positionen verwendet. Somit wird das bereits erzielte Resultat, die Eingrenzung auf den betreffenden Knoten, auf die neuen Analysefenster angewandt.

Das Verfahren entspricht dem Verfahren der reduzierten Auflösung und wird selbstverständlich mit diesem kombiniert. Die Einsparung an Rechenzeit ist enorm; bei einer Unterabtastung um den Faktor r und einer reduzierten Auflösung um ebenfalls den Faktor r erhält man als Gesamtfaktor für die Verringerung des Rechenaufwands r^4. Bei $r = 9$ in den ersten Schichten ergibt sich also eine Einsparung um den Faktor 6561. Dies verdeutlicht die Wirksamkeit dieser Verfahren für Echtzeitanwendungen.

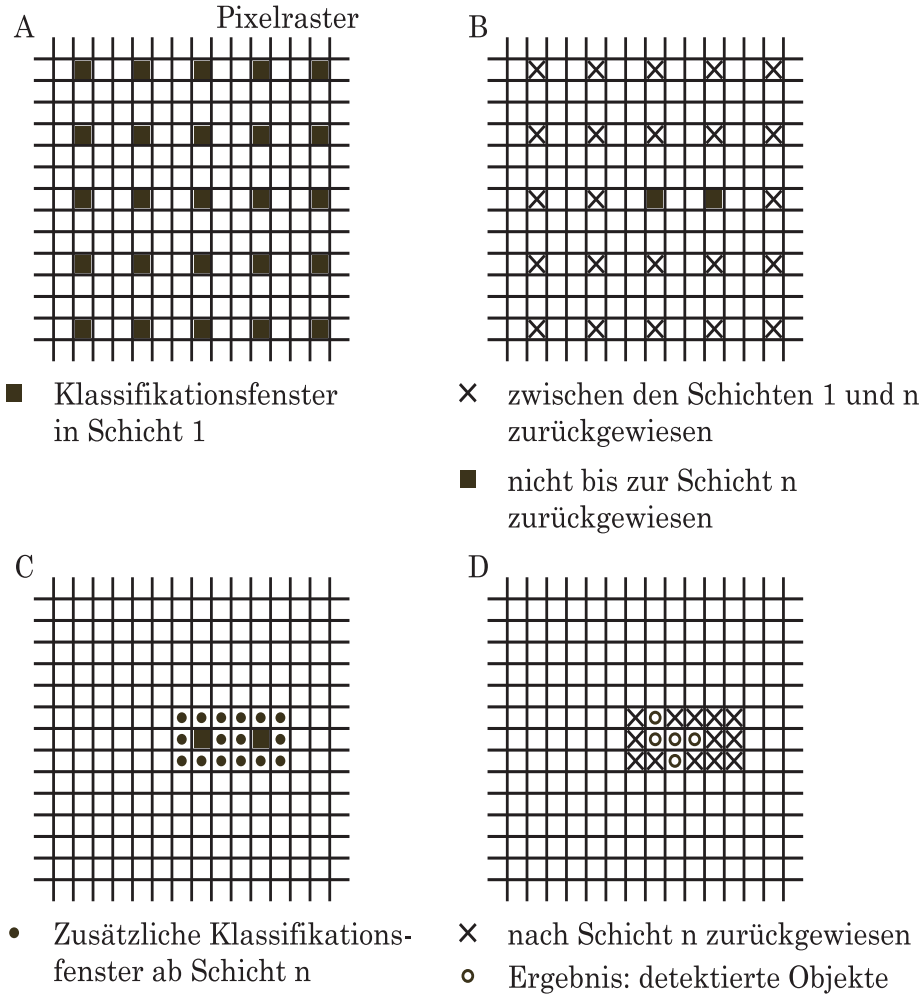

ABBILDUNG 4.2: Einfügen zusätzlicher Suchfenster, die Abbildungen A, B, C und D geben die einzelnen Schritte an.

Es stellt sich die Frage, wie stark die Unterabtastung in jeder Schicht des Baums sein darf. Eine zu starke Unterabtastung reduziert einerseits die erforderliche Rechenleistung. Andererseits wird die Klassifikation anhand einer kleineren Informationsmenge weniger robust. Würden die Vektoren beispielsweise anhand von nur 9 Pixel klassifiziert, könnten bei 255 Quantisierungsstufen theoretisch $255^9 = 4,5 \cdot 10^{21}$ Muster unterschieden werden. Trotzdem wären die 9 Pixel nicht repräsentativ und die Klassifikation nicht robust, obwohl die Trainingsvektoren getrennt werden können. Die Entropieoptimierung muß eine ausreichende Menge von Pixeln zur Verfügung haben, die sie geeignet gewichten kann.

Dementsprechend führt eine geringere Unterabtastung zu einer Vergrößerung der extrahierten Information, der Gewinn ist allerdings beschränkt; in der Praxis muß ein experimentell zu validierender Mittelweg gefunden werden, wobei die Auflösung solange verbessert wird, bis die Zunahme der extrahierten Information nicht mehr

nennenswert ist. Damit dies nicht für jeden Knoten im Baum getrennt durchgeführt werden muß, erhalten alle Knoten einer Schicht die selbe Auflösung. Die Reduktion der Auflösung wird heuristisch[1] festgelegt zu

$$r = o + q \cdot s,$$

wobei o und q Konstanten sind und s die Schichtnummer. Für $r \leq 1$ wird r zu 1 gesetzt. Die beiden Parameter werden nun so eingestellt, daß sich in der 1. Schicht Muster mit $1x1$ ergeben und die maximale Auflösung in der ca. 20. Schicht erreicht wird. Diese Werte sind experimentell ermittelt; sie sind jedoch nicht besonders kritisch, eine geringfügige Veränderung, beispielsweise eine Mustergröße von $3x3$ in der 1. Schicht, hat kaum eine systematische Veränderung für die gesamte Objekterkennung zur Folge.

4.2.2 Mehrfacherkennung

Der Musterbaum legt einen bestimmten Teilraum des Pixelraumes als Objektbereich fest. Für die Erkennung des Objekts müssen die Pixel eines Suchfensters innerhalb dieses Musterraumes liegen. Dies ist für ein Objekt im Bild nicht nur für ein einziges Suchfenster der Fall, sondern meist in einer kleinen Umgebung entsprechend Abb. 4.2 D. Das Objekt wird also innerhalb eines kleinen Bildbereichs mehrmals erkannt. Die Größe dieses Bereichs hängt zum einen davon ab, ob das Objekt bereits an der Grenze zur Erkennbarkeit ist und durch eine kleine Veränderung den Objektbereich des Erkenners verläßt. Zum anderen hängt die Größe dieses Bereichs davon ab, wie hochfrequent das Bild in dem betreffenden Bereich ist und wie stark sich somit der Inhalt des Suchfensters bei Verschiebung ändert.

Auch hinsichtlich der irrtümlichen Erkennung anderer Objekte als dem gesuchten Objekt spielt dieser Effekt eine Rolle. Wird ein anderes Muster mit dem gesuchten Objekt verwechselt, ist es in den meisten Fällen an der Grenze, noch als Objekt erkannt zu werden. Entsprechend wird es nur in einem sehr kleinen Gebiet als solches erkannt, die Anzahl der Mehrfacherkennungen ist vergleichsweise klein.

Der Effekt dieser Mehrfacherkennungen ermöglicht eine einfache Bewertung einer Erkennung anhand der Anzahl der Erkennungen. Wurde ein Objekt in einer kleinen Umgebung sehr oft detektiert, ist die Erkennung offensichtlich sehr robust. Eine Mehrfacherkennung bedeutet hier, daß das Objekt noch relativ gut erkannt werden kann. Eine geringe Anzahl von Erkennungen läßt darauf schließen, daß das Muster an

1 Die hier vorgestellte, heuristische Steuerung der Unterabtastung könnte beim Training der Musterbäume systematisiert werden. Beispielsweise könnte die Unterabtastung in jedem Knoten des Musterbaumes so eingestellt werden, daß im Vergleich zur maximalen Auflösung 5% Information verloren geht. Ein solches systematisches Verfahren würde zwar das Verhalten bei der Erkennung verbessern, allerdings ein mehrfaches an Trainingszeit kosten. Um die Trainingszeit nicht zu vergrößern, erscheint der heuristische Ansatz gerechtfertigt.

der Grenze des noch als Objekt definierten Bereichs liegt. Das Objekt ist dementsprechend entweder an der Grenze des durch die Trainingsmenge definierten Objektbereichs oder es ist ein dem Objekt ähnliches Muster anderen Ursprungs. Die Anzahl der Mehrfacherkennungen ist folglich eine Art Maß der Robustheit und kann für die weitergehende Verarbeitung des Erkennerergebnisses verwendet werden, wie in Abschnitt 5.3 eingehend diskutiert.

4.2.3 Partielle Verdeckung

Ein in der Praxis wichtiges Kriterium für die Robustheit eines Algorithmus zur Objekterkennung ist seine Robustheit gegenüber partieller Verdeckung des Objekts durch ein anderes Objekt. Verdeckung kann zunächst durch den Prozentsatz des Musters charakterisiert werden, der verdeckt wird. Des weiteren hängt das Problem aber auch davon ab, welcher Teil des gesuchten Objekts verdeckt wird und auch von dem Objekt im Vordergrund.

Zunächst werden von dem Objekt im Vordergrund die Pixelwerte an den betreffenden Stellen im Bild geändert. Dies wirkt sich auf jedes Skalarprodukt im Musterbaum aus, und damit auf die einzelnen binären Entscheidungen. Die Veränderung des Skalarprodukts ist um so größer, je mehr Pixel in ihrem Wert verändert werden und je größer die Änderungen in den einzelnen Pixeln sind. Darüber hinaus hängt die Veränderung auch von der unterschiedlichen Gewichtung der Pixel ab. Aus diesen Überlegungen folgt, daß eine kleine Störung, von der nur wenige Pixel betroffen sind, für die ansichtsbasierte Objekterkennung meist kaum Auswirkungen hat. Ob das Objekt noch erkannt werden kann, hängt in diesem Fall davon ab, ob es ohne Verdeckung bereits an der Grenze des Erkennungsbereiches war. Des weiteren hängt die Wirkung einer Verdekkung davon ab, welcher Teil des Objekts verdeckt wird. Die Verdeckung eines für die Erkennung sehr wichtigen Teils des Objekts führt schneller zu einer Fehlklassifikation als die Verdeckung eines weniger wichtigen Teils, allerdings ist nur selten bekannt, welche Teile das sind. Experimentell kann leicht gezeigt werden, daß auch der Helligkeitsunterschied zwischen dem verdeckten Objektbereich und dem Objekt im Vordergrund eine wesentliche Rolle spielt.

Die Verdeckunsproblematik im ganzen kann zwar nach obigen Gesichtspunkten strukturiert werden und es können auch qualitative Aussagen getroffen werden, allerdings erscheint eine genaue, quantitative Analyse kaum möglich. Diese würde zumindest ein Modell für das verdeckende Objekt erfordern.

4.2.4 Farbbildverarbeitung

Der bisher vorgestellte Algorithmus zur Objekterkennung ist selbstverständlich auch für die Analyse von Farbbildern geeignet. Die wesentliche Programmodifikation besteht in einer Vergrößerung des Mustervektors um den Faktor 3[1]. Neben den Helligkeitswerten der Pixel müssen nun pro Pixel zwei zusätzliche Werte ausgewertet werden, die die Farbe repräsentieren. Der zusätzliche Berechnungsaufwand ist also um einen Faktor 3 größer. Da zum Erhalt der Echtzeitfähigkeit nicht mehr Werte verarbeitet werden können, werden um einen Faktor 3 weniger Pixel betrachtet. Die Unterabtastung nach Abschnitt 4.2.1 muß also stärker ausfallen.

Ob zusätzlich zur Helligkeitsinformation Farbinformation verwendet werden soll, ist eine nicht triviale Entscheidung. Im allgemeinen wird die Entscheidung gegen die Farbinformation ausfallen. Ob die Farbinformation wesentlich zur Problemlösung beiträgt, hängt nicht nur von der Objektklasse, sondern auch von der Zurückweisungsklasse ab. In bestimmten Anwendungen wird die Farbinformation sicherlich eine wichtigere Rolle spielen als bei anderen.

Bei eigenen Versuchen zur Objekterkennung in Farbbildern und in Echtzeit wurde keine signifikante Verbesserung festgestellt. In vergleichenden Versuchen mit und ohne Farbe waren die Unterschiede in der Erkennungswahrscheinlichkeit selbst dann sehr gering, wenn die selbe Anzahl von Pixeln verwendet wurde. Als Konsequenz ergab sich eine geringere Geschwindigkeit der Farbbildverarbeitung im Vergleich zur Graubildverarbeitung. Der Versuch ist nicht repräsentativ, da lediglich ein bestimmtes Objekt verwendet wurde. Allerdings zeigt der Versuch auch, daß Helligkeitsinformation im Vergleich zur Farbinformation sicher sehr wichtig ist.

In einem anderen Versuch wurden zwei Objekte[2] mit identischer Geometrie, aber unterschiedlicher Einfärbung verwendet. Ansichten des einen Objekts wurden als Objektklasse verwandt, Ansichten des anderen wurden in die Zurückweisungsklasse aufgenommen. Auch hier konnten mit und ohne Farbinformation ähnliche Erkennungswahrscheinlichkeiten erreicht werden. Offensichtlich ist selbst bei *identischer* Form der Objekte die Helligkeitsinformation vergleichsweise wichtig.

4.3 Bestimmung von Objektparametern

Neben dem rein binären Klassifikationsresultat sind mit dem Musterbaum auch Parameter aus dem Bild extrahierbar; die Methode wurde anhand der Entfernungsbestimmung getestet und validiert. Ein Objekt wird je nach aktueller Ansicht durch verschie-

1 Auch die Verarbeitung von Bildern mit mehr als 3 Farbkanälen ist möglich.

2 Zwei elektrische Kondensatoren gleicher Geometrie, aber von unterschiedlichen Herstellern und deshalb unterschiedlicher, farbiger Aufdrucke. Aus drucktechnischen Gründen können keine farbigen Ansichten abgebildet werden.

dene Endknoten im Baum identifiziert. Bestimmte Objekt-Endknoten sind also für bestimmte Situationen zuständig, es gibt beispielsweise Endknoten, die das Objekt in großer Entfernung aufnehmen, oder Endknoten, die nur bei bestimmten Beleuchtungssituationen verwendet werden. Es liegt also nahe, aus dem Endknoten auf die Situation zurückzuschließen, die gerade vorliegt. Dazu muß die Situation beim Training erfaßt und als Attribut in dem betreffenden Endknoten gespeichert werden. Wurden also beim Training in einem bestimmten Endknoten nur Objektansichten aus einem bestimmten Entfernungsbereich aufgenommen, kann dieser Entfernungsbereich in dem Knoten gespeichert werden. Wird in der Erkennungsphase das Objekt durch diesen Knoten erkannt, kann zurückgeschlossen werden, daß sich das Objekt in dem im Knoten abgespeicherten Entfernungsbereich befindet.

In den Knoten können nicht nur kontinuierliche Größen wie die Entfernung oder die Drehung des Objekts gespeichert werden, sondern auch diskrete Größen wie ´Zusatzbeleuchtung ein/aus´. Der zusätzliche Rechenaufwand ist außerordentlich gering und beschränkt sich auf das Speichern der Information beim Training und das Auslesen der Information bei der Erkennung. Die musterbaumbasierte Objekterkennung ermöglicht also wenigstens teilweise die Bestimmung von Parametern bei der Objekterkennung.

Das Verfahren beeinflußt einerseits weder Training noch Erkennung, hängt aber andererseits stark von der Baumstruktur ab. Wird durch sehr unterschiedliche Objekt- und Zurückweisungsklassen der Baum sehr klein und erhält er durch geeignete Optimierung nur wenige Objekt-Endknoten, werden die situationsabhängigen Parameter nur sehr grob aufgeteilt. Für die Parameterbestimmung ist es also durchaus erwünscht, wenn die Objekt-Trainingsmenge in möglichst viele Fragmente aufgeteilt wird, denen damit jeweils ein eigener Parameter zugeordnet werden kann. Eine erfolgreiche Erkennung des Objekts ist allerdings wichtiger als eine genaue Parameterbestimmung.

Das Verfahren ersetzt nicht eine, für manche Anwendungen notwendige, präzise Bestimmung unterschiedlicher Parameter. Die Genauigkeit kann nicht vorgegeben werden. Es ist lediglich ein Nebenprodukt der musterbaumgestützten Objekterkennung, das durch den geringen Zusatzaufwand interessant ist.

5 Objektverfolgung in Videosequenzen

5.1 Einleitung

Die bisher betrachtete Objekterkennung bezog sich auf die Erkennung eines Objekts in einem einzelnen Bild. Für viele Anwendungen, besonders im Robotikbereich, ist dies nicht ausreichend, die Erkennung und Lokalisierung muß quasi-kontinuierlich und in Echtzeit auf eine *Bildfolge* angewandt werden. Es ergeben sich 2 neue Aufgabenstellungen:

- Korrespondenzproblem

  Werden 2 Vertreter eines Objekttyps in 2 aufeinanderfolgenden Bildern erkannt, ist zunächst nicht eindeutig, welche Objektabbildung zu welchem Objekt gehört. Das Problem wird vergrößert durch mögliche Nicht-Erkennung der Objekte und durch Fehl-Erkennungen anderer Objekte. Das Korrespondenzproblem ist nicht nur für 2 Bilder zu lösen, sondern für eine Bildfolge.

- Prädiktion

  Bei der Objekterkennung in einer Bildfolge kann die Position eines Objekts in einem neuen Bild $i+1$ prädiziert werden, wenn seine Trajektorie in den vergangenen Bildern $\ldots, i-2, i-1, i$ bekannt ist. Eine Schätzung der Position schränkt den Suchbereich für das Objekt ein, statt einer Suche im Vollbild läßt sich die Suche auf ein kleines Fenster (region of interest) beschränken. Die Prädiktion ermöglicht damit eine sehr schnelle Objektverfolgung.

Die Lösung jedes der beiden Probleme setzt jeweils die Lösung des anderen Problems voraus. Erst die Prädiktion einer Objektposition ermöglicht, die Position in den vergangenen Bildern mit der Position in einem neuen Bild zu vergleichen. Aufgrund der Objektbewegung ist eine Zuordnung einer Messung im Bild $i+1$ mit der letzten, direkt gemessenen Position im Bild i zumindest bei großen Objektgeschwindigkeiten nicht möglich. Alte und neue Position in Bildkoordinaten weichen stark voneinander ab und sind nur nach einer Prädiktion der alten Position vergleichbar.

Andererseits muß die korrekte Korrespondenz zwischen den Objekten aus den Bildern $\ldots, i-2, i-1, i$ bekannt sein, um die Objektpositionen in dem Bild $i+1$ prädizieren zu können.

Werden beide Aufgaben fehlerfrei gelöst, verschwinden wechselseitige Einflüsse. Prädiktion und Korrespondenzfindung können dann *voneinander unabhängig* betrachtet werden. Insbesondere ist also gefordert, daß die Objekterkennung das Objekt tatsächlich in der Nähe der prädizierten Position findet. Die optimale Suchstrategie wird in Abschnitt 5.4 diskutiert.

5.2 Prädiktion

Die Relativbewegung des Objekts gegenüber der Kamera soll hier in 2 unterschiedliche Kategorien eingeteilt werden:

- Bekannte deterministische Bewegung des Objekts oder der Kamera. Bei der Objekterkennung für das Fahrzeug ROMAN ist dies die Eigenbewegung des Fahrzeugs und die Drehbewegung des Kamerakopfes.
- Eine willkürliche, nicht vorhersehbare Bewegung, insbesondere des gesuchten Objekts wie beispielsweise bei der Gesichtsdetektion.

Beide Bewegungsarten sind oft überlagert. Andere Bewegungsarten wie beispielsweise statistisch beschreibbare Bewegungsformen werden hier nicht diskutiert, sie spielen für die hier betrachteten Anwendungsfälle keine Rolle.

5.2.1 Deterministische Bewegung

Die deterministische Bewegung ist vergleichsweise einfach zu behandeln. Aus der Relativbewegung kann eindeutig berechnet werden, wo sich das Objekt in einem neuen Bild befindet. Die Berechnung erfordert im allgemeinen die Kenntnis der Objektentfernung.

Für den rotatorischen Anteil der Kamerabewegung ist die Entfernungsinformation nicht notwendig. Dieser Fall ist insofern besonders wichtig, da eine Drehung der Kamera zu großen Bewegungen des Objekts im Bild führt.

5.2.2 Willkürliche Bewegung

Die willkürliche Relativbewegung, meist die Bewegung des Objekts, wird durch lineare Extrapolation aus den 2 vergangenen Bildern prädiziert. Da sich die lineare Prädiktion lediglich auf 2 vorangegangene Bilder stützt, ist sie besonders reaktiv und deshalb für große Beschleunigungen des Objekts im Bild geeignet. Die Position des Objekts in Bildkoordinaten berechnet sich zu

$$\underline{x}_{i+1} = 2 \cdot \underline{x}_i - \underline{x}_{i-1}$$

Da eine geradlinige, räumliche Bewegung des Objekts zu einer geradlinigen Bewegung der Objektabbildung in den 2D-Bildkoordinaten führt, kann die Prädiktion in Bildkoordinaten durchgeführt werden. Die Entfernungsinformation des Objekts wird nicht benötigt.

5.3 Korrespondenz

5.3.1 Zuordnung von Einzelerkennungen

Wird ein bestimmter Objekttyp in aufeinanderfolgenden Einzelbildern einer Videosequenz mehrmals erkannt, stellt sich die Frage, von wie vielen Objekten diese Erkennungen stammen und ob möglicherweise auch Fehl-Erkennungen enthalten sind. Dazu müssen Objekterkennungen aus verschiedenen Bildern einander zugeordnet und damit zeitlich fortgeschrieben werden. Diese Fortschreibung basiert auf einem Hypothesenspeicher variabler Größe, in dem alle in Frage kommenden, erkannten Objekte gespeichert werden. Wird in einem neuen Bild ein Objekt erkannt, wird diese Erkennung mit den bereits bekannten Objekthypothesen verglichen und gegebenenfalls einer Hypothese zugeordnet. Ist die neue Erkennung keiner Hypothese zuzuordnen, wird eine neue Hypothese angelegt.

Das Verfahren ermöglicht eine Trajektorienbestimmung einzelner Objekte bzw. eine dynamische Szenenanalyse. Es wird also nicht nur zu jedem Zeitpunkt die Position von Objekten eines bestimmten Typs ermittelt, sondern zusätzlich die Bewegung der einzelnen Objekte über die Zeit hinweg bestimmt. Insbesondere ist das Verfahren für die Prädiktion aus Abschnitt 5.2 notwendig, in der die Korrespondenz der Objekterkennungen in den vorangegangenen Bilder benötigt wird.

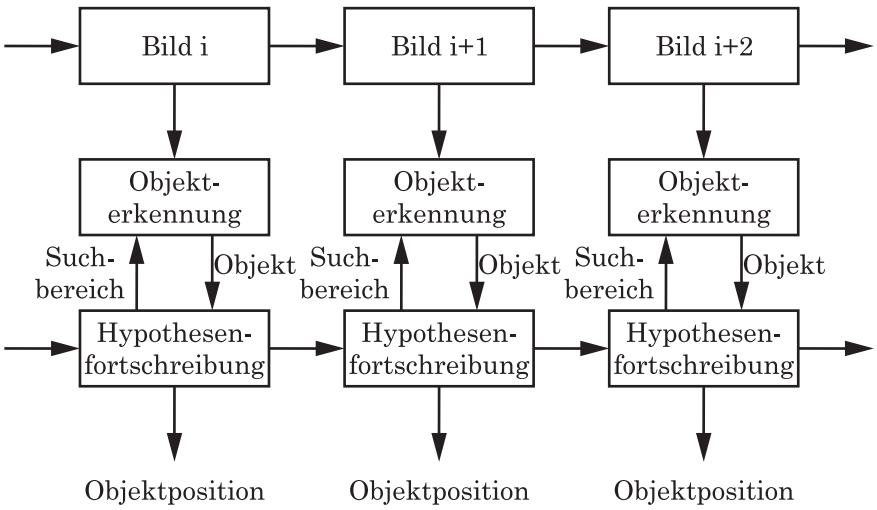

ABBILDUNG 5.1: Zeitlicher Ablauf der Hypothesenfortschreibung.

Die Zuordnung einer Hypothese zu einer aktuellen Messung erfolgt direkt in Pixelkoordinaten. Sind der prädizierte und der gemessene Objektort nahe beieinander, kann man von Identität der Objekte ausgehen. Dazu wird der euklidische Abstand zwischen Messung und prädizierter Hypothese berechnet und mit einem Schwellwert entschieden, ob Identität vorliegt oder nicht.

Da die Prädiktion gemäß Abschnitt 5.2 nur auf der Geschwindigkeit beruht, ergibt sich durch den nicht berücksichtigten Beschleunigungsterm eine Abweichung zu

$$e = a \cdot T_A^2/2$$

mit der Beschleunigung a in Bildkoordinaten und der Abtastzeit T_A. Für eine geringe Unsicherheit im Ort muß T_A möglichst klein gewählt werden, die Abtastzeit geht quadratisch in die Gleichung ein und wirkt sich dementsprechend stark aus; sie wird in Abschnitt 5.4 eingehend untersucht. Für die Festlegung des Schwellwerts für den euklidischen Abstand ergeben sich folgende Probleme. Zum einen ist die willkürliche Bewegung als Ursache des Fehlers numerisch nur schwer zu erfassen. Zum anderen hängt die Dimensionierung auch von der Häufigkeit des Objekttyps im Bild, der Anzahl der Fehl-Erkennungen und von deren durchschnittlichem Abstand zum Objekt ab. Ist beispielsweise das Objekt sicher nur einmal im Bild enthalten und können Fehl-Erkennungen ausgeschlossen werden, so ist die Schwelle entsprechend hoch zu setzen und jede Erkennung des Objekts der dann einzigen Objekthypothese zuzuordnen. In die Dimensionierung gehen also ein:

- Die Beschleunigung a,
- die Abtastzeit T_A,
- die Häufigkeit des Objekttyps im Bild und der typische Abstand zwischen zwei Objekten, und
- die Häufigkeit von Fehl-Erkennungen.

Die maximale Beschleunigung a geht in die Berechnung ein, sie wird direkt in Bildfolgen gemessen; dazu wird für jeden Objekttyp aus repräsentativen Videosequenzen ein Maximalwert bestimmt. Die Abtastzeit ist selbstverständlich exakt bekannt, somit läßt sich nach obiger Formel eine Schwelle berechnen.

Die Häufigkeit des Objekts im Bild und die Häufigkeit von Fehl-Erkennungen hängen von der jeweiligen Situation ab und lassen sich nicht allgemein angeben. Sind nur ein Objekt und wenige Fehl-Erkennungen zu erwarten, kann die Schwelle größer als berechnet gewählt werden. Die Beschleunigung darf dann entsprechend größer sein als

der gemessene Maximalwert. Für den Experimentalteil dieser Arbeit ermöglicht die sehr robuste Erkennung eine Vergrößerung der Schwelle um einen Faktor 2 im Vergleich zum berechneten Wert.

5.3.2 Nicht-Erkennung und Fehl-Erkennung

In der Praxis wird es vorkommen, daß beispielsweise durch Verdeckung die Erkennung des Objekts kurzzeitig nicht möglich ist. In diesem Fall kann man nicht entscheiden, ob das Objekt nicht mehr vorhanden ist oder nur kurzzeitig nicht erkannt wird. Die Hypothese wird deshalb nicht sofort gelöscht, sondern nur als *´nicht-aktuell´* gekennzeichnet. Sie wird jedoch weiterhin fortgeschrieben, also prädiziert und mit den aktuellen Messungen verglichen. Kann sie einer neuen Messung zugeordnet werden, wird sie wieder als *´aktuell´* gekennzeichnet. Kann die Hypothese innerhalb eines bestimmten Zeitraumes keiner Messung zugeordnet werden, wird sie gelöscht.

Neben der Kennzeichnung als ´aktuell´ oder ´nicht-aktuell´ wird zusätzlich ein Maß für die Robustheit eingeführt. Dazu wird gezählt wie oft das Objekt bereits erkannt wurde und zwar mit Berücksichtigung von Mehrfacherkennungen nach Abschnitt 4.2.2. Von diesem Zählmaß wird die Anzahl der Bilder ohne Erkennung des Objekts abgezogen. Je größer dieses Maß ist, desto robuster ist die Hypothese. Wird dieses Maß zu 0, wird die Hypothese gelöscht. Dieses Maß der Hypothese wird mit einer Schwelle verglichen und ergibt damit eine binäre Kennzeichnung aller Hypothesen als ´robust´ oder ´nicht robust´. Als tatsächlich erkannte Objekte werden nur diejenigen Hypothesen angesehen und als Resultat der Erkennung ausgegeben, die sowohl ´robust´ als auch ´aktuell´ sind. Daher folgt auch die Bezeichnung der Hypothesen als solche, da sie zunächst noch keine zuverlässige Erkennung des Objekts darstellen. Um bei längerer Erkennung eines Objekts einen über alle Maßen wachsenden Zählerstand zu verhindern, wird ein Sättigungswert für das Robustheitsmaß eingeführt.

Abb. 5.2 verdeutlicht den oben beschriebenen Zusammenhang. Die Kurve stammt aus einem Experiment, in dem sich das gesuchte Objekt zunächst aus großer Entfernung der Kamera nähert und anschließend seitlich aus dem Bildfeld verschwindet. Das Objekt wird ab einer bestimmten Entfernung in Bild i erkannt. Da es eine Zweifacherkennung ist, wird der Zähler auf den Wert 2 gesetzt. Es folgen eine Einfacherkennung im Bild $i + 1$ und eine Dreifacherkennung im Bild $i + 2$. Im Bild $i + 3$ wird das Objekt nicht erkannt und dementsprechend der Zähler um 1 verringert. In Bild $i + 4$ ist das Objekt nicht mehr an der Grenze zum Erkennungsbereich und wird deshalb bereits mit einer 8-fach-Erkennung erkannt, die jedoch aufgrund der Zählerbegrenzung nur zu einem Zählerstand von 10 führt. Im weiteren wird das Objekt mit einer Ausnahme stabil erkannt, bis es ab Bild $i + 11$ nicht mehr zu sehen ist.

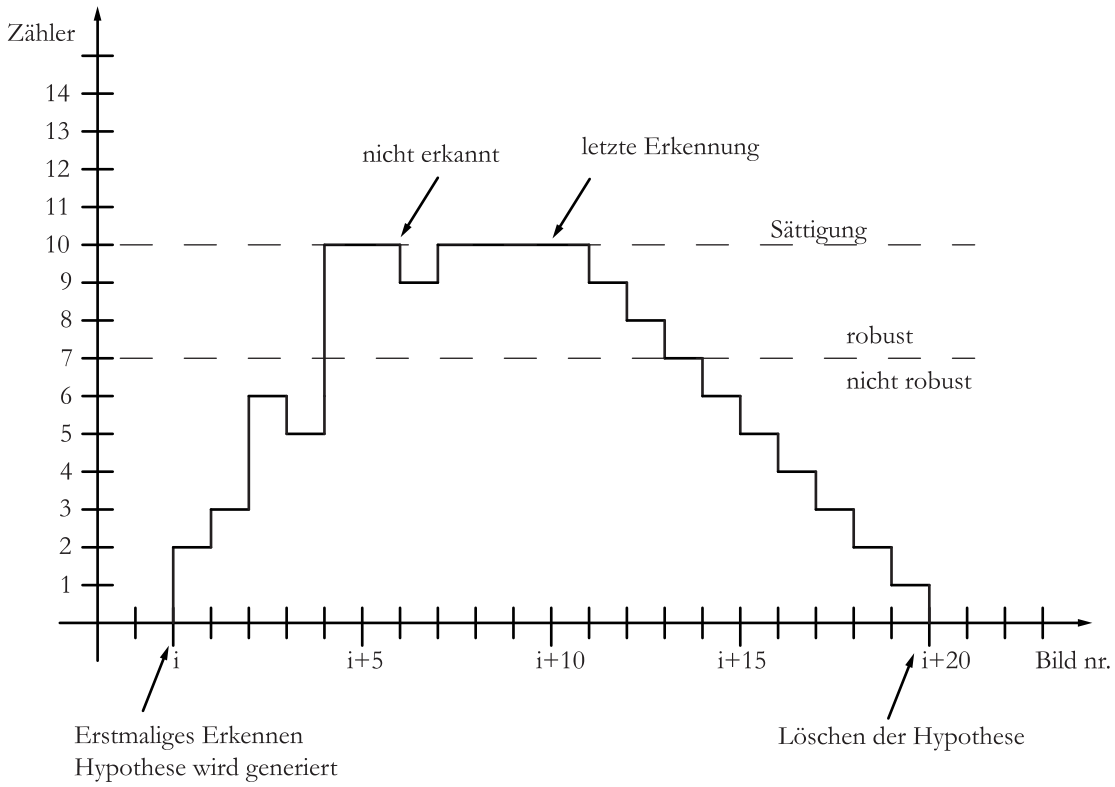

ABBILDUNG 5.2: Typischer Verlauf der ´Robustheit´.

Abb. 5.3 zeigt den typischen Verlauf einer Erkennung eines dem gesuchten Objekt ähnlichen Musters. Es wird in insgesamt 3 Bildern als das Objekt erkannt. Der Abstand zu der Schwelle für robuste Erkennung wird in keinem Fall überschritten und nach kurzer Zeit werden die Hypothesen wieder gelöscht.

Selbstverständlich ist statt der oben dargestellten, pragmatischen Methode auch eine exakte wahrscheinlichkeitstheoretische Betrachtung des Zusammenhangs möglich. Genauere Ergebnisse bedingen aber eine genaue Kenntnis der Parameter, hier insbesondere der kaum bekannten Wahrscheinlichkeiten für eine Fehl-Erkennung und Objekterkennung. Diese Wahrscheinlichkeiten sind *situationsabhängig* und damit nicht-ergodisch, sie lassen sich deshalb kaum genau erfassen. Dementsprechend sind von einer mathematisch korrekten Behandlung keine wesentlichen Verbesserungen zu erwarten. Darüber hinaus bleibt auch bei einer solchen Betrachtung das Problem des korrekten Einstellens des Schwellwerts offen.

Die Hypothesenfortschreibung kann durch einen geringen, zeitlichen Horizont Fehl-Erkennungen identifizieren und Nicht-Erkennungen des gesuchten Objekts überbrükken. Dies allerdings geschieht zu Lasten der Reaktionsgeschwindigkeit des Erkenners, d.h. eine einzelne Erkennung des Objekts kann nicht als solche gewertet werden, sondern es ist statt dessen die Analyse der folgenden Bilder abzuwarten. Die Erhöhung der Robustheit durch Hypothesenfortschreibung verringert also die Geschwindigkeit,

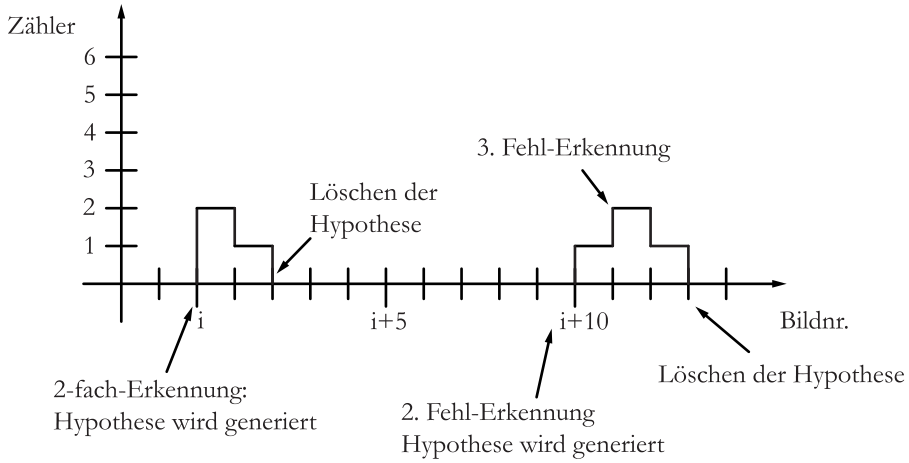

ABBILDUNG 5.3: Typischer Zählerverlauf bei einer Fehl-Erkennung.

beide Größen sind in bestimmten Grenzen austauschbar. Beides kann nur durch eine schnellere oder robustere Objekterkennung im Einzelbild verbessert werden, im Idealfall ist diese fehlerfrei und die Hypothesenfortschreibung dient nur der zeitlichen Zuordnung von Objekten aus unterschiedlichen Bildern der Videosequenz.

5.4 Optimale Abtastzeit und Suchfenster

5.4.1 Abtastzeit

Bei größerem zeitlichen Abstand der Einzelbilder einer Bildfolge steigt der räumliche Unsicherheitsbereich für die Position eines Objekts im Bild. Gleichzeitig steigt aber auch die zur Verfügung stehende Zeit zur Objektsuche. Als notwendige Bedingung für die kontinuierliche Objektverfolgung muß dieser Bereich in der vorhandenen Zeit analysiert werden, es gilt bei vollständiger Ausnutzung der Abtastzeit:

$$T_V = T_A$$

mit der Verarbeitungszeit T_V und der Abtastzeit T_A . Es gilt weiter

$$T_V = c \cdot b^2,$$

wobei b die Seitenlänge des als quadratisch angenommen Analysebereichs ist und c eine der Rechenleistung des verwendeten Bildverarbeitungssystems proportionale Konstante ist. Die Unsicherheit u in der Objektposition berechnet sich aus der Beschleunigung a des Objekts in Bildkoordinaten zu

$$u = \frac{1}{2} \cdot a \cdot T_A^2,$$

wobei offensichtlich $u \leq b$ gelten muß, das Objekt muß also im analysierten Bereich sein. Durch Umformen erhalten wir die Bedingung für das maximal zulässige a :

$$a \leq \frac{2}{\sqrt{c}} \cdot \frac{1}{T_A^{1,5}} \tag{5.1}$$

Im Ergebnis muß also die Abtastzeit so klein wie möglich gewählt werden, um das Objekt trotz einer großen Beschleunigung nicht aus dem Analysefenster zu verlieren. Dies ist anschaulich erklärbar, da die Abtastzeit linear zum Unsicherheitsbereich, die Fläche und damit der Rechenaufwand aber quadratisch dazu ansteigen. Steigt nun die Beschleunigung a über den maximal zulässigen Wert, ist ein Verfolgen des Objekts überhaupt nicht mehr möglich, auch nicht mit vergrößerter Abtastzeit T_A oder einem anderen Suchbereich.

In der Konsequenz ist die Objektverfolgung mit der minimal möglichen Abtastzeit T_A am robustesten gegen die Unsicherheit einer unbekannten Beschleunigung a , verwendet wird also die maximale Bildrate von Kamera und Framegrabber. Paradoxer Weise muß also bei geringer Rechenleistung eine schnelle Kamera verwendet werden oder entsprechend benötigt man bei einer schnellen Kamera weniger Rechenleistung. Die Wahl der minimal möglichen Abtastzeit führt im übrigen nicht nur zu einem robust dimensionierten Analysebereich, sondern auch zu einer robusten Lösung des Korrespondenzproblems entsprechend Abschnitt 5.3. Die beiden Kriterien führen also zum gleichen Ergebnis[1].

Die theoretische Grenze für die Verkleinerung des Analysebereichs ergibt sich aus der Quantisierung des Bildes. Zur Objektverfolgung müssen mindestens die 8 der prädizierten Objektposition benachbarten Analysefenster bearbeitet werden, ein kleinerer Analysebereich als 3x3 Pixel ist also nicht sinnvoll.

5.4.2 Optimaler Suchbereich

Wie in Abschnitt 5.4.1 erläutert, erfolgt die Objektverfolgung bei der minimalen, vom Kamerasystem vorgegebenen Abtastzeit. Das Suchfenster soll unter dieser Randbedingung möglichst groß gewählt werden. Da die Zeitdauer für die Objekterkennung vom Bildinhalt abhängt, kann keine Relation zwischen Zeitbedarf und der Größe des Such-

1 Dieser Gegebenheit kommt sicherlich die Entwicklung neuer Kameras mit CMOS-Technik entgegen, die das Einlesen kleiner Bildausschnitte mit einer hohen Bildrate, bis zu mehreren kHz, ermöglichen.

fensters angegeben werden. Dies wäre aber bei einem quadratischen Suchbereich notwendig, wenn die Analyse dieses Bereichs in einer Ecke desselben begonnen wird und die Größe im voraus festgelegt wird.

Abhilfe schafft statt zeilenweisem Abtasten ein spiralförmiges, beginnend an der prädizierten Objektposition. Das spiralförmige Austasten führt zu einem maximalen Suchbereich innerhalb der vorgegebenen Taktzeit der Kamera; es wird nach dieser Zeit abgebrochen. Es entsteht somit der maximal mögliche Suchbereich, unabhängig von der unbekannten Relation zwischen Analysezeit und Größe des Suchbereichs. Das Verfahren paßt die Größe des Suchbereichs automatisch der zur Verfügung stehenden Rechenzeit an. Bei Änderung der Rechenleistung oder bei Verwendung einer Kamera mit anderer Taktrate ergibt sich automatisch wieder die maximale und damit *optimale Größe*, ohne daß die Parameter des Algorithmus geändert werden müssen. Das gleiche gilt, wenn sich durch die Änderung des Bildinhaltes die Zeitdauer für eine einzelne Klassifikation ändert. Der Algorithmus arbeitet folglich immer im Optimum, unabhängig von der Relation zwischen der Größe des Suchbereichs und der dafür benötigten Zeit. Es kann stets die aus Rechenleistung und Videotaktrate berechenbare, maximale Beschleunigung des Objekts auch tatsächlich kompensiert werden.

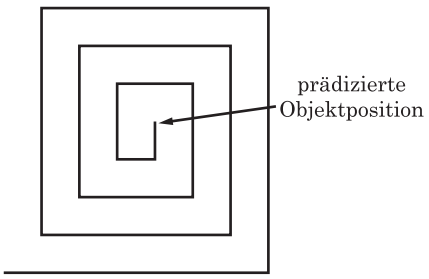

ABBILDUNG 5.4: Prinzipielle Suchstrategie: Spiralförmige Objektsuche, beginnend an der prädizierten Objektposition.

Das Verfahren bedarf einer genaueren Diskussion, da parallel dazu die Unterabtastung nach Abschnitt 4.2.1 zur Reduktion der Rechenzeit verwendet wird, die ebenfalls zu einer speziellen Auswahl von Analysefenstern beiträgt. Innerhalb der gröbsten, also in der ersten Schicht des Baums verwendeten Unterabtastung wird um die prädizierte Objektposition in der entsprechenden Auflösung die Spirale gelegt. Im Beispiel der Abb. 5.5 wird das Bild bei einer maximalen Unterabtastung um den Faktor 9 in entsprechende Blöcke zerlegt, wobei der erste Block an der prädizierten Position des Objekts zu liegen kommt.

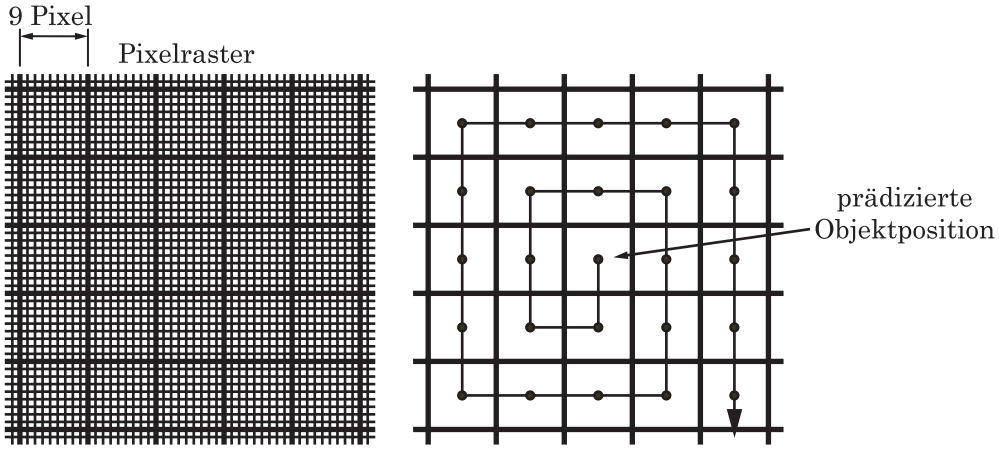

ABBILDUNG 5.5: Kombination von Unterabtastung und spiralförmiger Suche. Innerhalb jedes 9x9-Blockes wird nach Abschnitt 4.2.1 die Suchfensterposition festgelegt.

Innerhalb dieser Blöcke wird entsprechend Abschnitt 4.2.1 die Unterabtastung schrittweise verfeinert, d.h. abhängig von der Baumtiefe werden an den übersprungenen Positionen zusätzliche Analysefenster eingefügt und damit der ganze Bereich abgedeckt. Die durch die Unterabtastung entstehende zusätzliche Rasterung widerspricht geringfügig der obigen Zielvorstellung, möglichst in Form einer Spirale um den prädizierten Punkt zu wandern. Andererseits ist die zusätzliche Rasterung verhältnismäßig gering und der Geschwindigkeitsvorteil groß.

5.4.3 Initialisierung

In Abschnitt 5.4.1 und Abschnitt 5.4.2 wurde vorausgesetzt, daß bereits eine Prädiktion für die Objektposition vorliegt. Davon ausgehend wurde analysiert, unter welchen Voraussetzungen ein stabiles Verfolgen des Objekts möglich ist. Vernachlässigt wurden die Initialisierung der Verfolgung, bei der nur ein sehr grobes Vorwissen über die Objektposition vorliegt, und der in der Praxis sehr wichtige Fall, daß die Verfolgung durch Verletzung von Bedingung (5.1) oder eine Verdeckung des Objekts unterbrochen wird. Eine unterbrochene Verfolgung führt zu einer Verzögerung der nächsten Erkennung, damit zu großen Unsicherheiten im Ort und in der Konsequenz zu einem Verlust des Objekts aus dem Suchbereich. Dieser Zustand ist der Initialisierung ähnlich und kann gleichermaßen behandelt werden. In beiden Fällen kann die Eingrenzung des Suchbereichs nicht so schnell vorgenommen werden, daß die Bedingung für stabiles Verfolgen wieder erfüllt wäre.

Das Problem kann lediglich durch eine Objektsuche im gesamten Bild[1] gelöst werden. Da diese Analyse verhältnismäßig lange dauert, kann sie nicht mehrmals im ganzen Bild durchgeführt werden, bis wieder eine robuste Hypothese entsprechend Abschnitt 5.3.2 gefunden ist. Um die Bedingung (5.1) einzuhalten, muß das Objekt - sobald gefunden - sofort verfolgt werden. Stellt sich die verfolgte Hypothese als nicht robust heraus, muß wieder mit der Suche im ganzen Bild begonnen werden. Es ist also zwischen zwei Betriebsarten zu unterscheiden, der Objektsuche im Vollbild und der Objektverfolgung in einem kleinen Suchbereich, der dem Objekt nachgeführt wird. Zwischen beiden Modi ist je nach Erfolg oder Mißerfolg der Erkennung zu schalten, Abb. 5.6 verdeutlicht das Schaltgesetz.

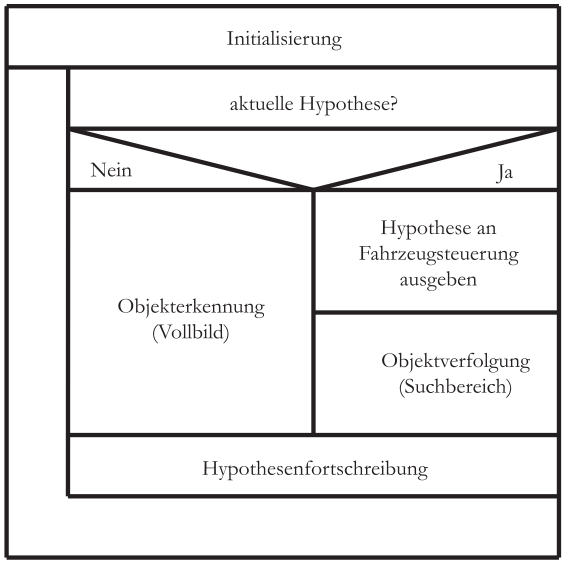

ABBILDUNG 5.6: Umschalten zwischen Objekterkennung im Vollbild und Objektverfolgen in einem kleinem Suchbereich.

1 Häufig existieren zusätzliche, anwendungsspezifische Einschränkungen.

Die Initialisierung des Verfahrens erfolgt immer mit der Objekterkennung im Vollbild. Die Umschaltung von der Objektverfolgung zur Objekterkennung im Vollbild kann in Anlehnung an den spiralförmigen Suchverlauf weich übergehen, indem die Spirale bei Mißerfolg der Objektverfolgung auf das gesamte Bild erweitert wird. An den Bildrändern muß die Spirale geeignet beschnitten werden, entsprechend Abb. 5.7.

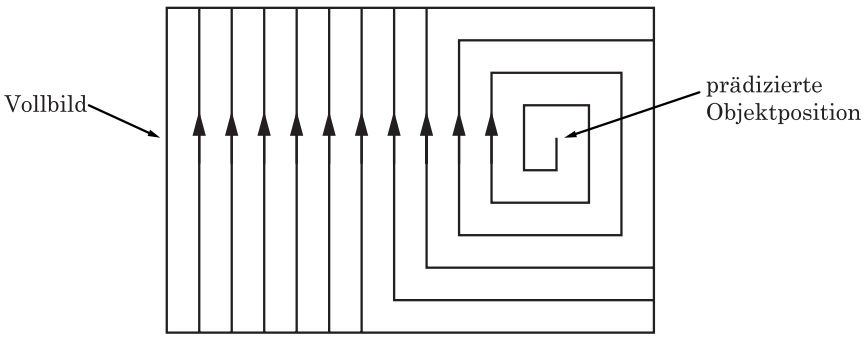

ABBILDUNG 5.7: Übergang der Suche an der prädizierten Objektposition auf die Suche im Vollbild. Ausgehend von der prädizierten Objektposition wird zunächst eine spiralförmige Suche durchgeführt und bei Mißerfolg auf das ganze Bild ausgedehnt.

6 Anwendungen

6.1 Videogestütztes Greifen für Serviceroboter

6.1.1 Problemstellung

Ein Beispiel für die Objekterkennung ist die bereits in der Einleitung genannte Erkennung von Objekten auf einem Tisch. Diese Aufgabe stellt sich mobilen Servicerobotern im Heimbereich und Krankenwesen. Sie sollen typische Objekte aus dem täglichen Leben wie Gläser und Flaschen greifen und abstellen, im wesentlichen für Hol-und-Bring-Aufgaben. Abb. 6.1 zeigt einen solchen Roboter, ROMAN. Das omnidirektionale Fahrzeug ist mit 3 unabhängig lenkbaren Rädern ausgestattet, von denen eines angetrieben wird. Objekte können mit einem Greifarm manipuliert werden. Für die *Selbstlokalisierung* besitzt das Fahrzeug eine laserbasierte Winkelmeßeinrichtung, die den Winkel zwischen Fahrzeughauptachse und fest an der Wand montierten und in ihrer Position bekannten, retroreflektierenden Landmarken bestimmt. Daraus können Position und Winkel des Fahrzeugs relativ zu einem Weltkoordinatensystem bestimmt werden. Zur *Detektion von Hindernissen* werden des weiteren 24 Ultraschallsensoren verwendet, die ringförmig am Fahrzeug angebracht sind. Ein Überblick über die gesamte Soft- und Hardwarearchitektur des Fahrzeugs findet sich in [7].

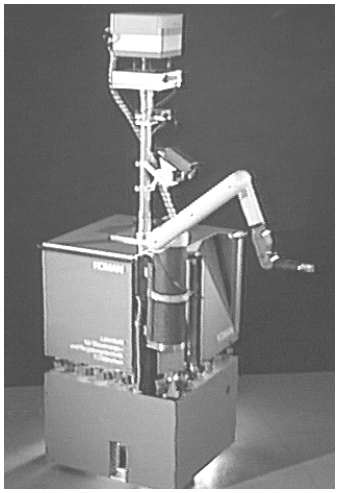

ABBILDUNG 6.1: Der semiautonome, mobile Serviceroboter ROMAN.

In einer typischen Aufgabe nach [17] soll ein Objekt von einem Tisch gegriffen werden, um es beispielsweise einer Person zu bringen. Vor Beginn des eigentlichen Greifmanövers wird der Roboter von dem Modul ´weiträumige Lokomotion´ zu einem

Anrückpunkt des Tisches gebracht, der mit diesem fest verknüpft ist und so gewählt wurde, daß der Tisch im Blickfeld der Kamera zu sehen ist. Der Bildverarbeitung werden bereits vorher die gesuchten Objekte mitgeteilt, so daß bereits typischerweise 3 m vor dem Tisch mit der Suche begonnen werden kann. Dadurch ist sichergestellt, daß die Objekte zum frühest möglichen Zeitpunkt erkannt und während der Annäherung kontinuierlich verfolgt werden.

Die wesentlichen Anforderungen an die videogestützte Objektlokalisierung können direkt aus diesem Szenario abgeleitet werden. Die Objektposition muß während der Annäherung an das Objekt in Echtzeit erkannt werden, ohne daß das Fahrzeug für die Erkennung des Objekts seine Geschwindigkeit verringert oder stehenbleibt. Kann das Objekt bei einer maximalen Entfernung von 2,0 m erkannt werden und bewegt sich das Fahrzeug mit einer Geschwindigkeit von ca. 200 mm/s, stehen zur Erkennung maximal 5 s zur Verfügung. Das Objekt muß spätestens in einem Abstand von ca. 1,0 m erkannt werden. Diese Erkennung wird bis zum eigentlichen Greifvorgang präzisiert und zur Korrektur der Fahrzeuglage verwendet.

6.1.2 Videotechnik und Beleuchtung

6.1.2.1 Kamera

Für die Sichtbarkeit des Objekts muß die Kamera geeignet auf dem Fahrzeug positioniert werden. Sie sollte weit vorne am Fahrzeug und deutlich oberhalb der typischen Arbeitsfläche, dem Tisch, angeordnet sein. Dies ermöglicht einen Überblick über die Objekte auf dem Tisch und reduziert die gegenseitige Verdeckung der Objekte. Darüber hinaus ermöglicht die Betrachtung der Tischfläche von schräg oben eine verhältnismäßig hohe Genauigkeit in der Objektlokalisierung. Bei dieser Anordnung wird jedoch der Arbeitsraum des Arms zu stark eingeschränkt, weshalb die Kamera zurückgesetzt werden muß. Des weiteren darf der Fahrzeugkörper nicht den Blick auf den Tisch verdecken, er muß gegebenenfalls vorne abgeschrägt werden. Bei ROMAN wurde in einem Kompromiß die Kamera ca. 15 cm vor der Fahrzeugmitte in einer Höhe von ca. 150 cm montiert, der Arbeitsbereich des Arms wird dadurch nur unwesentlich eingeschränkt. Mittels einem Neigekopf kann das Blickfeld für unterschiedliche Aufgaben während des Betriebs verändert werden.

Neben der Position der Kamera ist auch ihre optische Parametrierung der Problemstellung anzupassen. Einerseits muß die Kamera für das Suchen des Objekts auf der Tischfläche ein genügend großes Blickfeld besitzen, andererseits müssen die Objekte noch so groß abgebildet werden, daß sie in entsprechendem Abstand erkannt werden können. Dies hängt auch von der Kameraauflösung ab, die aber wiederum technisch begrenzt ist. Das Blickfeld muß so groß sein, daß es den Suchbereich auf dem Tisch

überdeckt. Die verwendete Kamera[1] besitzt ein Blickfeld von 44,3°, was bei einer Entfernung von ca. 1 m einem Suchbereich von ca. 80 cm Breite auf dem Tisch entspricht.

Als preiswerte Standardkomponente wird bei ROMAN eine Kamera nach PAL-Norm verwendet. Zur Begrenzung der Datenmenge auf 2,8 Mbyte/s wird die Kamera mit halber Auflösung betrieben, dies sind Bilder mit 384x288 Pixel bei 25 Hz. Diese Daten stammen von lediglich einem PAL-Halbbild, was die typischen Störeffekte von aus zwei Halbbildern zusammengesetzten Vollbildern vermeidet.

6.1.2.2 Digitalisierung

Die Digitalisierung erfolgt im Videotakt von 25 Hz. Über den Systembus werden diese Bilder innerhalb von 13 ms von der Videokarte[2] in den Bildverarbeitungsrechner übertragen, ein VME-BUS Rechner mit SuperSparc Architektur[3]. Die Übertragung der Bilder erfolgt per DMA und belastet den Rechner nicht, allerdings stehen durch die Übertragungszeit nur noch 27 ms für die eigentliche Bildanalyse zur Verfügung. Kamera und Framegrabber sind auch für Farbbilder ausgelegt, allerdings dauert die Übertragung eines Farbbildes zwischen Framegrabber und Rechner ca. 40 ms.

Um nach Abschnitt 5.4 möglichst robust gegen Objektbeschleunigungen zu werden, muß eine möglichst kleine Verzögerungszeit zwischen Bildaufnahme durch die Kamera und der Bildverarbeitung erreicht werden. Unter Berücksichtigung der technischen Randbedingungen[4] ergibt sich das Schema nach Abb. 6.2.

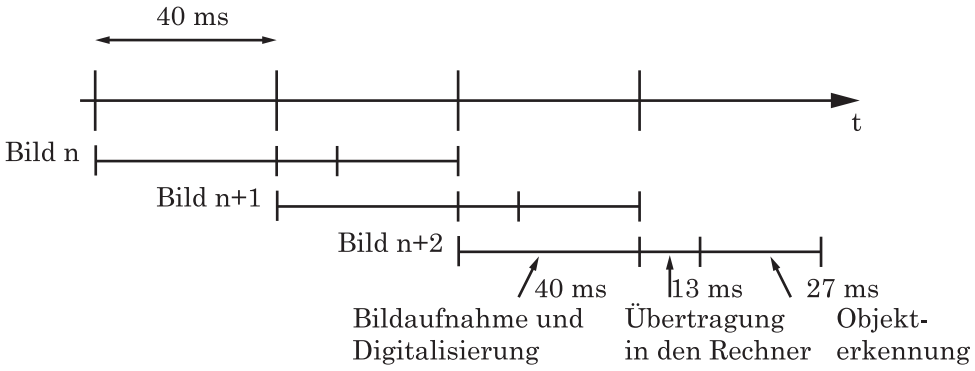

ABBILDUNG 6.2: Bildeinzug und synchrone Verarbeitung im Kameratakt.

1 Die PAL-Kamera EVI-311 der Fa. Sony mit einer Brennweite von 5,9 mm.

2 ´SunVideo X1086A´ der Firma Sun Microsystems.

3 Rechenleistung: ca. 143 SpecInt92 bei 170 MHz Taktfrequenz.

4 Durch genauere Steuermöglichkeiten des Framegrabbers könnte zusätzlich Zeit gewonnen werden. Beispielsweise könnte nach Einzug des ersten Halbbildes, also nach 20 ms, dieses bereits zum Rechner übertragen werden. Eine weitere Verbesserungsmöglichkeit wäre ein Framegrabber, der jeweils nur eine einzelne Zeile zwischenspeichert und alle 64 us eine Zeile zum Rechner überträgt. Derartige Framegrabber gibt es derzeit allerdings nur in Verbindung mit dem PCI-Bus.

Als maximale Verzögerungszeit ergeben sich für ein Objekt am oberen Bildrand 53 ms, für ein Objekt am unteren Bildrand 40 ms. Der Computer wird nur zu 67,5% ausgelastet.

Für die initiale Objektsuche im gesamten Bild ist die Verzögerungszeit von untergeordneter Bedeutung. Die Suche im Vollbild benötigt mehr als 40 ms und ist nicht konstant. Trotzdem werden die Bilder im 40 ms-Takt eingelesen. Ist die Erkennung eines Bildes beendet, steht bereits ein aktuelles Bild zur Verfügung, und es kann sofort mit der Objekterkennung in diesem Bild begonnen werden. Abb. 6.3 verdeutlicht das Vorgehen. Sofort nach der Erkennung in Bild n-2 wird das Objekt in Bild n gesucht, dem neuesten, im Computerspeicher zur Verfügung stehende Bild. Das Bild n-1 wird zwar eingelesen und an den Rechner übertragen, die Übertragung wird aber nicht rechtzeitig beendet. Der Computer wird bei diesem Verfahren zu 100% ausgelastet.

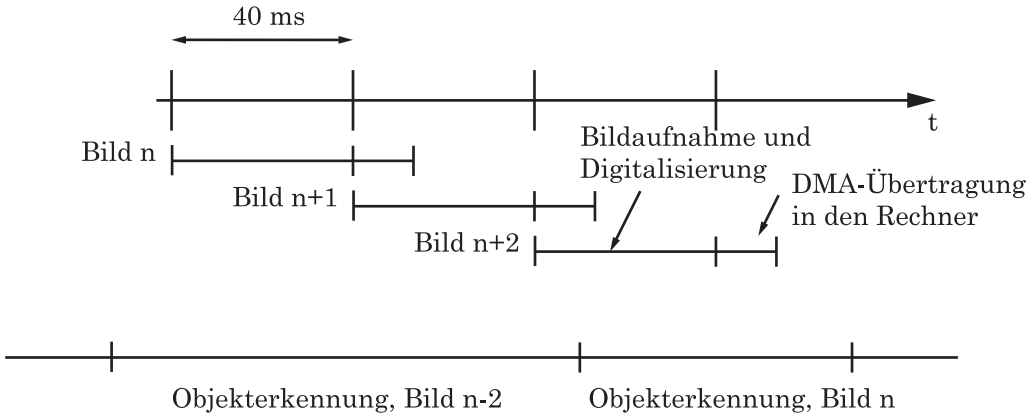

ABBILDUNG 6.3: Bildeinzug und asynchrone Verarbeitung für maximale Auslastung des Rechners.

6.1.2.3 Beleuchtung

Zur Kompensation von Schwankungen in der Beleuchtung wird eine in die Kamera integrierte, automatische Regelung der Blendeneinstellung und zusätzlich eine automatische Verstärkungsregelung (automatic gain control, AGC) verwendet. Beide Verfahren gleichen Schwankungen in der Beleuchtung weitgehend aus, solange diese über das Bildfeld gleichmäßig verteilt sind. Starke Kontraste im Bild, besonders partielles Sonnenlicht in ansonsten mäßig beleuchteten Räumen führen trotzdem zu lokaler Über- und Unterbelichtung.

Ein wenigstens teilweises Nivellieren solcher Unterschiede ermöglicht eine künstliche Beleuchtung des Blickfeldes durch eine Fahrzeugbeleuchtung. Diese Beleuchtung ist konstant und mischt sich mit dem veränderlichen Umgebungslicht, die Summe der Lichtquellen ist dadurch geringeren Schwankungen unterworfen als das Umgebungs-

licht alleine. Direkte Sonneneinstrahlung ist im Verhältnis so groß, daß sie durch diese Maßnahme nicht kompensiert werden kann. Außerdem stellt die künstliche Beleuchtung eine Mindestbeleuchtung dar. Für das Fahrzeug ROMAN wird eine 50 W Halogenlampe verwendet, die bei Objekten bis 2 m Abstand ausreicht. Damit das Blickfeld der Kamera gleichmäßig ausgeleuchtet wird, muß der Öffnungswinkel des Lichtkegels dem Blickwinkel der Kamera entsprechen und die Beleuchtung koaxial an der Kamera montiert sein. Außerdem ist auf einen geringen Abstand der Lichtquelle zur Kamera zu achten, damit der Schattenwurf von Objekten im Kamerabild nicht sichtbar wird.

6.1.3 Erzeugung von Trainingsbildern

Aus der oben beschriebenen Problemstellung lassen sich nun die geometrischen Freiheitsgrade des Objekts relativ zur Kamera bestimmen. Da die betrachteten Objekte nach Abb. 1.2 aufrecht auf dem Tisch bekannter Höhe stehen, können sie sich relativ zum Fahrzeug nur in 3 Freiheitsgraden bewegen. Dies sind die Drehung um die eigene Hochachse und die 2 translatorischen Freiheitsgrade auf der Tischfläche. Allerdings bleibt die Abbildung eines Objekts unverändert, wenn die Kamera um die Hochachse gedreht wird; in diesem Fall ändert sich nur die Position des Objekts im Bild. Insgesamt bleiben also zwei geometrische Freiheitsgrade übrig, im Fall eines um die Hochachse rotationssymmetrischen Objekts sogar nur ein Freiheitsgrad.

Das Werteintervall in jedem Freiheitsgrad wird durch die Problemstellung festgelegt. Beim rotatorischen Freiheitsgrad ist dies natürlich der Bereich von $0°$ bis $360°$. Der Entfernungsbereich wird durch die minimale und die maximale Entfernung festgelegt, in denen die Erkennung des Objekts erfolgen soll. Beide Freiheitsgrade werden gerastert, wobei beispielsweise für das Objekt ´Schachtel´ experimentell in jeder Dimension ca. 10 Rasterpunkte als ausreichend ermittelt wurden. Zusätzlich wurde das Objekt in 2 unterschiedlichen Beleuchtungssituationen aufgenommen. Dieses Objekt besitzt eine matt reflektierende Oberfläche, seine Abbildung wird deshalb durch die Beleuchtung nur wenig beeinflußt. Es ergeben sich also ca. 200 Trainingsbilder.

Für die Erkennung des Objekts ´Glas´ ergibt sich eine andere Situation. Dieses Objekt ist von der Beleuchtung und dem Hintergrund sehr stark abhängig, jedoch rotationssymmetrisch und besitzt deshalb nur einen geometrischen Freiheitsgrad. Zum Erfassen der unterschiedlichen Beleuchtungen wird es in 4 verschiedenen Situationen aufgenommen:

- nur künstliche Deckenbeleuchtung
- natürliche Beleuchtung von links
- natürliche Beleuchtung von rechts
- natürliche Beleuchtung von vorne

Der Entfernungsbereich von 0,8 m bis 2 m wird in 30 Schritten gerastert, so daß die Trainingsklasse des Objekts aus insgesamt 4x30 = 120 Ansichten besteht. Einfache rotationssymmetrische Objekte, wie das Objekt ´Flasche´, benötigen für eine robuste Erkennung noch weniger Ansichten, dort sind bereits ca. 50 Objektansichten ausreichend. Zweckmäßigerweise werden die Objektbilder während typischer Greifmanöver aufgenommen, damit die Ansichten des Objekts während des Trainings den Ansichten des Objekts für die reale Erkennung möglichst entsprechen. Eine Alternative dazu besteht in einem gezielten Anfahren aller festgelegten Rasterpunkte mit einem entsprechenden Programm. Unabhängig dazu ist für die in Abschnitt 4.3 beschriebene Entfernungsbestimmung des Objekts die Position des Fahrzeugs für jedes Trainingsbild zu speichern. Zusammen mit der Position des Objekts kann daraus für jedes Trainingsbild die dazugehörige Entfernung Kamera-Objekt berechnet werden und nach Generierung des Musterbaums in den jeweiligen Endknoten abgelegt werden.

Selbstverständlich hängt die Anzahl der notwendigen Objektansichten auch von der Zurückweisungsklasse ab, die in den durchgeführten Experimenten aus jeweils 100 bis 200 Bildern besteht und eine außerordentlich große Vielfalt an Objekten beinhaltet. Abb. 6.4 zeigt zwei Beispiele für typische Bilder der Zurückweisungsklasse. Um eine möglichst große Variabilität in der Zurückweisungsklasse zu erhalten, werden möglichst unterschiedliche Ansichten innerhalb der Laborumgebung mit vielen verschiedenen Objekten aufgenommen. Außerdem werden diverse Beleuchtungsvarianten verwendet, insbesondere Tageslicht zu verschiedenen Tageszeiten, künstliche Beleuchtung und Mischbeleuchtung. Die Bilder der Zurückweisungsklasse sind für alle Objekte dieselben und müssen jeweils lediglich um wenige Bilder ergänzt werden, die besonders verwechslungsrelevante Objekte beinhalten.

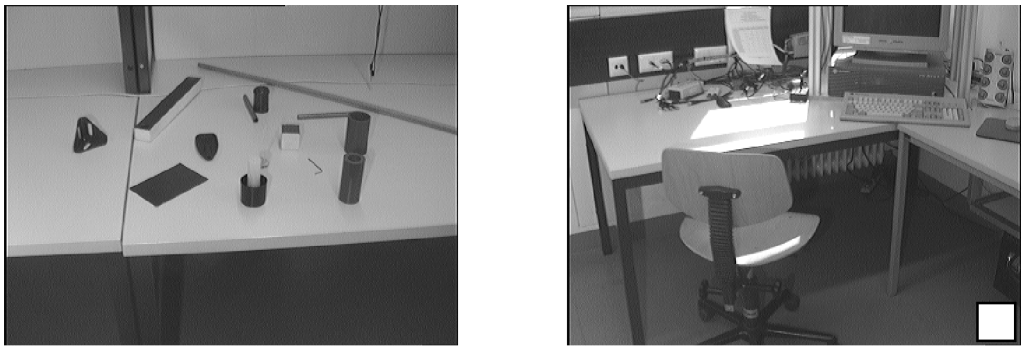

ABBILDUNG 6.4: Bilder der Zurückweisungsklasse. Jeder mögliche Bildausschnitt der Größe $31x31$ ist eines der Zurückweisungsmuster. Im rechten Bild ist zur Veranschaulichung ein solcher Ausschnitt eingezeichnet.

6.1.4 Markierung der Objekte

Um innerhalb der Objektbilder das Objekt zu markieren, werden dem Anwender programmunterstützt alle Objektbilder gezeigt. Er muß nur mittels einem Fadenkreuz denjenigen Punkt auf dem Objekt markieren, der später als Objekt erkannt wird. Wie in Abb. 6.5 gezeigt, wird ebenfalls die Größe des Vergleichsmusters angezeigt. Bei den Objekten nach Abb. 1.2 und den oben beschriebenen Kameraparametern ergibt sich eine Mustergröße von typisch 31x31 Pixel. Beim Markieren des Objekts muß darauf geachtet werden, daß ein *repräsentativer Ausschnitt* des Objekts in diesem Bereich zu sehen ist; bei kleinen Objekten umfaßt dieser Bereich das gesamte Objekt.

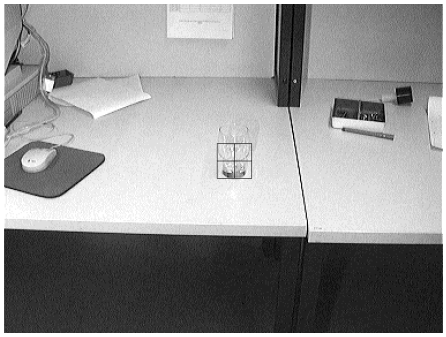

ABBILDUNG 6.5: Bildschirmkopie des Markierungsprogramms. Sichtbar sind das Gesamtbild, das markierte Objekt und das Fadenkreuz, das der Anwender mittels Computermaus auf dem Objekt positioniert.

Um das Markieren des Objekts zu vereinfachen, wird dem Anwender jedes Bild in doppelter Größe angezeigt[1]. Damit ist es leicht das Fadenkreuz mit einer Unsicherheit von maximal ±1 Pixel zu positionieren. Dies erlaubt, in jedem Bild denselben Punkt des Objekts zu markieren. Der Arbeitsschritt dauert nur einige Sekunden pro Bild, das Markieren von 100 Bildern nur wenige Minuten. Die Objektpositionen innerhalb der Bilder werden in einer Datei abgespeichert, die zusammen mit den Bildern der Objekt- und Zurückweisungsklasse zur Generierung des Baums als Eingangsdaten vom Trainingsprogramm eingelesen wird.

6.1.5 Baumgenerierung

Je nach Komplexität der Muster ergeben sich unterschiedlich große Musterbäume, die jeweils durch folgende Kenngrößen charakterisiert werden können:

- Anzahl der Knoten.

1 Jedes Pixel des Bildes wird durch 4 Bildschirmpixel dargestellt, dadurch kann der Anwender einzelne Pixel des Bildes erkennen.

- Anzahl aller Parameter (Einträge in den Gewichtungsvektoren und Offset).
- Durchschnittliche/Maximale Baumtiefe.

Selbstverständlich bedeuten jeweils kleine Kenngrößen bessere Bäume als große. An diesen Kenngrößen läßt sich wenigstens relativ die Güte eines Baums für ein Objekt messen, wobei komplexere Objekte natürlich auch zu größeren Bäumen führen und deshalb Bäume verschiedener Objekte nicht verglichen werden können. Die nachfolgende Tabelle zeigt für verschiedene Objekte im Servicebereich die Kenngrößen für die Bäume:

Objekt	Flasche	Dose	Spitzer	Glas
Anzahl der Knoten	897	1523	2035	3523
Anzahl der Parameter (in 1000)	34	79	70	215
Durchschnittliche/Max. Tiefe	10/15	9/15	12/18	12/20
Typische Zeit für die Objektsuche pro Bild	~80ms	~160ms	~160ms	~200ms

TABELLE 6.1: Größe des Musterbaumes für die Objekte im Serviceszenario, die Zeitangaben für die Objektsuche schwanken je nach Hintergrund um einen Faktor 0,5 bis 2.

Erwartungsgemäß führen einfach strukturierte Objekte mit sehr starken Kontrasten zu kleineren Bäumen. Beispielsweise läßt sich die ´Flasche´ mit nur wenigen Knoten erkennen, im Gegensatz zum ´Glas´, das sehr vielen, zufälligen Mustern ähnlich ist. Die Anzahl der Knoten in einem Baum ist dadurch auch ein Maß, wie ähnlich oder unterschiedlich ein Objekt zu anderen Mustern ist. Allerdings ist eine individuelle Interpretation der Baumstrukturen einzelner Objekte nicht möglich, da die Datenmenge nicht mehr überschaubar ist.

6.1.6 Typische Störungen und Erkennungswahrscheinlichkeit

6.1.6.1 Fehl-Erkennung

Bei der Erkennung eines Objekts können prinzipiell zwei unterschiedliche Fehler auftreten. Einerseits kann ein anderes Objekt als das gesuchte erkannt werden; wir bezeichnen diesen Fall als Fehl-Erkennung. Wird das gesuchte Objekt in bestimmten Fällen nicht erkannt, handelt es sich um eine Nicht-Erkennung.

Fehl-Erkennungen treten bei ähnlichen Objekten auf, wobei bei ansichtsbasierten Erkennern diese Fehler meistens plausibel sind. Die verwechselten Objekte sind auch für den Menschen dem gesuchten Objekt ähnlich. Verwechselte Objekte, die entsprechend der menschlichen Perzeption keinerlei Ähnlichkeit mit dem gesuchten Objekt haben, treten bei ansichtsbasierten Erkennern sehr selten auf. Abb. 6.6 zeigt eine verwechslungsrelevante, schwarze Strebe eines Labortisches neben dem gesuchten Objekt Flasche, wobei im Suchfenster nur der untere Teil der Flasche zu sehen ist.

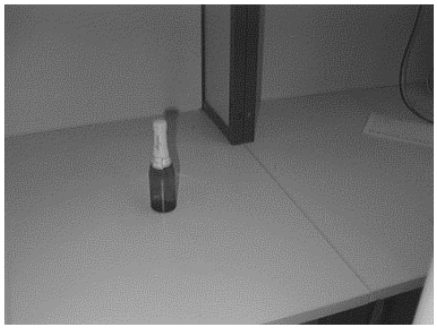

ABBILDUNG 6.6: Flasche und verwechslungsrelevante, schwarze Strebe eines Tisches.

Eine szenenunabhängige Wahrscheinlichkeit[1] für einen solchen Fehler läßt sich grundsätzlich nicht angeben. Die Wahrscheinlichkeit hängt in jedem Fall maßgeblich von der Szene ab. Ist ein dem gesuchten Objekt sehr ähnliches Objekt häufig im Bildfeld, kann die Fehlerwahrscheinlichkeit für eine Fehl-Erkennung sehr groß werden wie etwa in Abb. 6.6. Bei 8 durchgeführten Greifexperimenten wurde die Strebe zweimal als Flasche erkannt, entsprechend einer Wahrscheinlichkeit von 0,25. Festgestellt wurde dabei nicht die Wahrscheinlichkeit pro Bild, sondern die Wahrscheinlichkeit, daß eine robuste Hypothese aus einer entsprechenden Anzahl von Bildern generiert wurde. Allerdings war die Strebe nicht in der Trainingsmenge der Zurückweisungsklasse enthalten. Bei Erkennung an einem anderen Tisch ohne Strebe nach Abb. 1.1 wurde in 19 Greifexperimenten keine falsche Hypothese[2] generiert, entsprechend einer Wahrscheinlichkeit $<5{,}3\%$. In diesen Greifexperimenten wurden insgesamt

1 Die Wahrscheinlichkeit für eine Fehl-Erkennung oder Nicht-Erkennung ist immer von der Szene abhängig. Ist die Szene bei der Erkennung und beim Training exakt gleich, findet eine Erkennung des Objekts statt. Sind die Szenen bei Erkennung und Training nicht gleich, sinkt die Wahrscheinlichkeit für eine korrekte Erkennung und die Wahrscheinlichkeit für eine Erkennung eines anderen Objekts steigt. Die Wahrscheinlichkeit für eine korrekte Erkennung entspricht also einer ´Ähnlichkeit´ zwischen diesen Szenen. Diese ´Ähnlichkeit´ kann für reale Szenen nicht formalisiert behandelt werden, da zu viele Einflüsse eine Rolle spielen, die Form und Oberfläche anderer Objekte, die Beleuchtung und natürlich die Anordnung der Objekte in der Szene. Aus diesem Grund können die Wahrscheinlichkeiten nur exemplarisch für bestimmte Szenen angegeben werden und nicht auf andere Objekte bzw. Szenarien übertragen werden. In diesem Sinn sind die Wahrscheinlichkeitsangaben in diesem Kapitel nur mit Vorbehalt auf andere Szenarien übertragbar. Im Einzelfall kann nur eine experimentelle Validierung über diese Frage Auskunft geben.

2 Das Objekt wird nur dann im gesamten Bild gesucht, solange keine robuste Hypothese vorhanden ist.

253 Einzelbilder ausgewertet und dabei 8 mal ein anderes Muster als das Objekt erkannt. Dies entspricht einer Wahrscheinlichkeit von *3,2%* . Eine falsche Hypothese wurde deshalb nicht generiert, weil die 8 Fehl-Erkennungen zeitlich und/oder örtlich nicht zusammenfielen. Aus diesem Wert kann die Wahrscheinlichkeit für die Bildung einer falschen Hypothese berechnet werden. Werden bei einer Objekterkennung 20 Bilder der Szene analysiert und sind 10 Erkennungen des falschen Objekts für die Hypothese notwendig, wird diese mit einer rechnerischen Wahrscheinlichkeit von nur $2 \cdot 10^{-9}$ gebildet. Die Beispiele zeigen, wie stark die Wahrscheinlichkeit für die Erkennung von Szene zu Szene variieren kann.

6.1.6.2 Nicht-Erkennung

Der zweite mögliche Fehler besteht darin, daß das Objekt zwar im Bildfeld zu sehen ist, aber nicht erkannt wird. Hier kann ebenfalls kaum eine allgemeine Aussage über die Wahrscheinlichkeit gemacht werden. Der Grund ist ein ähnlicher; durch eine nicht im Training vorkommende Beleuchtung oder einen bestimmten Hintergrund kann die Erkennungswahrscheinlichkeit in einer bestimmten Szene sehr stark absinken. Andererseits ist die Erkennungswahrscheinlichkeit in einer Szene sehr groß, wenn ungefähr die selben Umstände wie beim Training vorhanden sind.

Allerdings ist die Wahrscheinlichkeit nur noch von der Beleuchtung des Objekts und dem unmittelbar angrenzenden Hintergrund abhängig, aber nicht mehr von den anderen Objekten in der Umgebung. Dementsprechend schwankt die Wahrscheinlichkeit sehr viel weniger. Tabelle 6.2 gibt für 3 Objekte im Serviceszenario Wahrscheinlichkeiten für die Erkennung in einem Einzelbild an. Die Stichprobe besteht aus jeweils 200 Einzelaufnahmen, die bei typischen Greifversuchen aufgenommen wurden.

Objekt	Glas	Flasche	Dose
Wahrscheinlichkeit	95%	97%	83%

TABELLE 6.2: Erkennungs-Wahrscheinlichkeit für 3 Objekte im Serviceszenario.

Die Wahrscheinlichkeiten wurden unter ähnlichen Bedingungen gemessen, unter denen auch das Trainingsmaterial aufgenommen wurde. Identisch waren der Abstandsbereich zwischen Kamera und Objekt, und das Objekt stand jeweils auf einem weißen Tisch. Die Beleuchtung schwankte im Rahmen der in Abschnitt 6.1.3 angegebenen Situationen.

Die geringe Wahrscheinlichkeit des Objekts ´Dose´ läßt sich damit erklären, daß dieses Objekt in der Trainingsmenge nur in bestimmten Orientierungen auftauchte. Da der Umfang des Objekts sehr unterschiedlich eingefärbt ist, wird es in einer bestimmten Orientierung fast nicht erkannt, in den anderen Orientierungen dagegen sehr gut.

Die Wahrscheinlichkeiten in der Tabelle genügen, um die Objekte reproduzierbar zu erkennen, um also bei Greifexperimenten eine robuste Hypothese zu generieren. Nur das Objekt ´Dose´ wurde in einer bestimmten Orientierung nicht erkannt.

6.1.7 Genauigkeit der Lokalisierung

Die Genauigkeit in Bildkoordinaten ist weitgehend unabhängig von der Entfernung der Kamera vom Objekt. Tabelle 6.3 gibt für 4 Objekte die Standardabweichung in Pixel an, über alle Objekte gemittelt beträgt sie 2,5 Pixel.

Objekt	Standardabweichung in Pixel
Dose	3
Flasche	2
Spitzer	2
Glas	3

TABELLE 6.3: Abweichung zwischen idealem und tatsächlichem Erkennungspunkt im Bild.

Zur Bestimmung der Genauigkeit wurden pro Objekt 10 Bilder in geringem Abstand (100 cm) und 10 Bilder in großem Abstand (250 cm) ausgewertet. Dafür wurde jeweils manuell der Idealpunkt für die Erkennung festgelegt und sein Abstand zum tatsächlichen Punkt der Erkennung bestimmt.

Diese in Bildkoordinaten konstante Genauigkeit führt zu einer entfernungsabhängigen Genauigkeit in kartesischen Fahrzeugkoordinaten. Entsprechend der Kameradaten in Abschnitt 6.1.2 entspricht der mittlere Fehler in Bildkoordinaten einem absoluten Fehler von 5 mm in 100 cm Abstand und 12,5 mm bei einem Abstand von 250 cm. Diese Werte gelten für den Fehler senkrecht zur Verbindungslinie Kamera-Objekt. Dieser Fehler geht in die Bestimmung der Entfernung nach dem Triangulationsverfahren aus Anhang D ein und führt zu einem Fehler von 7 mm für die Entfernung. Dieser Fehler wird durch die unsicheren Parameter vergrößert, die in die Entfernungsbestimmung eingehen. Der tatsächliche Entfernungsfehler ergibt sich je nach Objekt zu ca. 15 mm.

Die Genauigkeit reicht für das Greifen des Objekts aus, wenn bestimmte Bedingungen erfüllt sind. Zunächst wird dazu der Zweifingergreifer so positioniert, daß das Objekt zwischen den Fingern zu liegen kommt. Bei der Positionierung darf der Greifer das

Objekt nicht berühren. Berührt der Greifer das Objekt bei horizontaler Annäherung, schiebt er dieses vor sich her. Beim Schließen der Finger wird das Objekt nicht gegriffen. Berührt der Greifer das Objekt bei senkrechter Annäherung, drückt er dieses gegen die Tischplatte. Der Greifarm verformt sich oder das Objekt fällt um, in beiden Fällen kann es ebenfalls nicht gegriffen werden. Für ein großes Objekt muß dabei die Genauigkeit größer sein, da dort der Abstand zwischen Finger und Objekt kleiner ist. Nach Erreichen der Greifposition wird der Greifer geschlossen. In dieser Phase ist ein Fehler fast ausgeschlossen. Wurde der Greifer ungenau positioniert, wird das Objekt von einem Finger früher berührt. Dieser verschiebt das Objekt zwar ein wenig auf der Tischfläche, der Greifvorgang wird aber nicht gestört.

Die typische Greifposition ist aus Abb. 6.7 ersichtlich. Die Bewegung des Greifarms erfolgt im wesentlichen in Richtung des Sehstrahls. Die erforderliche Genauigkeit in x-Richtung entspricht der halben Länge eines Fingers, also 20 mm.

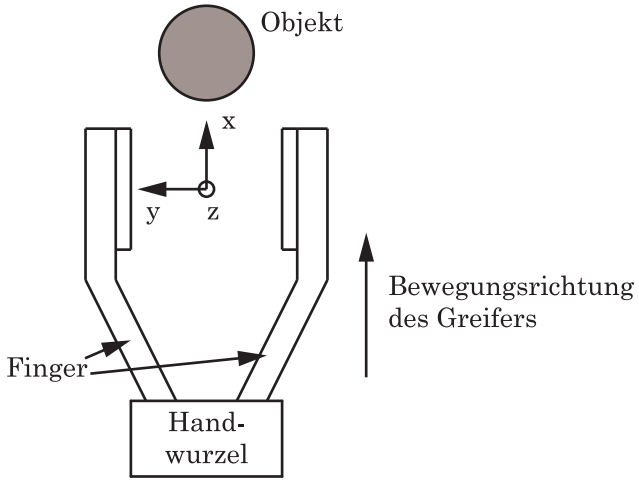

ABBILDUNG 6.7: Typische Greifposition. (Ansicht von oben.)

In die Genauigkeit des Greifvorgangs gehen neben dem Lokalisierungsfehler der Bildverarbeitung auch die Fehler des Greifarms und der Fahrzeuglokalisierung ein. Jeder dieser Fehler beträgt bei ROMAN ca. 5 mm, zusammen also 15 mm. Entsprechend kann bei einer Greiföffnung von 90 mm ein Objekt mit 60 mm Ausdehnung in y-Richtung gegriffen werden. Die erforderliche Genauigkeit in z-Richtung ist von der Höhe des Objekts abhängig. Da für die z-Koordinate das Vorwissen der Tischhöhe verwendet wird, und in dieser Koordinate auch nicht die Unsicherheit der Fahrzeuglokalisierung eingeht, ist die Positionierung auf ca. 5 mm genau. Dies ist für die hier diskutieren Objekte vollkommen ausreichend. Ebenfalls kein Problem stellt ein etwaiger Winkelfehler dar.

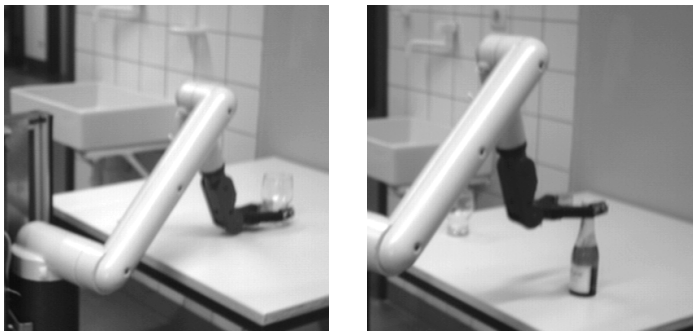

ABBILDUNG 6.8: Greifpositionen für die Objekte ´Glas´ und ´Flasche´.

6.1.8 Nicht trainierbare Objekte

Obwohl theoretisch durch eine entsprechend große Trainingsmenge jedes Objekt erkannt werden kann, führt eine zu große Variabilität in der Praxis zu Einschränkungen. Insbesondere sind Objekte problematisch, die ihr Aussehen in Abhängigkeit von einem bestimmten Freiheitsgrad sehr stark verändern. Ein Beispiel ist ein schmaler Stift, entsprechend Abb. 6.9, der bei einer nur geringfügigen Drehung um die senkrechte Achse sein Aussehen sehr stark ändert.

ABBILDUNG 6.9: Große Variabilität eines Stiftes bei Drehung: Schon durch eine geringfügige Drehung verändern sich sehr viele Pixel von schwarz zu weiß und umgekehrt.

Dieser Freiheitsgrad müßte durch sehr viele Aufnahmen abgedeckt werden, da sich bei einer Drehung um nur 5° fast alle Pixel des Stifts von weiß auf schwarz und umgekehrt ändern. Eine Rasterung dieses Freiheitsgrades bei einer Schrittweite von 2,5° erfordert bereits 144 Muster. Damit ist allerdings nur ein Freiheitsgrad abgedeckt, die Rasterung weiterer Freiheitsgrade führt zu entsprechend mehr Mustern. Hierbei müßte zusätzlich durch definierte Aufnahmebedingungen für das Trainingsset sichergestellt werden, daß

keine ´Löcher´ in den gerasterten Wertebereichen auftauchen; beispielsweise könnte der Stift während der Aufnahme mittels einem Drehtisch in genau festgelegten Intervallen gedreht werden.

Eine andere, nicht trainierbare Klasse von Objekten sind Objekte, die sich von der Zurückweisungsklasse nur durch eine Textur unterscheiden. Da Textur im allgemeinen relativ hochfrequente Anteile besitzt, ändert sie sich außerordentlich stark mit der Position des Objekts, also beispielsweise mit der Entfernung zur Kamera und der Drehung. Dementsprechend wird hier ein in der Praxis nicht mehr handhabbar großes Trainingsset benötigt. Ein Beispiel ist die in Abb. 6.10 gezeigte Tastatur, wobei das Tastenfeld als spezielle Textur erscheint. Eventuell kann ein solches Objekt anhand anderer Kennzeichen erkannt werden. Die Textur stört dann nicht, die mit Textur belegten Objektflächen werden von der Entropieoptimierung als irrelevant gewertet und weitgehend ignoriert.

ABBILDUNG 6.10: Objekt mit starker Textur: Tastatur.

6.1.9 Typisches Greifexperiment

In einem typischen Greifexperiment wird das Fahrzeug zunächst vom Anwender kommandiert, beispielsweise mit dem Befehl ´Hole die Flasche vom Eßtisch´ über die natürlich-sprachliche Eingabeschnittstelle des Roboters, beschrieben in [18]. Dieses natürlichsprachliche Kommando wird durch eine umfangreiche, sprachliche Verarbeitung in Teilbefehle für die einzelnen Teilmodule der Fahrzeugsteuerung umgesetzt. Die abstrakten Begriffe werden mittels einer Datenbank kartesischen Koordinaten zugeordnet.

Zunächst wird das Fahrzeug von dem Modul ´weiträumige Lokomotion´ an einen 1. Annäherungspunkt des betreffenden Tisches gebracht. Diese Teilaufgabe erfolgt mit eingefaltetem Arm und einer relativen großen Geschwindigkeit des Fahrzeugs, da gegebenenfalls größere Entfernungen zurückzulegen sind. Die Planung in dieser Phase ist verhältnismäßig einfach, da das Fahrzeug in nur 3 Freiheitsgraden bewegt wird. An dem 1. Annäherungspunkt, ca. 2200 mm vor dem Objekt, wird von einem übergeordneten Planer die Bildverarbeitung beauftragt, das gewünschte Objekt zu suchen. Damit

sich der Tisch im Blickfeld der Kamera befindet, ist mit dem Annäherungspunkt auch eine bestimmte Orientierung des Fahrzeugs verknüpft, bei der die Kamera in etwa auf den Tischmittelpunkt ausgerichtet ist.

Neben der Objektsuche wird mit dem Ausfalten des Arms auch das Greifen des Objekts vorbereitet. Dies leitet die Phase der sogenannten ´Mobilen Manipulation´ ein, in der Manipulatorarm und Fahrzeug gleichzeitig bewegt werden, um eine weiche und ruckfreie Greifbewegung zu erhalten. Die Planung und Regelung dieser Bewegung erfolgt in insgesamt 10 Freiheitsgraden, wobei 7 von dem Manipulatorarm und 3 von der Plattform stammen. In der 2. Phase wird von dem Fahrzeug ein weiterer Annäherungspunkt angesteuert, der von dem Tisch noch so weit entfernt ist, daß das Objekt noch sicher im Bildfeld der Kamera zu sehen ist. Außerdem ist bereits für diesen Annäherungspunkt eine ausgestreckte und für das eigentliche Greifen geeignete Armposition spezifiziert. Im allgemeinen wird auf dem Wegstück zwischen den Annäherungspunkten das Objekt erkannt und seine Position an den Softwaremodul ´Mobile Manipulation´ weitergegeben. In diesem Fall kann ohne Fahrzeugstop eine Greifbewegung bis zum Objekt geplant und ausgeführt werden. Wird das Objekt nicht erkannt, weil es beispielsweise nicht da ist oder verdeckt wird, bleibt das Fahrzeug an dem zweiten Annäherungspunkt stehen. Dort wird weiterhin das Objekt gesucht, bis es entweder gefunden wird oder der Anwender ein neues Kommando gibt.

Bei erfolgreicher Objektsuche wird, ausgehend von dem aktuellen Fahrzeug- und Manipulatorzustand, eine Trajektorie des Greifers zu einem Annäherungspunkt geplant, der bereits relativ zur Objektposition angegeben wird. Diese Trajektorie ist so beschaffen, daß das Objekt, wenn überhaupt, nur kurz durch den Greifer verdeckt wird und deshalb kontinuierlich verfolgt werden kann. Damit ist gewährleistet, daß eine Veränderung der Objektposition erkannt wird und die Trajektorie entsprechend angepaßt werden kann.

Bei einer Verdeckung des Objekts kann zunächst nicht entschieden werden, warum das Objekt nicht mehr erkannt wird. Zum einen kann es in einer Position sein, in der es nicht erkannt, aber auch nicht gegriffen werden kann, zum anderen kann es auch nur verdeckt sein. Im zweiten Fall ist zwar eine Weiterführung des Greifvorganges sinnvoll, nicht aber im ersten; dies kann sogar gefährlich sein, falls das Objekt falsch gegriffen und dadurch beschädigt oder umgeworfen wird. Der einfachste Ausweg ist sicherlich die hier praktizierte Vermeidung einer Verdeckung, die hier durch entsprechende Vorgaben bei der Trajektorienplanung erzielt wird[1]. Dies ermöglicht eine Objektverfolgung bis unmittelbar vor das Objekt und damit eine im Verlauf der Annäherung des Fahrzeugs an das Objekt durch den kleiner werdenden Abstand stetig wachsende Genauigkeit der Objektlokalisierung.

1 Ein möglicher Weg zur Vermeidung dieses Problems könnte eine Hindernisrechnung sein, bei der der Sehstrahl zwischen Objekt und Kamera als Hindernis für den Arm betrachtet wird, das nicht berührt werden darf.

Beim Greifvorgang selbst wird das Objekt zwangsläufig teilweise vom Greifer verdeckt, auch wenn der Greifpunkt so festgelegt wurde, daß dieses Problem möglichst spät auftaucht. Abb. 6.8 zeigt günstige Greifpositionen für ein Glas und eine Flasche. Beim Glas wird der Greifer knapp über dem Tisch an das Objekt geführt, der Sehstrahl verläuft über dem Greifer, Die Flasche wird am Flaschenhals gegriffen, sie ist unterhalb des Greifers sichtbar[1].

Die Objekterkennung endet, sobald der Greifer im Objektmuster sichtbar wird und das sichtbare Muster dadurch stark verändert wird, üblicherweise kurz vor dem eigentlichen Greifen. Die letzten ca. 50 mm bis 100 mm müssen dementsprechend blind zurückgelegt werden. Sind der vom Erkenner verfolgte Teil des Objekts und der gegriffene Teil des Objekts unterschiedlich und wird der vom Erkenner verfolgte Teil deshalb nicht beim Greifen verdeckt, kann das Objekt bis zum Abheben verfolgt werden wie im Beispiel der Flasche.

Die enge Kopplung zwischen dem Planungsmodul für das Fahrzeug in den 10 Freiheitsgraden und der Objekterkennung erfolgt über eine Schnittstelle mit zeitlich äquidistantem Informationsaustausch. Eine Taktrate von 5 Hz ist für die Aufgabe vollkommen ausreichend, im fehlerfreien Fall wäre für die reine Lokalisierung eines unbeweglichen Objekts eine einmalige Nachricht mit der Objektkoordinate ausreichend. Die Informationsübertragung mit 5 Hz dient deshalb nur der reinen Fehlerkorrektur. Wesentlich für die Schnittstelle ist allerdings eine geringe Verzögerungszeit in der Übertragung, da die Verzögerungszeit multipliziert mit der Geschwindigkeit des Fahrzeugs den Fehler in der Lokalisierung ergibt[2]. Dies ist besonders bei den hohen Drehgeschwindigkeiten problematisch, die durch den verwendeten Planungsmodul bei der Regelung des Roboters in 10 Freiheitsgraden auftreten[3].

1 In [46] wird als Lösung des Problems die Verwendung transparenter Greiffinger vorgeschlagen.

2 Der Fehler kann bei bekannter Verzögerungszeit oder Verwendung synchronisierter Zeitbasen und entsprechender Zeitstempel weiter verkleinert werden.

3 Ursache des Problems ist die nicht-holonome Kinematik des Roboterfahrzeugs, die von der Fahrzeugsteuerung als holonom vorausgesetzt wird und deshalb zu sehr schnellen Regelbewegungen führt.

6.1.10 Entfernungsbestimmung

Die in Abschnitt 4.3 beschriebene Parameterbestimmung wurde in einem Experiment anhand der Entfernung getestet. Dazu wurde für die Gewinnung der Objektansichten zusammen mit der für die entsprechende Parametrierung der Baumendknoten notwendigen Entfernungsinformation eine Bahn ähnlich Abb. 6.11 abgefahren.

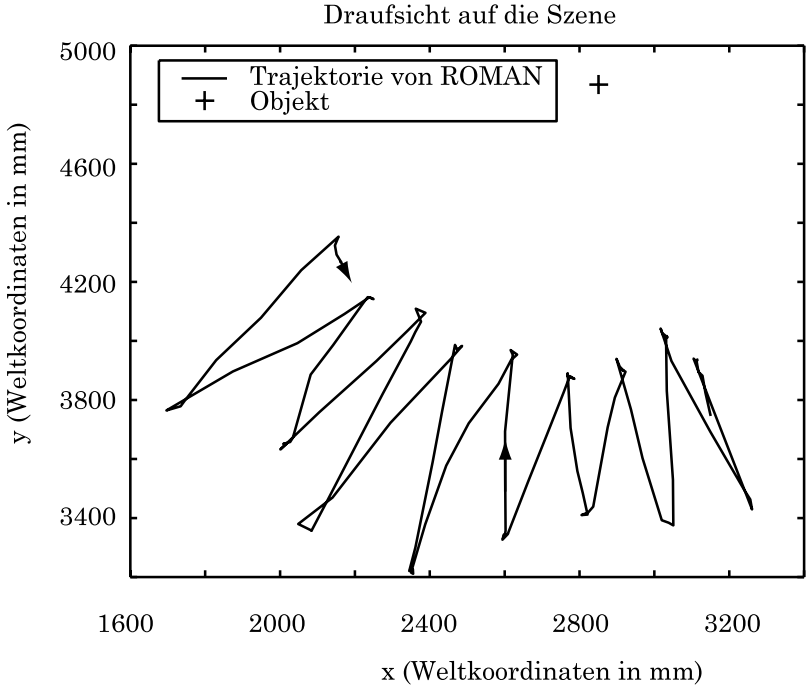

ABBILDUNG 6.11: Fahrzeugbahn bei der Meßfahrt.

Die dort gezeichnete Bahn stammt von dem anschließenden Experiment, in dem die Entfernungsbestimmung getestet wurde. Da eine Einzelmessung einen relativ großen Fehler aufweist, wurde ein Kalman-Filter implementiert, um die direkte Entfernungsmessung mit der Koppelnavigation des Fahrzeugs zu stützen. Damit lassen sich insbesondere einzelne Fehlmessungen eliminieren, die durch die Verzögerungszeit beim Bildeinzug zu großen Meßunsicherheiten führen, wenn sich das Fahrzeug schnell auf das Objekt zu- oder von diesem wegbewegt. Abb. 6.12 und Abb. 6.13 geben das Meßergebnis an, die Standardabweichung für den Fehler des gefilterten Signals ist kleiner als 50 mm. Bei einer Entfernung zwischen Kamera und Objekt von ca. 1000 mm bis 1750 mm entspricht dies einem prozentualen Meßfehler von nur 3% bis 5%. Damit erscheint diese Meßmethode für bestimmte Anwendungen durchaus erfolgversprechend, für die Durchführung der Greifaufgabe ist die Genauigkeit allerdings zu gering.

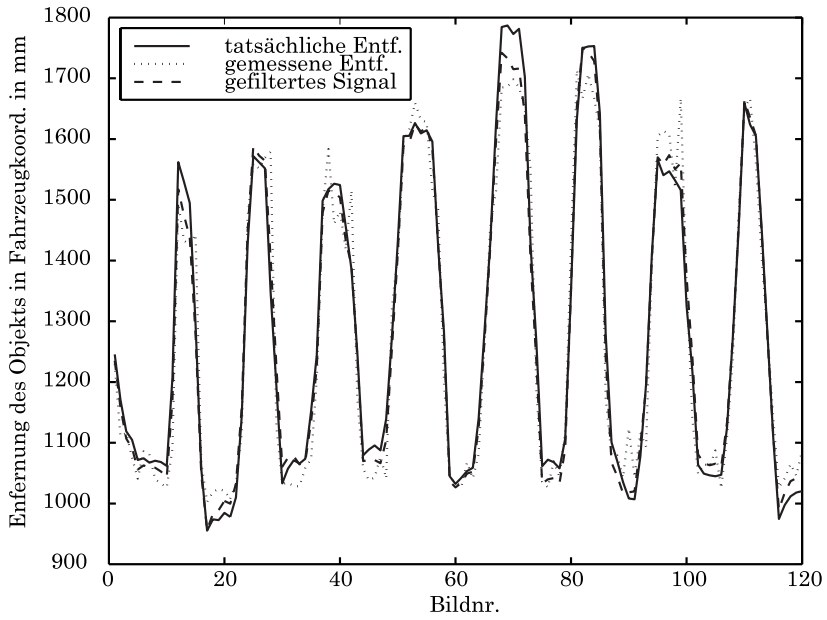

ABBILDUNG 6.12: Zeitlicher Verlauf der Entfernungsmessung.

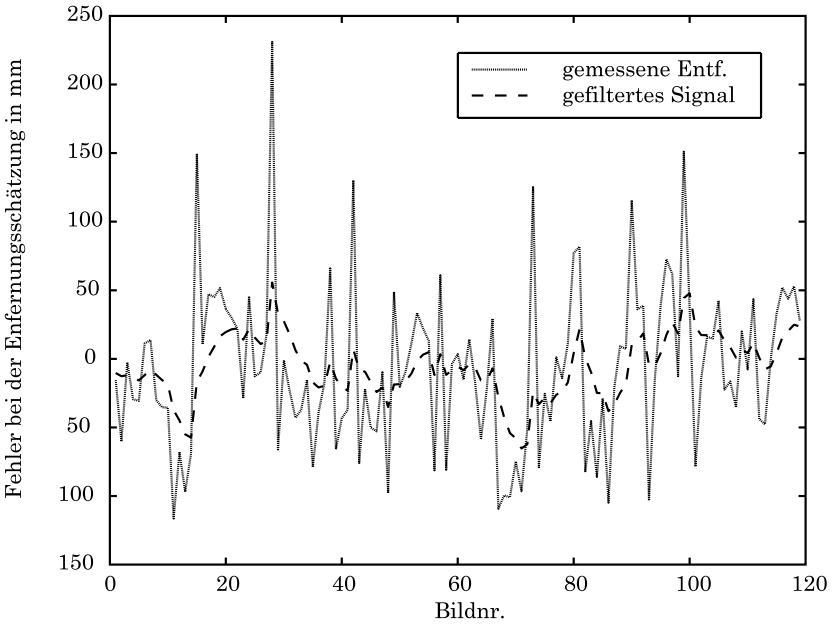

ABBILDUNG 6.13: Fehler der Entfernungsmessung.

6.1.11 Schnittstelle

Die Schnittstelle zwischen Erkennermodul und Fahrzeugsteuerung besteht aus einer Datenstruktur für die Objektanforderung an den Erkennermodul und einer Datenstruktur für die Antwort des Moduls. Die Objektanforderung beinhaltet die Fahrzeug-

position und die Namen der gesuchten Objekte. Es können mehrere Objekte gleichzeitig angegeben werden. Dies ist beispielsweise dann nötig, wenn auf einem Tisch gleichzeitig mehrere Objekte erkannt werden sollen.

Die Datenstruktur zur Rückgabe der erkannten Objekte ist komplizierter. Für jede Objektanforderung getrennt beinhaltet sie folgende Daten:

- Status der Objekterkennung

Status	Bedeutung
1	Objekt ist unbekannt.
2	Objekt ist gefunden.
3	Objekt ist nicht im Bildfeld.
4	Objekt wird noch gesucht.

TABELLE 6.4: Bedeutung der unterschiedlichen Ergebniskennungen.

  Ist keine Hypothese des Objekts vorhanden, wird für eine bestimmte Zeit, ca. 30 sec., der Wert ´4´ ausgegeben. Diese Zeitspanne liegt deutlich über dem Erwartungswert für die Objekterkennung. Wird in dieser Zeit keine robuste Hypothese gefunden, wird anschließend der Wert ´3´ ausgegeben. Für diesen Fall enthält die Fahrzeugsteuerung nach [15] eine Strategie, bei der der Benutzer in die Ausnahmebehandlung eingebunden wird und über das weitere Vorgehen entscheidet.

- Objektposition

  Wird für ein bestimmtes Objekt die Kennung ´2´ zurückgegeben, das Objekt also als erkannt gemeldet, werden die Daten des Objekts an die Fahrzeugsteuerung übermittelt.

Die Schnittstelle wurde auf Basis der sogenannten RPC[1]-Kommunikation implementiert.

1 'Remote Procedure Call', ein von der Firma Sun Microsystems definierter Standard.

6.2 Gesichtsdetektion

6.2.1 Problemstellung

Eine für andere, aktuelle Anwendungen wichtige Funktionalität ist das Erkennen, wo sich im Bildfeld Gesichter befinden. Die Lösung dieser Aufgabe ist beispielsweise Grundlage für die Identifikation der im Bildfeld befindlichen Gesichter. Andere Anwendung finden sich im Kontext der Mensch-Maschine Kommunikation. Eine weitere Einsatzmöglichkeit sind 3D-Bildschirme, die für die korrekte Synthese des 3D-Bildeindrucks die Richtung feststellen müssen, in der sich der Betrachter befindet.

Diese Aufgaben sind durch folgende Probleme gekennzeichnet:

- Der Bildhintergrund weist in realen Anwendungen eine extrem große Vielfalt von Mustern auf, entsprechend Abb. 6.17. Der Hintergrund ist nicht durch das Serviceszenario und die dort üblichen Einrichtungen bestimmt. Der Hintergrund in unmittelbarer Objektumgebung ist ebenfalls variabler und nicht dadurch eingeschränkt, daß das Objekt auf einem standardisierten Möbel liegt.
- Das Gesicht beinhaltet vergleichsweise viele und kleine Strukturen wie die Augen oder die Mundpartie. Es handelt sich um die bereits in Abschnitt 6.1.8 diskutierte Problematik. Diese Strukturen sind außerdem variabel, im Gegensatz zu den starren Objekten des vorhergehenden Abschnitts.
- Um die Bewegungsfreiheit der Person nicht unnötig einzuschränken, soll die Kamera ein großes Bildfeld besitzen. Variiert gleichzeitig die Entfernung des Gesichts zur Kamera stark, ergibt sich eine sehr große Änderung der Gesichtsabbildung. Die in unserem Experiment verwendete Kamera[1] mit Weitwinkelobjektiv besitzt ein laterales Blickfeld von 92° bei 384 Pixel. Entsprechend wird ein Gesicht von 15 cm Breite in einer Entfernung von 250 cm nur ca. 14 Pixel breit abgebildet, in einer Entfernung von 50 cm jedoch 71 Pixel breit. Dies bedeutet einerseits, daß in der großen Entfernung das Gesicht nur einen kleinen Teil des Suchfensters ausfüllt, und andererseits, daß bei kleiner Entfernung das Suchfenster nur einen kleinen Teil des Gesichts bedeckt. Dementsprechend befindet sich in der großen Entfer-

1 Standard Grauwert-Kamera, Vollbild: 768x576 Pixel, entsprechend PAL-Standard. Verwendet wird aber nur eine um den Faktor 2 verringerte Auflösung. CCD-Chip: 1/2"- Sensor, Objektivbrennweite: 4,8 mm.

nung auch ein Teil des Hintergrundes im Suchfenster, während bei der kleinen Entfernung nur der mittlere Ausschnitt des Gesichts als Information zur Verfügung steht. Abb. 6.14 gibt 2 Extrembeispiele wieder.

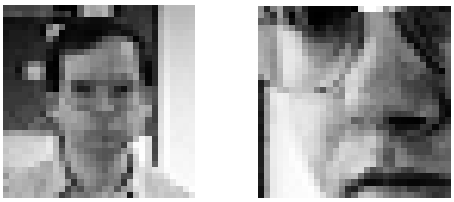

ABBILDUNG 6.14: Extrembeispiele für sehr große und kleine Distanz.

Diese große Spreizung der Entfernung um einen Faktor 5 ist für realistische Anwendungen notwendig, da sich die Situation nicht so genau vorhersehen läßt wie bei dem im wesentlichen festgelegten Greifvorgang aus Abschnitt 6.1.

- Eine weitere Schwierigkeit hinsichtlich der Verfolgung des Gesichts besteht in der willkürlichen Bewegung der Person. Die Bewegungen lassen sich nicht prädizieren wie im Greifszenario. Die Mechanismen für eine schnelle und robuste Objektverfolgung sollen trotzdem eine beliebige Bewegung des Gesichts erlauben.

6.2.2 Training

Ist der Hintergrund teilweise im Suchfenster enthalten, tragen die betreffenden Pixel keine Information über das Objekt und sind deshalb irrelevant. Dementsprechend werden aufgrund der Entropieoptimierung die korrespondierenden Einträge im Gewichtungsvektor auf betragsmäßig kleine Werte gesetzt und damit die betreffenden Pixel nicht oder nur eingeschränkt für die Klassifikation verwendet. Voraussetzung ist hier lediglich, daß eine genügend große Anzahl von Trainingsbildern vorhanden ist. Andernfalls kann durch Zufall der Bildhintergrund an einer bestimmten Stelle immer ähnliche Werte aufweisen, die dann durch die Optimierung modelliert und zur Detektion des Gesichts verwendet werden.

Abb. 6.15 gibt ein Trainingsset für die Gesichtsdetektion wieder, es enthält 120 Einzelbilder. Die Bilder wurden in einem Abstand von 50 cm bis 250 cm aufgenommen, das Gesicht ist jeweils in etwa senkrecht, der Neigungswinkel beträgt maximal 10°. Zusammen mit ca. 100 unterschiedlichen Zurückweisungsbildern, 4 davon sind in Abb. 6.17 gezeigt, ergibt sich ein Baum mit den Kenngrößen entsprechend Tabelle 6.5.

Objekt	Gesicht
Anzahl der Knoten	4171
Anzahl der Parameter (in 1000)	1110
Durchschnittliche/Max. Tiefe	12/21
Typische Zeit für die Objektsuche pro Bild	~200ms

TABELLE 6.5: Größe des Musterbaumes für das Gesicht

6.2.3 Erkennung

Zur Demonstration für die Generalisierung können Gesichtsansichten dienen, die ebenfalls noch erkannt werden. Allerdings kann keine Systematik angegeben werden, welche Gesichter noch erkannt werden und welche nicht. Außerdem werden unterschiedliche Gesichter in bestimmten Situationen, beispielsweise in besonders großer oder kleiner Entfernung, nicht erkannt.

Durch die iterative Vervollständigung der Trainingsmenge nach Abschnitt 4.1.3 können Fehl-Detektion weitgehend ausgeschlossen werden können; diese treten in seltenen, meist sehr extremen Beleuchtungssituationen auf. Abb. 6.18 zeigt eine typische Fehl-Detektion; der Raum wird durch einen Spalt am Bildrand mit Sonnenlicht beleuchtet, wodurch ein kleiner Streifen im Bild sehr hell wird und das restliche Bild durch die AGC[1]- und Blendenregelung der Kamera fast schwarz wird. Die Gegenstände sind nur noch schemenhaft zu erkennen. Die extremen Kontraste sind nicht in der Trainingsmenge der Zurückweisungsklasse enthalten. Entsprechend kann das Objekt nicht mehr korrekt erkannt werden, statt dessen wird durch Zufall ein Bereich an dem hell erleuchteten Spalt als Objekt erkannt. Dies ist ein Hinweis darauf, daß der Objektbereich im Musterraum in bestimmten Richtungen unbeschränkt offen ist und nur durch die Amplitudenbeschränkung der Pixel begrenzt wird, wie in Abschnitt 3.2 bereits diskutiert. Außerdem ist durch die Sättigung der Kamera bei hohen Lichtintensitäten und die damit verbundene *Nichtlinearität* die in Abschnitt 2.4.4.2 hergeleitete, mögliche Invarianz linearer Klassifikatoren gegen unterschiedliche Helligkeit und Kontrast nicht mehr gegeben.

1 Anpassung der Signalverstärkung an die Helligkeit des Bildes.

ABBILDUNG 6.15: Vollständiges Trainingsset für das Gesicht mit 120 Ansichten.

ABBILDUNG 6.16: Ebenfalls noch detektierte Ansichten anderer Personen. An der Position der erfolgreichen Detektion ist das entsprechende Suchfenster eingezeichnet.

ABBILDUNG 6.17: Typische Bilder der Zurückweisungsklasse. Jeder 31x31-Block ist ein Zurückweisungsmuster. Im rechten, unteren Bild ist zur Veranschaulichung ein Block dieser Größe eingezeichnet.

Tabelle 6.6 zeigt die Wahrscheinlichkeit für die Fehl-Detektion und die Nicht-Detektion des trainierten Gesichts. Der Abstandsbereich war derselbe wie im Training, die Beleuchtung ebenfalls eine durchschnittliche Laborbeleuchtung. Es wurden in einem Zeitraum von 117 s bei 5 Bildern/s 580 Bilder ausgewertet. Die Person näherte sich

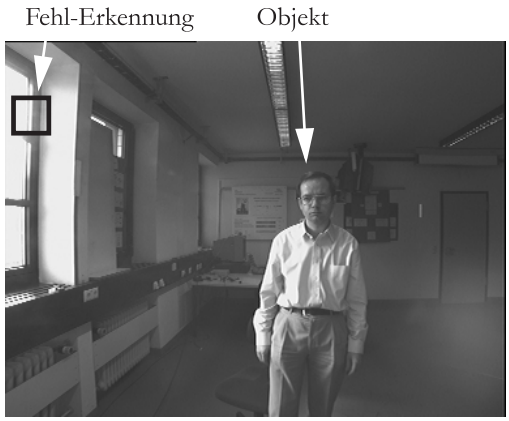

ABBILDUNG 6.18: Typische Fehl-Detektion am linken Bildrand bei partieller Überbelichtung des Fensterbereichs. Durch die automatische Blenden- und Verstärkungssteuerung wird das restliche Bild unterbelichtet.

der Kamera in diesem Zeitraum beginnend bei maximaler Entfernung, 250 cm, bis auf 50 cm. Die Person wurde in der gesamten Videosequenz erkannt. Insgesamt gab es lediglich 7 Nicht-Detektionen. Da diese isoliert auftraten, wurde die Hypothese in dem gesamten Zeitraum stabil aufrecht erhalten.

	Fehl-Detektion	Nicht-Detektion
Anzahl	18	7
Wahrscheinlichkeit	3,1%	1,2%

TABELLE 6.6: Fehl- und Nicht-Detektionen des Gesichts in 580 Bildern.

Während dieser Annäherung gab es 16 zusammenhängende Fehl-Detektionen auf der Kleidung der Person. Diese Fehl-Detektionen führten über einen Zeitraum von ca. 2 s zu einer stabilen Fehl-Hypothese. Außerdem gab es zwei zeitlich isolierte Fehl-Detektionen. Das Experiment zeigt wieder die Tatsache, daß die Wahrscheinlichkeit für Fehler außerordentlich stark von der Situation abhängt. In einer bestimmten Situation traten vergleichsweise viele, zusammenhängende Fehl-Detektionen auf, in der restlichen Stichprobe nur eine verschwindend kleine Anzahl.

Die Genauigkeit der Detektion wird definiert als Winkelabweichung zwischen dem vom Erkenner angezeigten Punkt und dem tatsächlich in den Bildern vorhandenen Mittelpunkt der Nase. Dieser Mittelpunkt der Nase wurde auch in den Trainingsbildern vom Anwender als Mittelpunkt der Objektmuster definiert. Die Genauigkeit ist über den gesamten Entfernungsbereich einheitlich, ihre Standardabweichung beträgt

ca. 3 Pixel, ausgezählt in 573 Bildern. Eine genauere Angabe ist nicht sinnvoll, da der ideale Punkt für die Detektion nur auf ca. 1 Pixel festgelegt werden kann. Dieser Wert entspricht bei oben genannten Werten für Kameraauflösung und Öffnungswinkel einer Genauigkeit von 0,7°. Je nach Entfernung entspricht dieser Winkelfehler einem absoluten Fehler von 6 mm bei einer Entfernung von 50 cm und 30 mm bei 250 cm.

7 Zusammenfassung

In dieser Arbeit wurde ein Verfahren zur schnellen, ansichtsbasierten Objekterkennung entwickelt und experimentell validiert. Als wesentlicher, konzeptioneller Bestandteil wird nicht nur das zu erkennende Objekt betrachtet, sondern zusätzlich auch der *Hintergrund* des Objekts, der eben nicht mit dem Objekt verwechselt werden soll. Damit läßt sich die Objekterkennung auf ein Klassifikationsproblem mit 2 Klassen zurückführen. Die eine Klasse ist die Objektklasse, die andere Klasse - die sogenannte Zurückweisungsklasse - ist der Objekthintergrund.

Die Betrachtung von 2 Klassen erlaubt, die Information über die Klassenzugehörigkeit mittels des *mathematisch-physikalischen Begriffs der Information bzw. Entropie* zu beschreiben. Die Entropie ist ein numerisch berechenbarer Wert für die Unsicherheit bezüglich der Klassenzugehörigkeit eines noch nicht klassifizierten Musters. Damit läßt sich die Entropie als Gütemaß verwenden, um die Parameter eines Klassifikators optimal einzustellen. Vor dem Klassifikationsvorgang ist die Unsicherheit maximal, nach der Klassifikation ist sie 0.

Die Klassifikation basiert *unmittelbar auf den Pixeln* und nicht auf einer Objektrepräsentation mittels Modellmerkmalen wie Ecken, Kanten oder Flächen. Dadurch können beliebige Details des Objekts zur Erkennung verwendet werden. Diese Details werden nicht vorab festgelegt, sondern erst beim Training des Musterbaumes anhand der Entropie. Das Maß der Entropie hat für das gestellte Problem grundsätzliche Bedeutung und findet für jedes Objekt die passenden Parameter. Die Formulierung der Objekterkennung als Maximierung der Information vermeidet eine heuristische Anpassung des Erkenners an bestimmte Objekte mittels Expertenwissen. Statt dessen kann eine große Vielfalt von Objekten ohne manuelle Adaption trainiert und erkannt werden.

Die ansichtsbasierte Musterklassifikation wird als Vektorklassifikation formuliert. Dabei werden neben den bereits bekannten Ansätzen folgende Klassifikatoren und Parametrierungen angegeben:

- Fehlerminimaler, linearer Klassifikator für Normalverteilungen. Ähnlich dem Verfahren nach Fisher [20] werden zwei als analytische Funktionen gegebene Normalverteilungen mittels einem linearen Klassifikator getrennt. Im Gegensatz zu der Lösung nach Fisher wird ein halbanalytisches Verfahren angegeben, das zu einer fehlerminimalen Lösung führt.
- Entropieoptimale, lineare und quadratische Klassifikatoren für Vektormengen. Ausgehend von der analytischen Berechnung der Entropie und ihrer Ableitung nach den Klassifikatorparametern führt ein numerisches Verfahren zu einer entropieoptimalen Einstellung der Parameter; die Optimierung wird anhand zweier Vektormengen durchgeführt.

Für die Anwendung im Kontext der Objekterkennung wird der lineare, entropieoptimale Klassifikator eingesetzt, da seine Eigenschaften eine bessere Generalisierungsfähigkeit der Objekterkennung vermuten lassen. Er führt zu den gewünschten Resultaten hinsichtlich der Klassifikationsgeschwindigkeit, darüber hinaus ist der technische Aufwand geringer verglichen mit quadratischen Klassifikatoren; sein Einsatz in einem Musterbaum ermöglicht durch stückweise lineare Approximation die Nachbildung beliebiger Trennflächen und stellt somit keinen Nachteil dar.

Diese Überlegungen fließen in die Konstruktion eines sehr schnellen und effizienten Objekterkennungswerkzeuges mit folgenden Einzelschritten ein:

- Datengewinnung. Der Anwender muß für eine repräsentative Auswahl von Ansichten des Objekts und des möglichen Hintergrundes sorgen. Seine Arbeit besteht lediglich in der Aufnahme von typischen Ansichten und einer grafisch unterstützten Markierung des gewünschten Objekts.
- Training. Die vom Anwender bereitgestellten Ansichten sind Ausgangspunkt für den automatischen Aufbau des Musterbaumes. Dieser Musterbaum ist ein binärer Entscheidungsbaum. Er enthält in jedem Knoten einen linearen Klassifikator. Die Verwendung der Musterbäume führt zu nur logarithmisch mit der Problemgröße anwachsenden Klassifikationszeiten. Jeder dieser linearen Klassifikatoren wird anhand der Entropie so parametriert, daß er die Unsicherheit über die Klassenzugehörigkeit eines Musters möglichst stark verkleinert. Die Folge dieser Optimierung sind Musterbäume, die im Vergleich zu anderen Parametrierungsmethoden wenige Knoten besitzen.
- Erkennung. Für die Erkennung wird der im Training generierte Musterbaum auf ein neues Bild angewandt. Über den relevanten Ausschnitt des Bildes wird ein Suchfenster geschoben und nach jeder Verschiebung der Inhalt des Suchfensters als Objekt oder Hintergrund klassifiziert. Ergebnis der Erkennung ist die Position des Objekts in Bildkoordinaten.
- Experimentelle Validierung. Testweise kann der Anwender den Klassifikator auf das Erkennungsproblem anwenden und untersuchen, ob Verwechslungen von Objekt- und Zurückweisungsklasse auftreten. Gegebenenfalls können die betreffenden Ansichten für einen zweiten, verbesserten Trainingslauf zu der originalen Trainingsmenge hinzugefügt werden. Dieser Vorgang erfordert ebenfalls einfache, grafisch unterstützte Arbeit.

Folgende Methoden ermöglichen eine Objekterkennung in Videoechtzeit und damit den Einsatz des Verfahrens in Realzeitsystemen:

- Binärbaum. Durch die Verwendung des Binärbaumes wachsen die Rechenzeiten nur logarithmisch mit der Problemgröße, auch komplexe Objekte wie das Gesicht können so schnell erkannt werden.

- Unterabtastung. Die Steuerung der Unterabtastung dient der drastischen Erhöhung der Klassifikationsgeschwindigkeit. Für die Grobklassifikation in den ersten Schichten des Baumes reicht eine geringe Auflösung, die in tieferen Schichten erhöht wird.
- Suchstrategie. Die spiralförmige Objektsuche führt zu einem optimal dimensionierten Suchbereich. Bei gegebener Rechenleistung und Bildrate kann die für die Objektverfolgung theoretisch erlaubte Objektbeschleunigung a auch tatsächlich kompensiert werden.

Die Unterabtastung kann so parametriert werden, daß keine Anpassung an bestimmte Objekte notwendig ist. Sie erlaubt eine extrem schnelle und robuste Klassifikation und ermöglicht den Einsatz der ansichtbasierten Objekterkennung in realen Anwendungen.

Für diese Anwendungen müssen jeweils geeignete Trainingsmengen an Beispielbildern aufgenommen werden, die das Objekt und seinen typischen Hintergrund für alle Situationen repräsentieren, die in der Anwendung vorkommen. Die geometrischen Freiheitsgrade und die Beleuchtung müssen so variiert werden, wie sie sich auch bei der Erkennung verändern können. Allerdings existieren keine theoretischen Erkenntnisse, wie fein diese Parameter gerastert werden müssen, um eine Erkennung auch in den Zwischenpunkten sicherzustellen. Das gewählte Verfahren stellt allerdings sicher, daß wenigstens die Trainingsvektoren korrekt klassifiziert werden. Die in der Arbeit vorgestellte Methode für extrem große Zurückweisungsklassen mit bis zu 10^8 Vektoren ist für viele Anwendungen in der Lage, den Bildhintergrund in ausreichender Vielfalt zu erfassen.

Die wesentlichen Einschränkungen des ansichtsbasierten Erkenners bestehen in der *Generalisierungsfähigkeit*, so daß verhältnismäßig viele Ansichten des Objekts benötigt werden. Objekte mit vielen geometrischen Freiheitsgraden lassen sich kaum mit dieser Methode erkennen, in vielen Anwendungen ist aber die Anzahl der Freiheitsgrade auf 3 bis 4 beschränkt. Da 2 Freiheitsgrade bereits durch die 2-dimensionale Suche im Bild abgedeckt werden, müssen durch die Trainingsmenge nur 1 bis 2 Freiheitsgrade erfaßt werden, was für praktische Probleme oft ausreicht.

Der gesamte Erkenner wurde experimentell in 2 unterschiedlichen Anwendungen erprobt, der Erkennung von Objekten für den Serviceroboter ROMAN und der Detektion von Gesichtern, wobei nicht zwischen verschiedenen Gesichtern unterschieden wird. Die Experimente zeigen sowohl die Echtzeitfähigkeit als auch die Robustheit der Erkennung, die auch in größeren, integrierten Experimenten eingesetzt wird und innerhalb einer Regelschleife das Greifen von Objekten ermöglicht.

Anhang A: Notation

$\underline{x}$	Vektor allgemein
$\underline{x}_{kn}$	n. Vektor der Klasse k
N	Anzahl aller Vektoren in einer Trainingsmenge
N_k	Anzahl der Vektoren von Klasse k
Δ	Summe aller gewichteten Vektoren
Δ_k	Summe der gewichteten Vektoren der Klasse k
K	Anzahl der Klassen in einer Trainingsmenge
T	Transformationsmatrix
C	Kovarianzmatrix einer Vektormenge
L	Inverse der Kovarianzmatrix
A	quadratische Gewichtsmatrix eines Klassifikators
$\underline{c}$	quadratischer Gewichtsvektor eines Klassifikators
$\underline{w}$	linearer Gewichtsvektor eines Klassifikators
a	Offset eines Klassifikators
$\underline{z}$	quadratische Terme eines Klassifikators in Vektorschreibweise oder Vektor allgemein (in Anhang C)
$\underline{s}$, $\underline{u}$, $\underline{y}$	Vektoren in diversen Koordinatensystemen (in Anhang C)
$\underline{m}$	Mittelpunkt einer Verteilung
D	Diagonalmatrix
p	Wahrscheinlichkeit, allgemein
p_k	Wahrscheinlichkeit der Klasse k
$p(i)$	Wahrscheinlichkeit der Entscheidung i (bei Klassifikatoren)
$g(\underline{x})$	Wahrscheinlichkeitsdichtefunktion
I	Information
H	Entropie
f	Brennweite der Kamera
G	Knotenanzahl des Musterbaumes
E	Endknotenanzahl des Musterbaumes
i	Bildindex in einer Videosequenz
γ	Interpolationsvariable
s	Schicht in einem Musterbaum
η	Konstante für die stetig differenzierbare Approximation der *sgn*-Funktion
r	Reduktionsfaktor für die Auflösung

m	Seitenlänge eines Musters
$\underline{1}$	Einsvektor $\left[1 \ldots 1\right]^T$
σ	Schrittweite bei der Optimierung
ν	Index für die Optimierung
$\delta(x)$	Dirac-Funktion

Anhang B: Approximation einer Vektormenge mittels Normalverteilung

Die analytische Approximation einer Vektormenge soll wesentliche Merkmale derselben erhalten. Dies sind insbesondere die Momente 1. und 2. Ordnung, also der Schwerpunktsvektor $\underline{m}$ und die Kovarianzmatrix C. Bei der Approximation durch eine Normalverteilung tauchen diese beiden Größen explizit in der Funktionsgleichung auf. Sie werden wie folgt berechnet:

$$\underline{m} = \frac{1}{N}\sum_{i=1}^{N} \underline{x}_i$$

$$C = \frac{1}{N}\sum_{i=1}^{N} \underline{x}_i \cdot \underline{x}_i^{\,T} - \underline{m} \cdot \underline{m}^{T}$$

Damit ergibt sich die Normalverteilung:

$$g(\underline{x}) = \frac{1}{\sqrt{(2\pi)^n |det(C)|}} e^{-\frac{1}{2}(\underline{x}-\underline{m})^T C^{-1} (\underline{x}-\underline{m})}$$

Anhang C: Integrale für den linearen Klassifikator bei Normalverteilungen

C.1 Skalarwertiges Integral

Zunächst werde das Integral $q = \int_{\mathbb{R}^n} g(\underline{x})\delta(\underline{w}^T\underline{x} + a)d\underline{x}$ berechnet, es wird für die Lösung von Gl. 2.13 benötigt. Die Substitution $L = C^{-1}$ führt zu der Verteilungsfunktion

$$g(\underline{x}) = \frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}(\underline{x}-\underline{m})^T L(\underline{x}-\underline{m})}$$

Substituiert man $\underline{s} = \underline{x} - \underline{m}$, erhalten wir

$$g(\underline{x}) = \frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}\underline{s}^T L\underline{s}}$$

und mit $\underline{w}^T\underline{m} = c$ die neue Trennebene

$$\delta(\underline{w}^T\underline{x} + a) = \delta(\underline{w}^T(\underline{s} + \underline{m}) + a) = \delta(\underline{w}^T\underline{s} + \underline{w}^T\underline{m} + a) = \delta(\underline{w}^T\underline{s} + c + a)$$

Es gilt zunächst die Modaltransformation in die Modalkoordinaten $\underline{z}$

$$\underline{s} = M\underline{z}$$

wobei die Modalmatrix M eine reine Drehmatrix darstellt und deshalb gilt:

$$det(M) = 1$$

Damit ergibt sich die neue Trennebene

$$\underline{w}^T\underline{s} + c + a = \underline{w}^T M\underline{z} + c + a = \underline{v}^T\underline{z} + c + a$$

mit dem neuen Vektor

$$\underline{v} = M^T\underline{w}$$

und das Integral

$$q = \frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} \int_{I\!R^n} e^{-\frac{1}{2}\underline{z}^T D\underline{z}} \delta(\underline{v}^T\underline{z} + c + a) d\underline{z}$$

In $\underline{z}$-Koordinaten beschreibt auch diese Darstellung ein achsenparalleles Ellipsoid mit der Diagonalmatrix D, das durch eine weitere Matrix $K = Diag(\sqrt{d_i^{-1}})$ mit den Diagonalelementen d_i der Matrix D in eine kugelsymmetrische Verteilung in $\underline{y}$-Koordinaten transformiert wird. Daraus folgt der Zusammenhang

$$detK = \frac{1}{\sqrt{detL}} \tag{C.1}$$

Es gelte für die neuen Koordinaten $\underline{y}$:

$$\underline{z} = K\underline{y}$$

und damit die Trennebene

$$\underline{v}^T\underline{z} + c + b = \underline{v}^T K\underline{y} + c + b = \underline{t}^T\underline{y} + c + b$$

mit dem Vektor

$$\underline{t} = K^T\underline{v} = K\underline{v}$$

Als Integral ergibt sich damit

$$q = \left(\frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} \int_{I\!R^n} e^{-\frac{1}{2}\underline{y}^T\underline{y}} \delta(\underline{t}^T\underline{y} + c + a) d\underline{y} \right) \cdot \mu$$

Der Streckungsfaktor μ berechnet sich allgemein zu $det(K)$. Da aber aufgrund der δ-Funktion senkrecht zu der betrachteten Trennebene keine Streckung stattfindet, darf diese Streckung nicht berücksichtigt werden. Sie ist betragsmäßig $\frac{|K\underline{v}|}{|\underline{v}|}$, damit ergibt sich also mit ihrem Kehrwert

$$q = det(K)\frac{|\underline{v}|}{|K\underline{v}|}\frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} \int_{I\!R^n} e^{-\frac{1}{2}\underline{y}^T\underline{y}} \delta(\underline{t}^T\underline{y} + c + a) d\underline{y}$$

Da die Verteilung $e^{-\frac{1}{2}\underline{y}^T\underline{y}}$ kugelsymmetrisch ist, darf statt des Trennvektors $\underline{t}$ ein beliebiger Trennvektor verwendet werden, die Trennebene muß nur denselben Abstand zum Ursprung aufweisen. Dieser ist gegeben durch $\frac{c+a}{|\underline{t}|}$, man darf also

$$q = det(K)\frac{|\underline{v}|}{|K\underline{v}|}\frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}}\int_{I\!R^n} e^{-\frac{1}{2}\underline{y}^T\underline{y}}\delta\left(y_1 - \frac{c+a}{|\underline{t}|}\right)d\underline{y} \tag{C.2}$$

schreiben mit y_1 als der ersten Koordinate des Vektors $\underline{y}$. Damit ergibt sich als Integral der Wert

$$q = det(K)\frac{|\underline{v}|}{|K\underline{v}|}\frac{\sqrt{|detL|}}{\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\left(\frac{c+a}{|\underline{t}|}\right)^2},$$

der mit Gl. C.1 noch vereinfacht werden kann zu

$$q = \frac{|\underline{w}|}{|KM^T\underline{w}|}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\left(\frac{\underline{w}^T\underline{m}+a}{|KM^T\underline{w}|}\right)^2}$$

Der Term $|KM^T\underline{w}|$ kann vereinfacht werden, es gilt:

$$|KM^T\underline{w}| = \sqrt{(KM^T\underline{w})^T\cdot(KM^T\underline{w})} = \sqrt{\underline{w}^TMD^{-1}M^T\underline{w}} = \sqrt{\underline{w}^TC\underline{w}}$$

und damit

$$q = \frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^TC\underline{w}}}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m}+a)^2}{\underline{w}^TC\underline{w}}} \tag{C.3}$$

Es wird also weder die Modalmatrix M noch die Diagonalform D für das Endergebnis benötigt.

C.2 Vektorwertiges Integral

Das zweite, vektorwertige Integral $\underline{p} = \int_{I\!R^n} g(\underline{x})\delta(\underline{w}^T\underline{x}+a)\underline{x}d\underline{x}$ wird ähnlich gelöst, mit den Substitutionen $L = C^{-1}$ und $\underline{s} = \underline{x} - \underline{m}$ erhalten wir die Trennebene

$$\delta(\underline{w}^T\underline{x}+a) = \delta(\underline{w}^T(\underline{s}+\underline{m})+a) = \delta(\underline{w}^T\underline{s}+\underline{w}^T\underline{m}+a) = \delta(\underline{w}^T\underline{s}+c+a)$$

und das Integral

$$\underline{p} = \frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} \int_{I\!R^n} e^{-\frac{1}{2}s^T Ls} \delta(\underline{w}^T \underline{s} + c + a)(\underline{s} + \underline{m}) d\underline{s}$$

Dies kann als Summe der beiden Terme

$$\underline{p}_1 = \frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} \int_{I\!R^n} e^{-\frac{1}{2}\underline{s}^T L\underline{s}} \delta(\underline{w}^T \underline{s} + c + a)\underline{s} d\underline{s}$$

und

$$\underline{p}_2 = \frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} \underline{m} \int_{I\!R^n} e^{-\frac{1}{2}\underline{s}^T L\underline{s}} \delta(\underline{w}^T \underline{s} + c + a) d\underline{s}$$

geschrieben werden, wobei mit Gl. C.3 für $\underline{p}_2$ gilt:

$$\underline{p}_2 = \underline{m} \frac{\sqrt{\underline{w}^T \underline{w}}}{\sqrt{\underline{w}^T C \underline{w}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \cdot \frac{(\underline{w}^T \underline{m} + a)^2}{\underline{w}^T C \underline{w}}}$$

Die Berechnung von $\underline{p}_1$ ist etwas komplexer:
Wird wiederum eine Modaltransformation

$$\underline{s} = M\underline{z}$$

durchgeführt, erhalten wir

$$\underline{p}_1 = \int_{I\!R^n} \left(\frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}\underline{z}^T D\underline{z}} \right) \delta(\underline{w}^T M\underline{z} + c + a) M\underline{z} d\underline{z}$$

Durch die Transformation mit der bereits bekannten Matrix K ergibt sich weiter

$$\underline{z} = K\underline{y}$$

und damit

$$\underline{p}_1 = det(K)\frac{|\underline{v}|}{|K\underline{v}|}\int_{I\!R^n}\left(\frac{\sqrt{|detL|}}{\sqrt{(2\pi)^n}}e^{-\frac{1}{2}\underline{y}^T\underline{y}}\right)\delta(\underline{w}^T MK\underline{y} + c + a)MK\underline{y}d\underline{y}$$

Wie bereits bekannt, gilt $det(K) = \sqrt{|detL|}^{-1}$ und damit

$$\underline{p}_1 = \frac{1}{\sqrt{(2\pi)^n}}\frac{|\underline{v}|}{|K\underline{v}|}MK\int_{I\!R^n} e^{-\frac{1}{2}\underline{y}^T\underline{y}}\delta(\underline{t}^T\underline{y} + c + a)\underline{y}d\underline{y}$$

Zur Lösung des Integrales wird eine Drehmatrix M_2 eingeführt [1], für die gilt:

$$M_2\left[1\ 0\ \ldots\ 0\right]^T = \frac{\underline{t}}{|\underline{t}|} \quad \text{bzw.} \quad \underline{t}^T M_2 = \left[|\underline{t}|\ 0\ \ldots\ 0\right] \tag{C.4}$$

Diese Matrix wird zu einer weiteren Koordinatentransformation verwendet, es gelte:

$$M_2\underline{u} = \underline{y}$$

In $\underline{u}$ lautet nun das Integral

$$\underline{p}_1 = \frac{1}{\sqrt{(2\pi)^n}}\frac{|\underline{v}|}{|K\underline{v}|}MK\int_{I\!R^n} e^{-\frac{1}{2}\underline{u}^T\underline{u}}\delta(\underline{t}^T M_2\underline{u} + c + a)M_2\underline{u}d\underline{u}$$

oder mit Gl. C.4 vereinfacht

$$\underline{q} = \frac{1}{\sqrt{(2\pi)^n}}\frac{|\underline{v}|}{|K\underline{v}|}MKM_2\int_{I\!R^n} e^{-\frac{1}{2}\underline{u}^T\underline{u}}\delta(|\underline{t}|u_1 + c + a)\underline{u}d\underline{u}$$

Unter dem Integral steht ein Vektor, entsprechend kann elementweise integriert werden. Zunächst werde das erste Element integriert, es gilt:

$$\int_{I\!R^n} e^{-\frac{1}{2}\underline{u}^T\underline{u}}\delta\left(u_1 + \frac{c + a}{|\underline{t}|}\right)u_1 d\underline{u}$$

Integriert man nun die Variablen $u_i, i \in \{1, 2, \ldots, N\}$ getrennt, ergibt sich für u_1 das Integral

1. Die Matrix M_2 ist nicht eindeutig bestimmt, fällt aber im weiteren Verlauf der Rechnung wieder heraus und muß deshalb nicht näher festgelegt werden.

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2}u_1^2} \delta\left(u_1 + \frac{c+b}{\|\underline{t}\|_2}\right) u_1 du_1 = -e^{-\frac{1}{2}\left(\frac{c+a}{|\underline{t}|}\right)^2} \frac{c+a}{|\underline{t}|}$$

und für die restlichen Variablen

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2}u_i^2} du_i = \sqrt{2\pi},\ i \neq 1\,,$$

so daß sich insgesamt als Wert ergibt:

$$-e^{-\frac{1}{2}\left(\frac{c+b}{\|\underline{t}\|_2}\right)^2} \frac{c+a}{|\underline{t}|} \sqrt{2\pi}^{n-1}$$

Für die restlichen Einträge $i \in \{2, 3, \ldots, N\}$ des Vektors erhalten wir bei Integration

$$\int_{I\!R^n} e^{-\frac{1}{2}u_i^2} u_i d\underline{u} = 0$$

Es ergibt sich also insgesamt

$$M_2 \int_{I\!R^n} e^{-\frac{1}{2}\underline{u}^T\underline{u}} \delta\left(u_1 + \frac{c+a}{|\underline{t}|}\right) \underline{u} d\underline{u} = M_2 \begin{bmatrix} -e^{-\frac{1}{2}\left(\frac{c+a}{|\underline{t}|}\right)^2} \left(\frac{c+a}{|\underline{t}|}\right) \sqrt{2\pi}^{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

und mit Gl. C.4 ergibt dies den Ausdruck

$$-e^{-\frac{1}{2}\left(\frac{c+a}{|\underline{t}|}\right)^2} \left(\frac{c+a}{|\underline{t}|}\right) \sqrt{2\pi}^{n-1} \cdot \frac{\underline{t}}{|\underline{t}|}\,,$$

für das Integral ergibt sich damit:

$$-\frac{1}{\sqrt{(2\pi)^n}} \frac{|\underline{v}|}{|K\underline{v}|} MK e^{-\frac{1}{2}\left(\frac{c+a}{|\underline{t}|}\right)^2} \left(\frac{c+a}{|\underline{t}|}\right) \sqrt{2\pi}^{n-1} \cdot \frac{\underline{t}}{|\underline{t}|}$$

Wird $\underline{t}$ durch die ursprünglichen Größen ausgedrückt, erhalten wir

$$-\frac{1}{\sqrt{2\pi}}\frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C\underline{w}}}MKe^{-\frac{1}{2}\frac{(c+a)^2}{\underline{w}^T C\underline{w}}}\left(\frac{c+a}{\sqrt{\underline{w}^T C\underline{w}}}\right)\cdot\frac{\underline{t}}{\sqrt{\underline{w}^T C\underline{w}}}$$

und damit

$$\underline{p}_1 = -\frac{1}{\sqrt{2\pi}}\frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C\underline{w}}}e^{-\frac{1}{2}\frac{(\underline{w}^T\underline{m}+a)^2}{\underline{w}^T C\underline{w}}}\left(\frac{\underline{w}^T\underline{m}+a}{\sqrt{\underline{w}^T C\underline{w}}}\right)\cdot\frac{C\underline{w}}{\sqrt{\underline{w}^T C\underline{w}}}$$

Für die Summe aus $\underline{p}_1$ und $\underline{p}_2$ ergibt sich also:

$$\underline{p} = \underline{m}\frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C\underline{w}}}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m}+a)^2}{\underline{w}^T C\underline{w}}} - \frac{1}{\sqrt{2\pi}}\frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C\underline{w}}}e^{-\frac{1}{2}\frac{(\underline{w}^T\underline{m}+a)^2}{\underline{w}^T C\underline{w}}}\left(\frac{\underline{w}^T\underline{m}+a}{\sqrt{\underline{w}^T C\underline{w}}}\right)\cdot\frac{C\underline{w}}{\sqrt{\underline{w}^T C\underline{w}}}$$

Die Gleichung kann etwas vereinfacht werden:

$$\underline{p} = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\cdot\frac{(\underline{w}^T\underline{m}+a)^2}{\underline{w}^T C\underline{w}}}\cdot\frac{\sqrt{\underline{w}^T\underline{w}}}{\sqrt{\underline{w}^T C\underline{w}}^3}(\underline{m}\underline{w}^T - (\underline{w}^T\underline{m}+a)E)C\underline{w} \qquad \text{(C.5)}$$

Anhang D: Entfernungsbestimmung im Serviceszenario

Die Entfernungsbestimmung wird mittels Vorinformation über den Objektort gewonnen. Bekannt ist im vorliegenden Szenario jeweils die Objekthöhe, da das Objekt durch das Fahrzeugkommando teilweise eingeschränkt ist, beispielsweise 'nehme das Objekt vom Tisch'. In diesem Fall kann die bekannte Tischhöhe als Höhe des Objekts angenommen werden, durch Schneiden der bekannten Linie Objekt-Kamera mit der Tischebene kann die fehlende Entfernungsinformation gewonnen und die 3D-Koordinate des Objekts in Fahrzeugkoordinaten bestimmt werden.

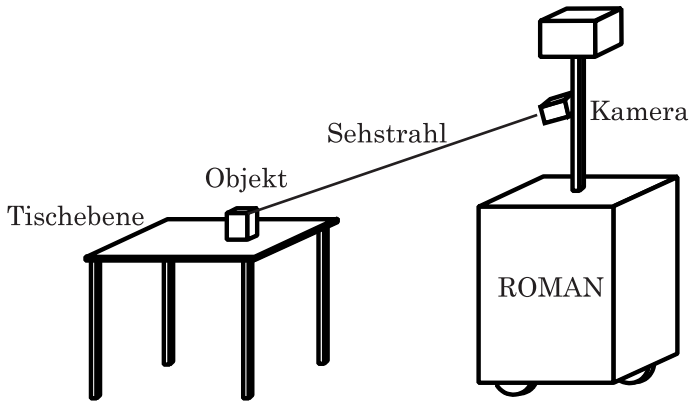

ABBILDUNG 7.1: Entfernungsbestimmung durch Schneiden des Sehstrahls mit der Tischebene.

Das Verfahren ist relativ einfach und genau, es lassen sich in dem beschriebenen Experiment Genauigkeiten von 5 mm erreichen, erfordert aber zusätzliches, geometrisches Vorwissen. Wesentliche Voraussetzung ist die präzise Kenntnis der Sehstrahllage in Fahrzeugkoordinaten, insbesondere wirken sich Winkelfehler des Sehstrahls bei einem kleinem Schnittwinkel zwischen Sehstrahl und Bezugsebene sehr stark aus; auf eine genaue Kamerabefestigung bzw. einen genauen Kameraneigekopf ist zu achten. Deshalb werden bei der Kamerakalibrierung die externen Parameter 'Neigung' bzw. 'Kamerahöhe' und die internen Parameter der Kamera zusammen und nicht getrennt kalibriert.

Das in Kapitel 4.3 beschriebene, direkte Verfahren für die Entfernungsbestimmung erfordert prinzipiell kein geometrisches Vorwissen, für die Greifaufgabe im Serviceszenario ist es allerdings zu ungenau. Ebenfalls zu ungenau ist das Bewegungsstereo-Verfahren [73], da die Kenntnis der Stereobasis von der wenig präzisen Navigation des Fahrzeugs abhängt. Des weiteren gehen Kalibrierfehler der Kamera sehr stark in die Entfernungsbestimmung ein, außerdem ist das Verfahren auf ruhende Objekte begrenzt.

Als weitere, aber in Hinblick auf den Hardwareaufwand aufwendige Alternative, kommt das direkte Stereoverfahren mit 2 Kameras in Frage. Inwieweit dieses Verfahren mit der ansichtsbasierten Objekterkennung kombiniert werden kann, ist nicht bekannt.

Eine andere Methode, das 'visual servoing', vermeidet die explizite Bestimmung der Objektentfernung, indem die Relation zwischen Greifer und Objekt in Bildkoordinaten bestimmt wird. Das Verfahren läßt sich vorteilhaft mit zwei Kameras einsetzen, einen ausführlichen Überblick gibt [6].

Anhang E: Kamerakalibrierung

Obwohl die Kamera des mobilen Serviceroboters ROMAN nur geringe Verzerrung aufweist, wird sie zum Erhöhen der Genauigkeit auf das für die Greifaufgabe notwendige Maß kalibriert. Zu diesem Zweck werden die Bildkoordinaten der realen Kamera x_r, y_r in die Koordinaten x_i, y_i einer idealen Lochkamera umgerechnet. Dies geschieht mit Polynomen mit Termen bis 3. Ordnung:

$$x_i = c_1 + c_2 x_r + c_3 x_r^2 + c_4 x_r^3 + c_5 x_r \cdot y_r$$

$$y_i = c_6 + c_7 y_r + c_8 y_r^2 + c_9 y_r^3 + c_{10} y_r \cdot x_r$$

Damit können die wichtigsten Verzerrungen ausgeglichen werden, allerdings ist, bis auf die Parameter c_1 und c_6, keine direkte physikalische Interpretation möglich wie bei anderen Kalibrationsverfahren, beispielsweise nach [37]. Ein weiterer Nachteil besteht in der fehlenden, exakten Rücktransformation, die jedoch für unser Verfahren zur Objekterkennung nicht benötigt wird. Vorteile der polynomialen Kalibration sind die einfache Erweiterungsmöglichkeit um weitere Terme für Kameras mit stärkeren Verzerrungen, insbesondere Weitwinkelkameras, und die große Anzahl der Parameter, die relativ viele Fehler ausgleichen können.

Neben den inneren Parametern werden noch zwei externe Parameter berücksichtigt. Dies sind die Höhe der Kamera im Fahrzeugkoordinatensystem h und ihre Neigung α Anderen Parameter wie die Verdrehung der Kamera um die optische Achse oder ein lateraler Versatz auf dem Fahrzeug können aufgrund der präzisen Montage vernachlässigt werden.

Um diese insgesamt 12 Parameter zu bestimmen, wird ein ebener Kalibrierkörper bekannter Geometrie aus unterschiedlichen Perspektiven aufgenommen. Der Körper befindet sich dazu jeweils in horizontaler Lage in einer Höhe h_1 bzw. h_2; es werden jeweils ca. 25 Aufnahmen aufgenommen. Die unterschiedliche Höhe ist notwendig, um eine Tiefenabhängigkeit der Kalibration zu erhalten.

Die eigentliche Bestimmung der Parameter wird durch quadratische Ausgleichsrechung durchgeführt, wobei zur Initialisierung der inneren Parameter der fehlerfreie Fall verwendet wird:

$$c_1 = c_6 = 0 \quad c_2 = c_7 = 1 \quad c_3 = c_4 = c_5 = c_8 = c_9 = c_{10} = 0$$

Die Initialisierung der externen Parameter h, α wird mittels der manuell gemessenen Werte durchgeführt. Für die Ausgleichsrechung werden die Abstände von 4 Marken auf dem Kalibrierkörper über die Kamera und die optischen Abbildungsgesetze gemessen, der aufsummierte, quadratische Fehler ist direkt das Gütemaß für die Ausgleichsrechung. In diese Abbildungsgesetze geht als weiterer Parameter die Brennweite des Kameraobjektivs ein, der auf seinen nominalen Wert festgelegt wird.

Literaturverzeichnis

[1] Aladjem, M.E., Two-Class Pattern Discrimination via Recursive Optimization of Patrick-Fisher Distance, Proc. of the 13th Int. Conf. on Pattern Recognition ICPR 1996, Vienna, Austria, 25.-29. Aug. 1996.

[2] Bischof, W. F., Caelli, T., Visual Learning of Patterns and Objects, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 27, No. 6, December 1997.

[3] Bronstein, I.N., Semendjajew, K.A., Taschenbuch der Mathematik, 22. Auflage, Moskau, Leipzig, 1985.

[4] Brown, R.L., Accelerated Template Matching Using Template Trees Grown by Condensation, IEEE Transactions on Systems, Man and Cybernetics, Vol. 25, No. 3, März 1995.

[5] Chou, P., Optimal Partitioning for Classification and Regression Trees, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vo. 13, No. 4, April 1991.

[6] Corke, P. I., Visual Control of Robots: High Performance Visual Servoing, Research Studies Press Ltd, Baldock Hertfordshire, England, 1996.

[7] Daxwanger, W.; Ettelt, E.; Fischer, C.; Freyberger, F.; Hanebeck, U. und Schmidt, G.: ROMAN-Ein mobiler Serviceroboter als persönlicher Assistent in belebten Innenräumen. 12. Fachgespräch Autonome Mobile Systeme, München, S. 314-333, 1996.

[8] Djouadi, A., Bouktache, E., A Fast Algorithm for the Nearest-Neighbor Classifier, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 3, March 1997.

[9] Ettelt, E. and Schmidt, G.: Optimization of Template Trees Based on Information Theory. - In: Proc. of the 6th European Congress on Intelligent Techniques an Soft Computing (EUFIT), Aachen, Germany, Sept. 7-10, 1998.

[10] Ettelt, E. and Schmidt, G.: Optimized Templates Trees for Appearance Based Object Recognition. - In: Proc. of the Intl. Conf. on Systems, Man and Cybernetics, San Diego, Calif., USA, Oct. 11-14, 1998.

[11] Ettelt, E., Furtwängler R.; Hanebeck, U. D. and Schmidt, G.: Design Issues of a Semi-Autonomous Robotic Assistant for the Health Care Environment. Journal of Intelligent and Robotic Systems, Kluwer Academic Publishers, Netherlands, 1998.

[12] Ettelt, E.; Schmidt, G.: Vision-based guidance and control of mobile forklift robot. Proc. of Int. Conf. on Recent Advances in Mechatronics. Istanbul, Turkey, August 14-16, 1995.

[13] Ettelt, E. und Schmidt, G.: Videobasierte Objekterkennung mittels musterbaumgestützter Kreuzkorrelation. - In: 13. Fachgespräch Autonome Mobile Systeme 1997, Stuttgart 1997, S. 72-83.

[14] Ettelt, E.; Schmidt, G.: 'Musterbaumgestuetzte Bildanalyse zur Objekterkennung'. Vortrag beim GMA-FA 5.5, 2. Juli, 1997 , Frankfurt, Germany.

[15] Fischer, C., Eine fortgeschrittene Bedienschnittstelle für einen semiautonomen mobilen Serviceroboter. Dissertation, Technische Universität München, Germany, 1998

[16] Fischer, C.; Buss, M.; Schmidt, G.: Hierachical supervisory control of service using human-robot-interface. - In: Proc. of the 1996 IEEE/RJS Int. Conf. of Intelligent Robots and Systems (IROS'96) Osaka, 4-8. November 1997, page 1408-1416.

[17] Fischer, C.; Hanebeck, U.D.; Schmidt, G.: A mobile service robot for the hospital and home environment. - In: Proc. International Advanced Robotics Programme IARP'97, Genova, Italy, 23- 24. Oktober. 1997, pp. 34-45.

[18] Fischer, C. and Schmidt, G.: Multi-Modal HuMan-Robot-Interface for Interaction with a Mobile Service Robot. - In: Proceedings of the 6th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD), Cassino, Italy, 1997, page 559-564

[19] Fischer, C. and Schmidt, G.: Multi-Modal HuMan-Robot-Interface for Mobile Health Care Robot. - In: Proceedings of the 1st Mobile Robotics Technology for Health Care Services Research Network (MobiNet), Athens, Greece, 1997, page 1-9.

[20] Fisher, R. A., The Use of Multiple Measurements in Taxonomy Problems, Ann. Eugenics, 7, S. 179-188, 1936.

[21] Fisher, R. A., The Statistical Utilization of Multiple Measurements, Ann. Eugenics, 8, S. 376-386, 1938.

[22] Gelfand, S. B., Ravishankar, C. S., Delp, E. J., An Iterative Growing and Pruning Algorithm for Classification Tree Design, IEEE Transactions on Pattern and Machine Intelligence, Vol. 13, No. 2, February 1991.

[23] Ghali, A., Daemi, M.F., Recognition Information, Proc. of the 13th Int. Conf. on Pattern Recognition ICPR 1996, Vienna, Austria, 25.-29. Aug. 1996.

[24] Gochet, W., Stam, A., Srinivasan, V., Chen, S., Multigroup Discriminant Analysis Using Linear Programming, Operations Research, Vol. 45, No. 2, March-April 1997.

[25] Gonzales, R.C., Wintz, P.: Digital Image Processing, Addison Wesley, Reading, USA, 1987.

[26] Hagar, G., The XVision System: A General-Purpose Substrate for Portable Real-Time Vision Applications (with K. Toyama). In Computer Vision and Image Understanding 69(1). S. 23 - 37.

[27] Hanebeck, U.D.; Fischer, C. and Schmidt, G.: ROMAN: A Mobile Robotic Assistant for Indoor Service Applications. Proc. of the IEEE RSJ International Conference on Intelligent Robots and Systems (IROS), Grenoble, France, 1997, S. 518-525.

[28] Hanebeck, U.D. and Schmidt, G.: Set-theoretic Localization of Fast Mobile Robots Using an Angle Measurement Technique. In Proceedings of the 1996 Int. Conf. on Robotics and Automation, Minneapolis, MN, Bd. 2, S. 1387-1394, 1996.

[29] Hanebeck, U.D. and Schmidt, G.: Mobile robot localization based on efficient processing of sensor data and set-theoretic state estimation. Proc. of ISER'97, (IROS), Barcelona, Spain, 15-18. June 1997, page 321-332.

[30] Hanebeck, U., Lokalisierung eines mobilen Roboters mittels effizienter Auswertung von Sensordaten und mengenbasierter Zustandsschätzung. Dissertation, Technische Universität München, Germany, 1997

[31] Haralick, R.M., Shapiro, L.G., Computer and Robot Vision, Vol. I, Vol. II, Addison-Wesley Publishing Company, 1992.

[32] Ishida, T., Korf, R., Moving-Target Search: A Real-Time Search for Changing Goals, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 6, June 1995.

[33] Jain, R.C., Binford, T.O., Ignorance, Myopia, and Naiveté in Computer Vision Systems, CVGIP Image Understanding, 53(1):112-117, 1991.

[34] Krebs, B., Schnelle Segmentierung von Tiefenbildern durch Approximation von Modellflächen in Ebenensegmente, Dissertation, TU Braunschweig, 1994.

[35] Krebs, B., Wahl, F., Robuste Objekterkennung auf der Basis eines flexiblen 3D Tiefensensors. In: Badisches Seminar für Robotik, Dezember 1995.

[36] Lanser, S., Zierl, C., MORAL: Ein System zur videobasierten Objekterkennung im Kontext autonomer, mobiler Systeme. - In: 12. Fachgespräch Autonome Mobile Systeme 1996, München 1996, S. 88-105.

[37] Lenz, R., Ein Verfahren zur Schätzung der Parameter geometrischer Bildinformationen, Dissertation, Lehrstuhl für Nachrichtentechnik, TU München, 1986.

[38] Li, X., Ferdousi, M., Chen, M., Nugyen, T.T., Image Matching with Multiple Templates, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, S. 610-613, 1986.

[39] Longstaff, I.D., On Extensions to Fisher's Linear Discriminant Function, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. Pami-9, No. 2, March 1987.

[40] Mostaghimi, M., Bayesian Estimation of a Decision Using Information Theory, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 27, No. 4, July 1997.

[41] Nayar S. K., Baker, S.: A Theory of Pattern Rejection. Columbia University Technical Report CUCS-013-95, Department of Computer Science, Columbia University, New York, 1995.

[42] Nayar S. K., Murase, H.: Dimensionality of Illumination in Appearance Matching. Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation, Minneapolis, Minnesota, April 1996, p. 1326-1332.

[43] Nayar S. K., Nene S. A., Murase, H.:Real-Time 100 Object Recognition System. Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation, Minneapolis, Minnesota, April 1996.

[44] Ney H., On the Probabilistic Interpretation of Neural Network Classifiers and Discriminative Training Criteria, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 2, February 1995.

[45] Niemann, H., Methoden der Mustererkennung, Akademische Verlagsgesellschaft, Frankfurt am Main, 1974.

[46] Nikolaev, A., Nayar, S. K., Transparent Grippers for Robot Vision, Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation, Minneapolis, Minnesota, S. 1644-1649, April 1996.

[47] Papageorgiou, M.: Optimierung. Oldenbourg Verlag, München, Wien, 1991.

[48] Pavlidis, T., Why progress in machine vision is so slow, Pattern Recognition Letters, 14(4):221-225, 1992.

[49] Pope, A. R., Model Based Object Recognition. A Survey of Recent Research. Technical Report TR-94-04, University Berkeley California, 1994.

[50] Preusche, C., Vertical Edge Tracking for Mobile Robot Localization and Map Building, Diplomarbeit, Lehrstuhl für Steuerungs- und Regelungstechnik, TU-München, 1998.

[51] Pinz, A., Bildverstehen, Springerverlag, Wien, New York, 1994.

[52] Quinlan, J. R., Decision Trees and Decisionmaking, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, March/April 1990.

[53] Ramapriyan, H.K., A multilevel approach to sequential detection of pictorial features, IEEE Transactions on Comput., Vol. 25, no. 1, pp. 66-78, 1976.

[54] Rojas, R., Theorie der neuronalen Netze, Springer-Verlag, Berlin, Heidelberg New York, 1993.

[55] Rontogiannis, A., Dimopoulos, N. J., A Probabilistic Approach for Reducing the Search Cost in Binary Decision Trees, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 2, February 1995.

[56] Rowley, H., Baluja, S., Kanade, T., Neural Network-Based Face Detection, IEEE Transactions on Pattern and Machine Intelligence, Vol. 20, No. 1, Januar 1998.

[57] Ruske, G., Automatische Spracherkennung, Methoden der Klassifikation und Merkmalsextraktion, Oldenbourg Verlag, München, Wien, 1988.

[58] Russer, P., Informationstechnik, VCH Verlagsgesellschaft, Weinheim, 1988.

[59] Schneidermann, H., Nashman, M., A Discriminating Feature Tracker for Vision-Based Autonomous Driving, IEEE Transactions on Robotics and Automation, Vol. 10, No. 6, December 1994.

[60] Schmidt, G., Grundlagen Intelligenter Roboter, Skriptum zur Vorlesung, München, 1998.

[61] Seitz, M., Untersuchungen zur Nutzung von Bildverarbeitung für Manipulationsaufgaben in der Robotik, Dissertation, Shaker Verlag, Aachen, 1996.

[62] Shannon, C. E., A Mathematical Theory of Communication, The Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656, July, October, 1948.

[63] Shannon, C. E., Communication in the presence of noise, Proceedings of IRE, S. 10-22, 1949.

[64] Sheridan, T. B., Reflections on Information and Information Value, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 1, January 1995.

[65] Smith, C. A., Some Examples of Discrimination, Ann. Eugenics, 13, S. 272-282, 1947.

[66] Stahs, T., Wahl, F., Recognition of Polyhedral Objects under Perspective View. Computers and Artificial Intelligence, 11(2), S. 155-172, 1992.

[67] Sung, K., Poggio, T., Example-Based Learning for View-Based Human Face Detection, IEEE Transactions on Pattern and Machine Intelligence, Vol. 20, No. 1, Januar 1998.

[68] Swets, D., Weng, J., Discriminant Analysis and Eigenspace Partition Tree for Face and Object Recognition from Views, 2nd International Conference on Automatic Face- and Gesture-Recognition, pp 192-197, Vermont, October 1996.

[69] Swets, D., Weng, J., Using Discriminant Eigenfeatures for Image Retrieval, IEEE Transactions on Pattern and Machine Intelligence, Vol. 18, No. 8, S. 831-836, August 1996.

[70] Turk, M. A., Pentland, A. P., Face Recognition Using Eigenfaces, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 586-591, June 1991.

[71] Vapnik, V., Chervonenkis, A., On the uniform convergence of relative frequencies of events to their probabilities, Theory of Probability and its Applications, Vol. 16, S. 264-280., 1971

[72] Wunsch, P., Hirzinger, G., Real-Time Visual Tracking of 3-D Objects with Dynamic Handling of Occlusion, Proc. of the 1997 IEEE Int. Conf. on Robotics and Automation, Albuquerque, New Mexico, S. 2868-2873, April 1997.

[73] Zeitler, S., Objektlokalisierung auf der Basis von Musterbäumen, Diplomarbeit, Lehrstuhl für Steuerungs- und Regelungstechnik, TU-München, 1997.

[74] Zhou X.J., Dillon, T.S., A Statistical-Heuristic Feature Selection Criterion for Decision Tree Induction, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 8, August 1991.